UNIVERSITY OF CAMBRIDGE

Advanced Data Science

Lecture 5 : Introduction to Statistical Learning

Carl Henrik Ek - che29@cam.ac.uk

7th of November, 2022

http://carlhenrik.com

The course so far

Access how to get and combine the data-sources for a potential problem

The course so far

Access how to get and combine the data-sources for a potential problem

Assess things to do with data before you have a question

The course so far

Access how to get and combine the data-sources for a potential problem

Assess things to do with data before you have a question

Address what you can do when you have data and a question to answer

Access

- ?

Assess

- Introduction to Probability (IA)
- Scientific Computing (IA)
- Cloud Computing (II)
- ....

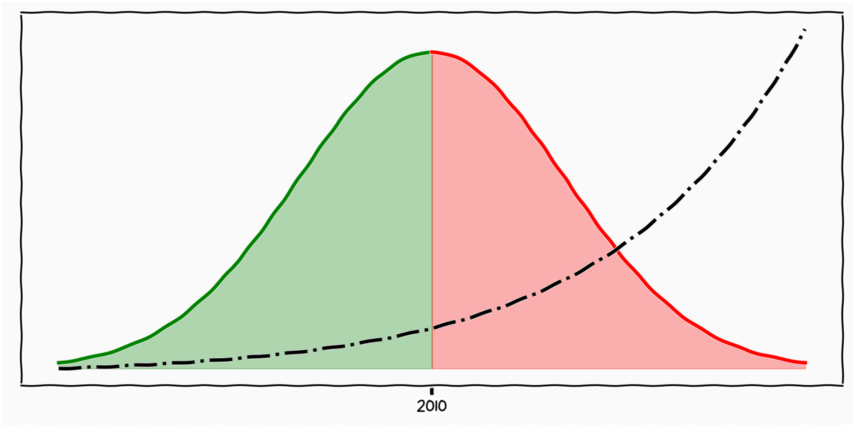
Address

- ML and Real-world Data (IA)
- Data Science (IB)
- AI (IB)
- ML & Bayesian Inference (II)
- Deep NN (II)
- Randomised Algorithms (II)
- ....

Why?

Data centric thinking

"You need to put Machine Learning in the context of data (and humans)"


2010


STOP

Reasons to use ML

- Tasks that are too hard to program

Reasons to use ML

- Tasks that are too hard to program
  - speech recognition

Reasons to use ML

- Tasks that are too hard to program
  - speech recognition
  - image understanding

Reasons to use ML

- Tasks that are too hard to program
  - speech recognition
  - image understanding
- Tasks beyond our capability

Reasons to use ML

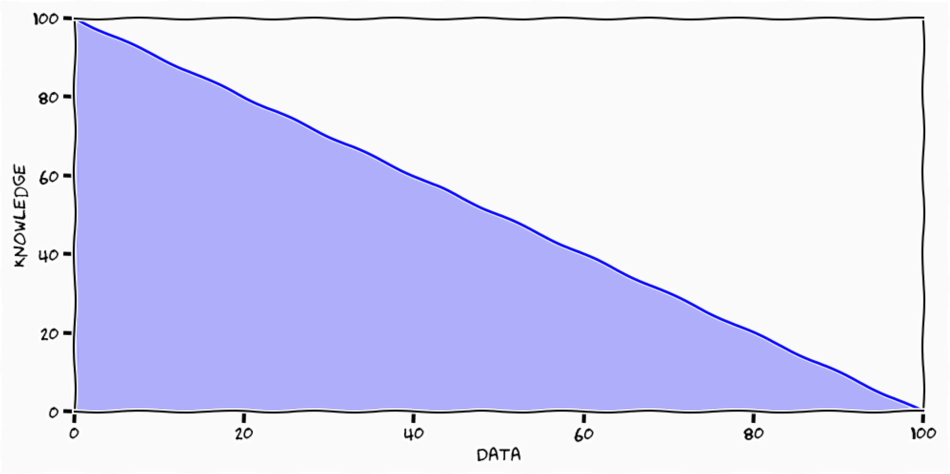
- Tasks that are too hard to program
  - speech recognition
  - image understanding
- Tasks beyond our capability
  - weather prediction

Reasons to use ML

- Tasks that are too hard to program
  - speech recognition
  - image understanding
- Tasks beyond our capability
  - weather prediction
  - web search

Reasons to use ML

- Tasks that are too hard to program
  - speech recognition
  - image understanding
- Tasks beyond our capability
  - weather prediction
  - web search
- Machine Learning bridges the knowledge gap by data


KNOWLEDGE
100
80
60
40
20
0
0
20
40
60
80
100
DATA

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta)p(\theta)}{p(\mathcal{D})}$$

- Inductive biases comes into the learning procedure

Machine Learning and Knowledge

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta)p(\theta)}{p(\mathcal{D})}$$

- Inductive biases comes into the learning procedure
- *Most knowledge is introduced before we apply ML*

  access what data did I acquire?

  assess how did I prepare/treat the data?

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta)p(\theta)}{p(\mathcal{D})}$$

- Inductive biases comes into the learning procedure
- *Most knowledge is introduced before we apply ML*

  access what data did I acquire?

  assess how did I prepare/treat the data?
- The idea of the $80/20$

Today

- what is actually machine learning?

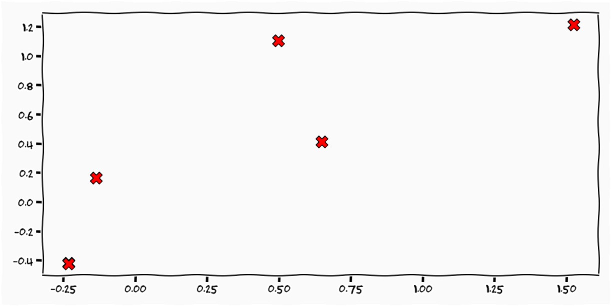
Today

- what is actually machine learning?
- what can machine learning actually do?

Today

- what is actually machine learning?
- what can machine learning actually do?
- put machine learning into context

Statistical Learning

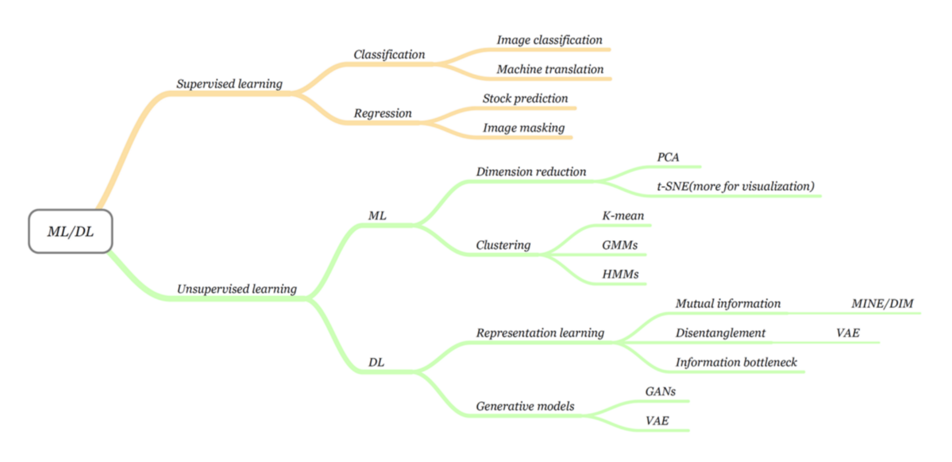


Supervised Learning $p(y \mid x)$

"Unsupervised" Learning $p(y)$

Reinforcement Learning $p(\pi, f \mid \mathcal{L})$

Machine Learning Methods



Formalism [Shalev-Shwartz et al., 2014]

Domain Set $\mathcal{X}$ the set of measurements/objects that we want to label (input)

Domain Set $\mathcal{X}$ the set of measurements/objects that we want to label (input)

Label Set $\mathcal{Y}$ the set of outputs

Formalism [Shalev-Shwartz et al., 2014]

Domain Set $\mathcal{X}$ the set of measurements/objects that we want to label (input)

Label Set $\mathcal{Y}$ the set of outputs

Training Data $\mathcal{S}$ a finite sequence of pairs in $\mathcal{X} \times \mathcal{Y}$

Data Distribution $\mathcal{D}$ probability distribution governing the measurements

Formalism [Shalev-Shwartz et al., 2014]

Data Distribution $\mathcal{D}$ probability distribution governing the measurements

Data Generation $f : \mathcal{X} \to \mathcal{Y}$ the underlying generating process that we wish to recover

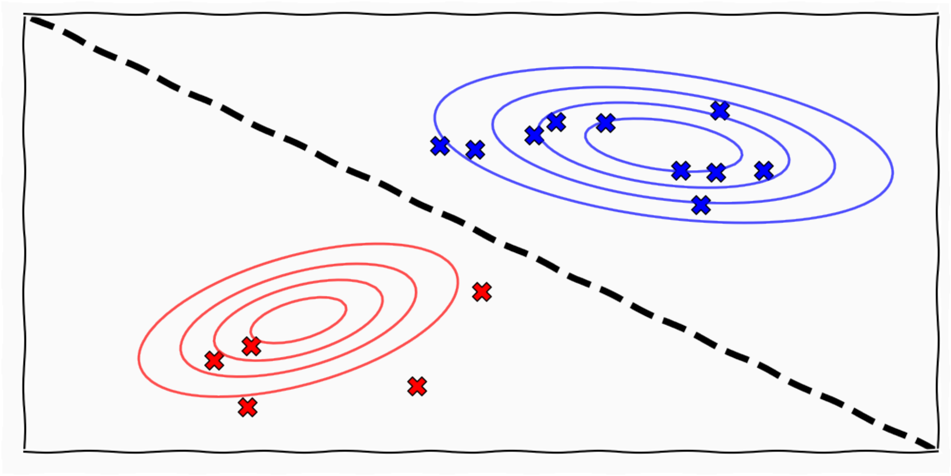
Formalism [Shalev-Shwartz et al., 2014]

Data Distribution $\mathcal{D}$ probability distribution governing the measurements

Data Generation $f : \mathcal{X} \to \mathcal{Y}$ the underlying generating process that we wish to recover

Prediction Rule $h : \mathcal{X} \to \mathcal{Y}$ what we wish to recover, the object that encodes the recovered knowledge

Classification



Measure of Success

$$L_{\mathcal{D},f}(h) := \mathcal{D}(\{x : h(x) \neq f(x)\})$$

- measure of success as probability of misclassified points (true risk)

Measure of Success

$$L_{\mathcal{D},f}(h) := \mathcal{D}(\{x : h(x) \neq f(x)\})$$

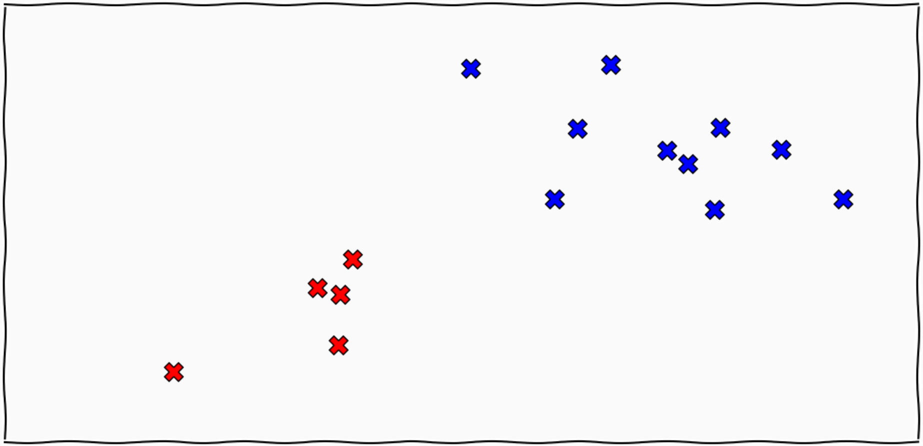
- measure of success as probability of misclassified points (true risk)
- we do not have access to $\mathcal{D}$

Measure of Success

$$L_{\mathcal{D},f}(h) := \mathcal{D}(\{x : h(x) \neq f(x)\})$$

- measure of success as probability of misclassified points (true risk)
- we do not have access to $\mathcal{D}$
- we do not have access to f

Classification

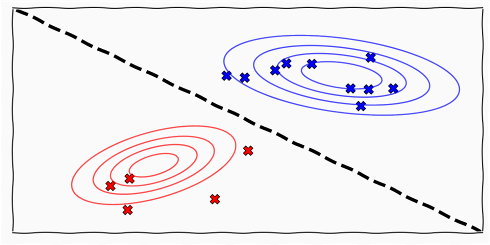


Empirical Risk Minimisation

$$L_{\mathcal{S}}(h) := \frac{|\{i \in [m] : h(x_i) \neq y_i\}|}{m}$$

- We assume that $\mathcal{S} \sim \mathcal{D}$
- Empirical measure of risk

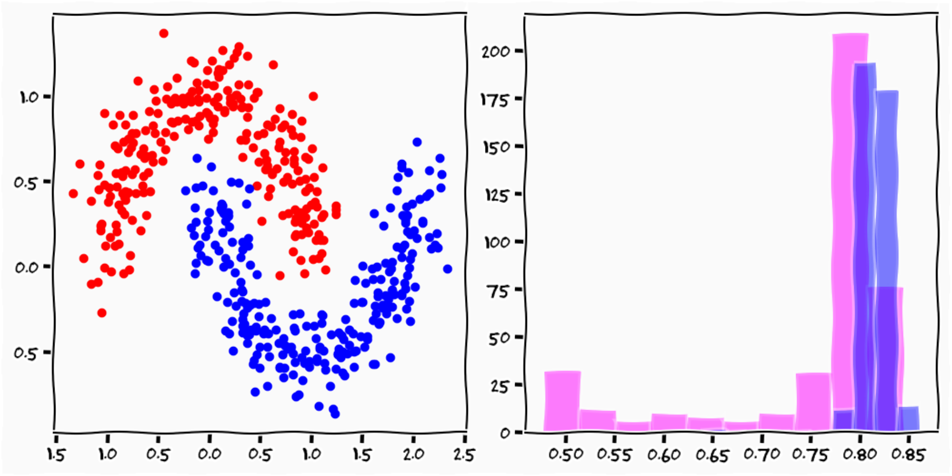
Overfitting



$$\mathcal{D} = \frac{1}{3}\mathcal{N}(\cdot,\cdot) + \frac{2}{3}\mathcal{N}(\cdot,\cdot)$$

$$h_{\mathcal{S}}(x) = \begin{cases} y_i & \text{if } \exists i \in [m] \text{s.t. } x_i = x \\ 0 & \text{otherwise} \end{cases}$$

- $L_{\mathcal{S}}(h_{\mathcal{S}}) = 0$ for all training data-sets
- if label 0 corresponds to red $L_{\mathcal{D}}(h_{\mathcal{S}}) = \frac{1}{3}$
- if label 0 corresponds to blue $L_{\mathcal{D}}(h_{\mathcal{S}}) = \frac{2}{3}$


1.0
0.5
0.0
0.5
1.5
1.0
0.5
0.0
0.5
1.0
1.5
2.0
2.5
200
175
150
125
100
75
50
25
0
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85

Algorithm

$$L_{\mathcal{S}}(A(\mathcal{S})) := \frac{|\{i \in [m] : h(x_i) \neq y_i\}|}{m}$$

- We use an algorithm $A : \mathcal{S} \to h$ to find a hypothesis

Finite Hypothesis Classes

$$h_{\mathcal{S}} \in \operatorname*{argmin}_{h \in \mathcal{H}} L_{\mathcal{S}}(h)$$

- We cannot parametrise all possible hypothesis

Statistical Learning Questions

m How much "better" will my estimate get with more data do I need?

Statistical Learning Questions

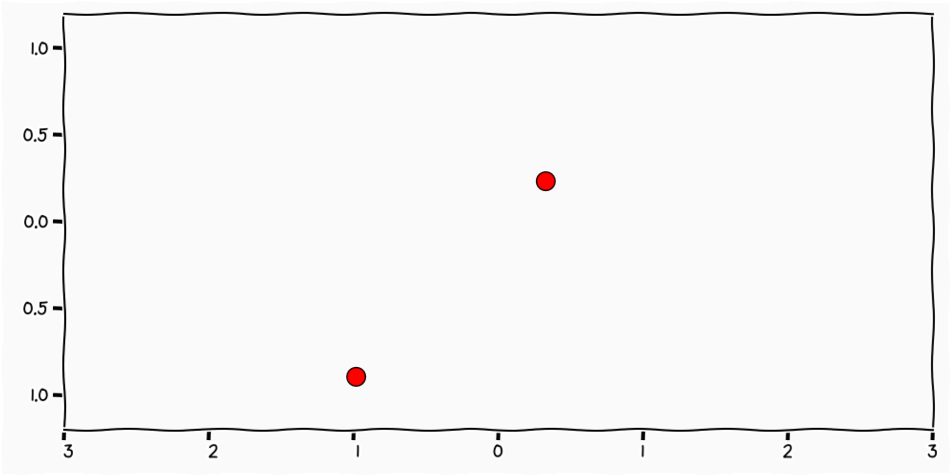
m How much "better" will my estimate get with more data do I need?

A How much does my solution depend on what the algorithm find?

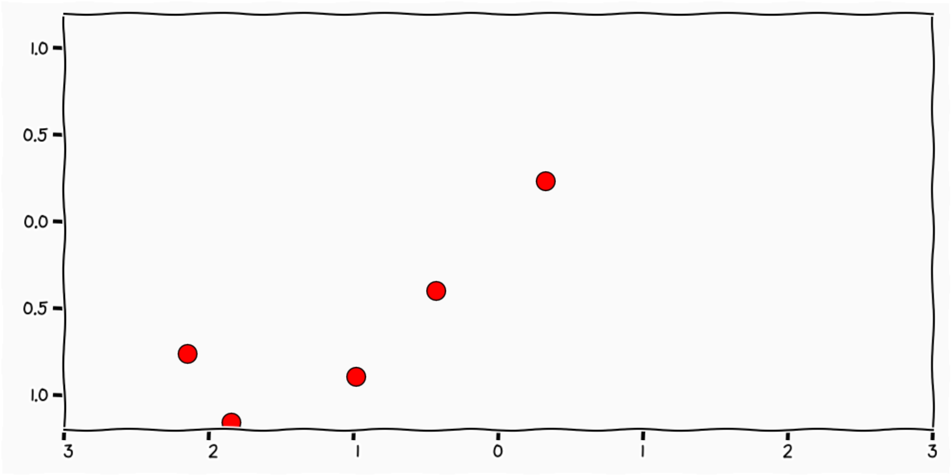
Statistical Learning Questions

- m How much "better" will my estimate get with more data do I need?
- A How much does my solution depend on what the algorithm find?
- H How does my solution depend on the hypothesis class I choose?

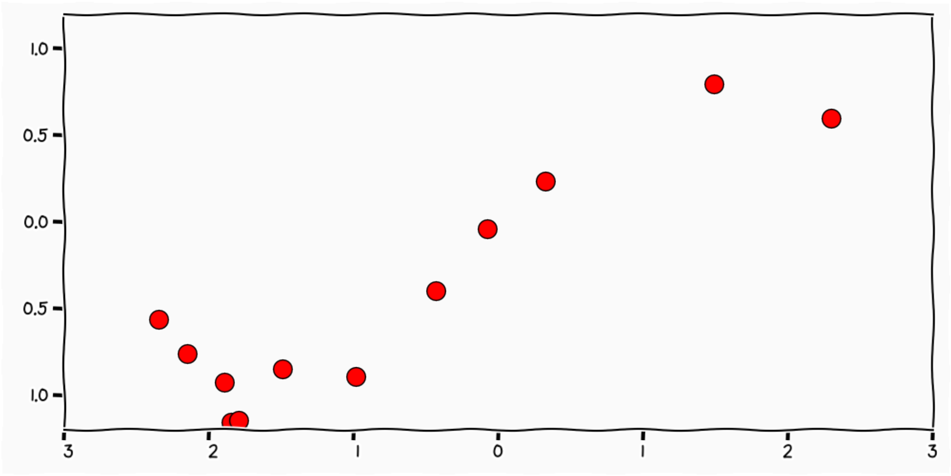
Regression



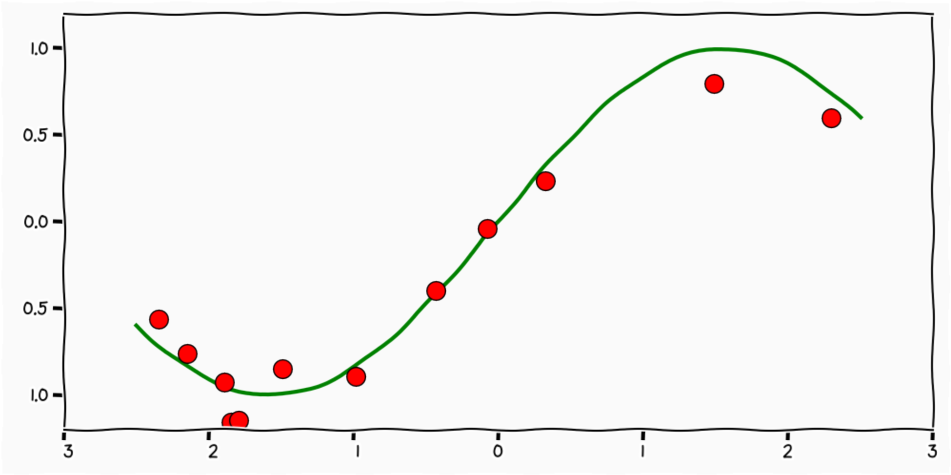
Regression



Regression



Regression



The No-Free Lunch Theorem

- Every algorithm that learns something useful does so by making assumptions

The No-Free Lunch Theorem

- Every algorithm that learns something useful does so by making assumptions
- There exists no universal learner/method/algorithm

The No-Free Lunch Theorem

- Every algorithm that learns something useful does so by making assumptions
- There exists no universial learner/method/algorithm
- There is no free lunch algorithm

The Components of a Learning System

$\mathcal{A}$ my learning algorithm

The Components of a Learning System

$\mathcal{A}$ my learning algorithm

$\mathcal{H}$ my hypothesis class

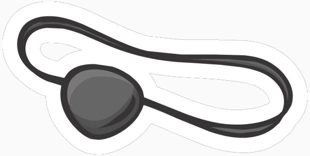
The Components of a Learning System

$\mathcal{A}$ my learning algorithm

$\mathcal{H}$ my hypothesis class

$\mathcal{S}$ my finite trainingset

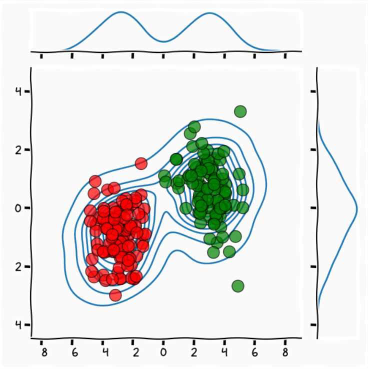
Assumptions: Algorithms



Statistical Learning

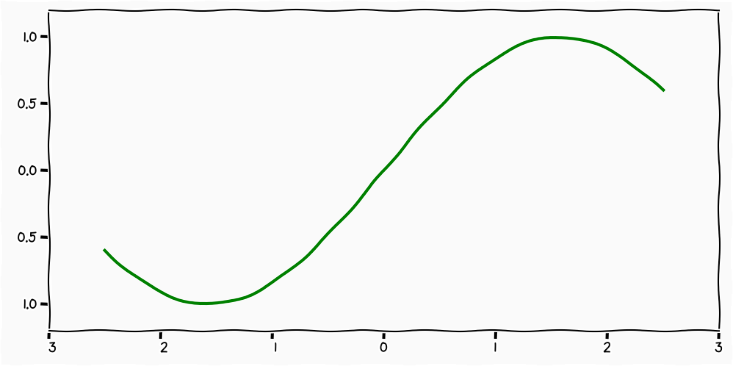
$$\mathcal{A}_{\mathcal{H}}(\mathcal{S})$$

Assumptions: Biased Sample



Statistical Learning

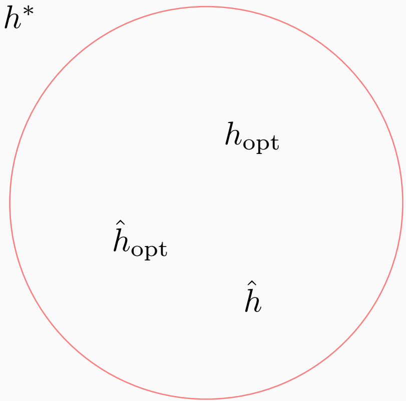
$$\mathcal{A}_{\mathcal{H}}(\mathcal{S})$$



Statistical Learning

$$\mathcal{A}_{\mathcal{H}}(\mathcal{S})$$

The Error Decomposition



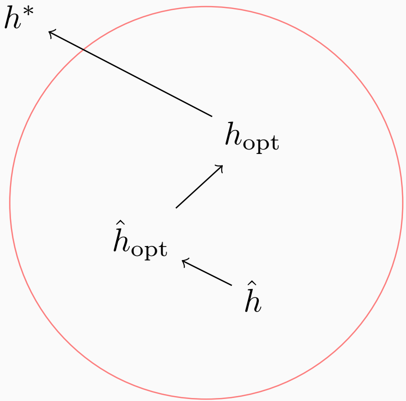
h^* the optimal predictor

$h_{\textbf{opt}}$ the optimal hypothesis

$\hat{h}_{\textbf{opt}}$ the optimal hypothesis on training data

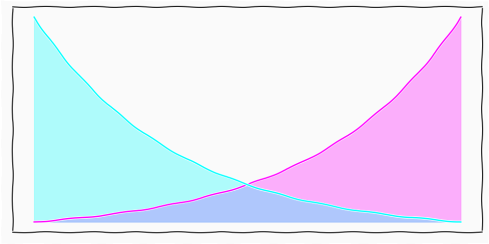
$\hat{h}$ the hypothesis found by learning algorithm

The Error Decomposition



$$
\begin{aligned}
&\epsilon(\hat{h}) - \epsilon(h^*) \\
&= \underbrace{\epsilon(h_{\text{opt}}) - \epsilon(h^*)}_{\text{Approximation}} \\
&+ \underbrace{\epsilon(\hat{h}_{\text{opt}}) - \epsilon(h_{\text{opt}})}_{\text{Estimation}} \\
&+ \underbrace{\epsilon(\hat{h}) - \epsilon(\hat{h}_{\text{opt}})}_{\text{Optimisation}}
\end{aligned}
$$

The Bias-Complexity Trade-off



High Complexity low bias (ϵ_{app} small), but high risk of overfitting (ϵ_{est} large)

Low Complexity high bias (ϵ_{app} large), low risk of overfitting (ϵ_{est} small)

Universal Learner

Theorem (The No-Free-Lunch Theorem)
Let A be any learning algorithm fo the task of binary classification with respect to $0-1$ loss over the domain $\mathcal{X}$. Let m be any number smaller than $\frac{|\mathcal{X}|}{2}$. Then there exists a distribution $\mathcal{D}(\{\mathcal{X} \times \{0,1\}\})$ such that,

- *There exists a function $f : \mathcal{X} \to \{0,1\}$ with $L_{\mathcal{D}}(f) = 0$*
- *With probability at least $\frac{1}{7}$ over the choice of $\mathcal{S} \sim \mathcal{D}^m$ we have $L_{\mathcal{D}}(A(S)) \geq \frac{1}{8}$*

The No-Free Lunch Theorem

- There exists no universal learner

The No-Free Lunch Theorem

- There exists no universal learner
- For every learner there exist a task on which it fails

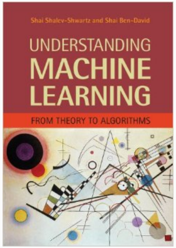
The No-Free Lunch Theorem

- There exists no universal learner
- For every learner there exist a task on which it fails
- Every algorithm that learns something useful does so by assumptions

The No-Free Lunch Theorem

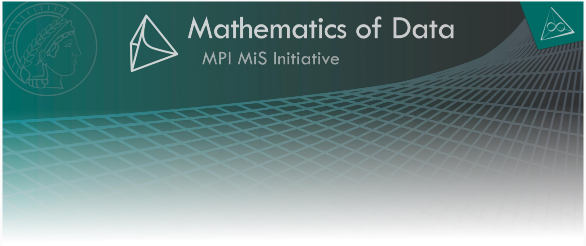
- There exists no universal learner
- For every learner there exist a task on which it fails
- Every algorithm that learns something useful does so by assumptions
- *There is no free lunch algorithm*

Statistical Learning Theory Further Reading



- Shai Shalev-Shwartz et al. (2014). *Understanding Machine Learning: From Theory to Algorithms*. New York, NY, USA: Cambridge University Press https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/

Statistical Learning Theory Further Reading



- O. Bousquet et al. (2004). "Introduction to Statistical Learning Theory". In: vol. Lecture Notes in Artificial Intelligence 3176. Heidelberg, Germany: Springer, pp. 169–207, http://www.econ.upf.edu/~lugosi/mlss_slt.pdf

- We can never have sufficient data

Statistical Learning Summary

- We can never have sufficient data
- We can never find a method that will guarantee to find the right solution

Statistical Learning Summary

- We can never have sufficient data
- We can never find a method that will guarantee to find the right solution
- We can never be certain about the true risk of our outcome

Explicit vs. Tacit Knowledge

Dangers of misattribution

The Gap

Access enormous inductive bias in what data to acquire

The Gap

Access enormous inductive bias in what data to acquire

Assess human bias in what questions will probably be asked

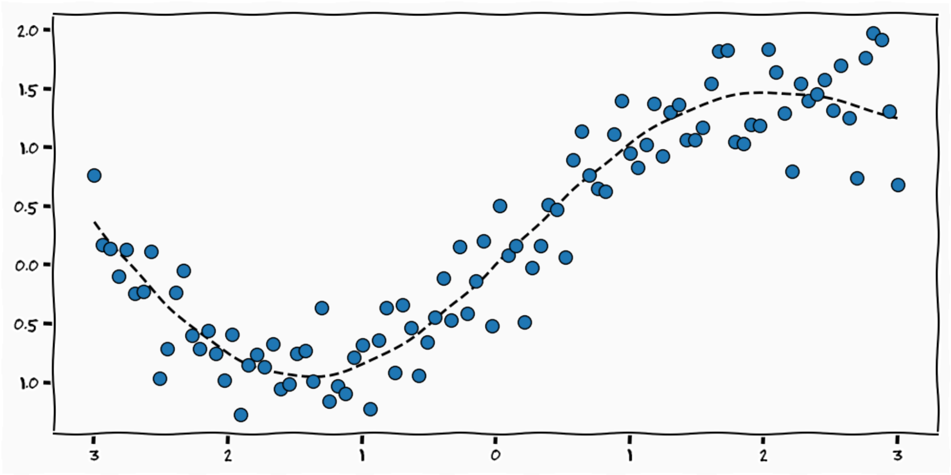
The Gap

Access enormous inductive bias in what data to acquire

Assess human bias in what questions will probably be asked

Address "it is just curve fitting"

Curve Fitting is Really Fun

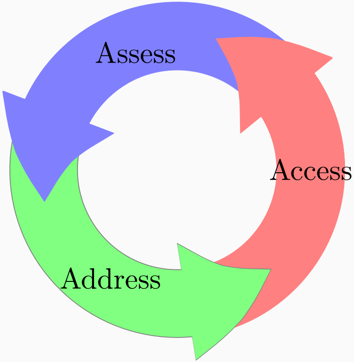


Generalised Linear Models

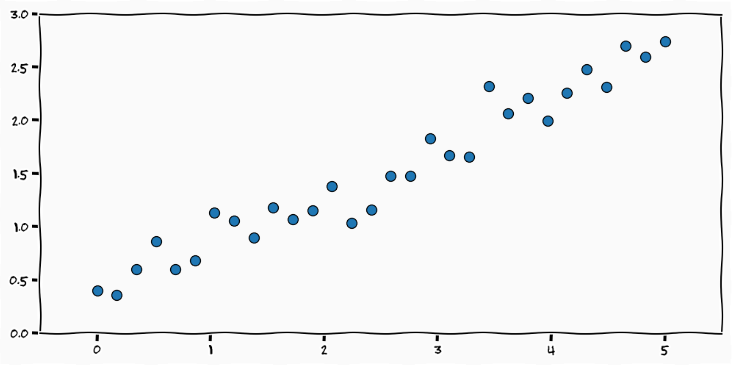
Limited Hypothesis Classes

$$h \in \mathcal{H}$$

Address Requirements for Data Science



Formalism

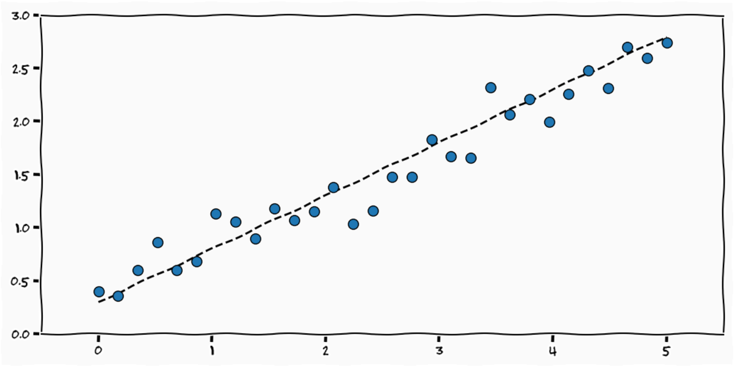


$\mathbf{x} \in \mathcal{X}$ explanatory variable

$y \in \mathcal{Y}$ response variable

Task *explain the response by the explanatory variables*

Linear Regression [Bishop, 2006]



$$y_i = \sum_{j=1}^{d} \beta_j x_{ij} + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma^2).$$

Linear Regression Prediction

$$\mathbb{E}[y_i \mid \mathbf{x}_i] = \mathbb{E}\left[\sum_{j=1}^{d} \beta_j x_{ij} + \epsilon\right]$$

Linear Regression Prediction

$$
\begin{aligned}
\mathbb{E}[y_i \mid \mathbf{x}_i] &= \mathbb{E}\left[\sum_{j=1}^{d} \beta_j x_{ij} + \epsilon\right] \\
&= \mathbb{E}\left[\sum_{j=1}^{d} \beta_j x_{ij}\right] + \mathbb{E}[\epsilon]
\end{aligned}
$$

Linear Regression Prediction

$$
\begin{aligned}
\mathbb{E}[y_i \mid \mathbf{x}_i] &= \mathbb{E}\left[\sum_{j=1}^{d} \beta_j x_{ij} + \epsilon\right] \\
&= \mathbb{E}\left[\sum_{j=1}^{d} \beta_j x_{ij}\right] + \mathbb{E}[\epsilon] \\
&= \sum_{j=1}^{d} \beta_j x_{ij} + 0.
\end{aligned}
$$

Linear Regression

$$y_i = \sum_{j=1}^{d} \beta_j x_{ij} + \epsilon,$$

Linear Regression

$$y_i = \sum_{j=1}^{d} \beta_j x_{ij} + \epsilon,$$

$$y_i + \epsilon = \sum_{j=1}^{d} \beta_j x_{ij},$$

$$y_i = \sum_{j=1}^{d} \beta_j x_{ij} + \epsilon,$$

$$y_i + \epsilon = \sum_{j=1}^{d} \beta_j x_{ij},$$

$$\hat{y}_i = \sum_{j=1}^{d} \beta_j x_{ij},$$

Linear Regression

$$y_i = \sum_{j=1}^{d} \beta_j x_{ij} + \epsilon,$$

$$y_i + \epsilon = \sum_{j=1}^{d} \beta_j x_{ij},$$

$$\hat{y}_i = \sum_{j=1}^{d} \beta_j x_{ij},$$

$$\hat{y}_i \sim \mathcal{N}(y_i, \sigma^2) = \mathcal{N}\left(\sum_{j=1}^{d} \beta_j x_{ij}, \sigma^2\right),$$

Generalised Linear Models [McCullagh et al., 1989]

$$g(\mathbb{E}[y_i \mid \mathbf{x}_i]) = \sum_{j=1}^{d} \beta_j x_{ij},$$

$g(\cdot)$ link function

$y \sim \mathcal{D}$ Exponential Dispersion Family

$\sum_{j=1}^{d} \beta_j x_{ij}$ Linear predictor

Generalised Linear Models [McCullagh et al., 1989]

$$\mathbb{E}[y_i \mid \mathbf{x}_i] = g^{-1}\Big(\sum_{j=1}^{d} \beta_j x_{ij}\Big),$$

- The inverse of the *link* maps the linear predictor to the first moment of the response

[1] https://towardsdatascience.com/glms-part-iii-deep-neural-networks-as-recursive-generalized-linear-URL

Generalised Linear Models [McCullagh et al., 1989]

$$\mathbb{E}[y_i \mid \mathbf{x}_i] = g^{-1}(\sum_{j=1}^{d} \beta_j x_{ij}),$$

- The inverse of the *link* maps the linear predictor to the first moment of the response
- Linear regression the link is identity

[1] https://towardsdatascience.com/glms-part-iii-deep-neural-networks-as-recursive-generalized-linear-URL

Generalised Linear Models [McCullagh et al., 1989]

$$\mathbb{E}[y_i \mid \mathbf{x}_i] = g^{-1}(\sum_{j=1}^{d} \beta_j x_{ij}),$$

- The inverse of the *link* maps the linear predictor to the first moment of the response
- Linear regression the link is identity
- Looks an awful lot like a neural network[1]

[1] https://towardsdatascience.com/glms-part-iii-deep-neural-networks-as-recursive-generalized-linear-URL

Exponential Dispersion Family [2]

$$f(y;\theta,\phi) = e^{\frac{\theta y - b(\theta)}{a(\phi)} + c(y,\phi)},$$

θ location parameter

ϕ scale parameter

[2] https://en.wikipedia.org/wiki/Exponential_dispersion_model

Model	Response Variable	Link	Explanatory Variable
Linear Regression	Normal	Identity	Continuous
Logistic Regression	Binomial	Logit	Mixed
Poisson Regression	Poisson	Log	Mixed
ANOVA	Normal	Identity	Categorical
ANCOVA	Normal	Identity	Mixed
Loglinear	Poisson	Log	Categorical
Multinomial response	Multinomial	Generalized Logit	Mixed

Summary

Summary

- Brief introduction to statistical learning theory
- Take home
  - ML models and algorithms is only a small part of the story
  - we are doing a lot better than we should be
  - we are not sure what we are doing but it somehow works

Summary

- Brief introduction to statistical learning theory
- Take home
  - ML models and algorithms is only a small part of the story
  - we are doing a lot better than we should be
  - we are not sure what we are doing but it somehow works
  - it is not explicit knowledge that pushes data-science forward, it is tacit among data scientist

The rest of the module

Tuesday (8/11) Lab: Generalised Linear Models

Wednesday (9/11) Lecture Generalised Linear Models

Friday (11/11) Lecture: Unsupervised Learning

Monday (14/11) Lecture: Visualisation

Tuesday (15/11) Tick: Generalised Linear Models

Wednesday (16/11) Lecture

Thursday (17/11) Tick 4

Friday (18/11) Summary and Q&A

eof

References

- Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.
- Bousquet, O., S. Boucheron, and G. Lugosi (2004). "Introduction to Statistical Learning Theory". In: vol. Lecture Notes in Artificial Intelligence 3176. Heidelberg, Germany: Springer, pp. 169–207.
- McCullagh, P. and J. A. Nelder (1989). *Generalized Linear Models*. London, UK: Chapman Hall / CRC: Chapman Hall / CRC.

Shalev-Shwartz, Shai and Shai Ben-David (2014). *Understanding Machine Learning: From Theory to Algorithms*. New York, NY, USA: Cambridge University Press.