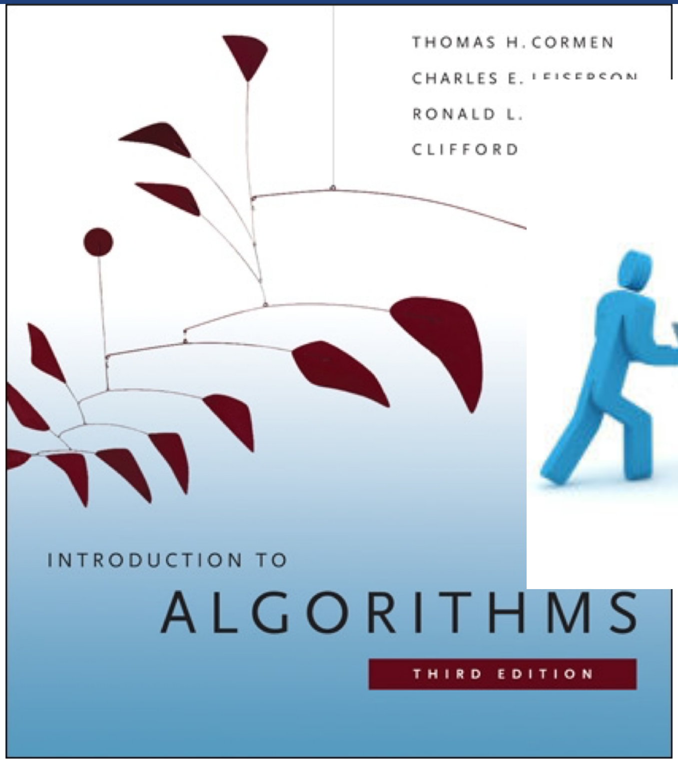
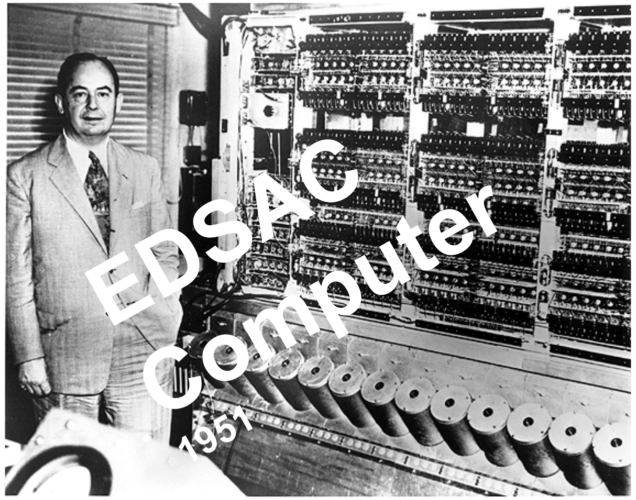
Robust algorithmics - a foundation for science?!

Joachim M. Buhmann

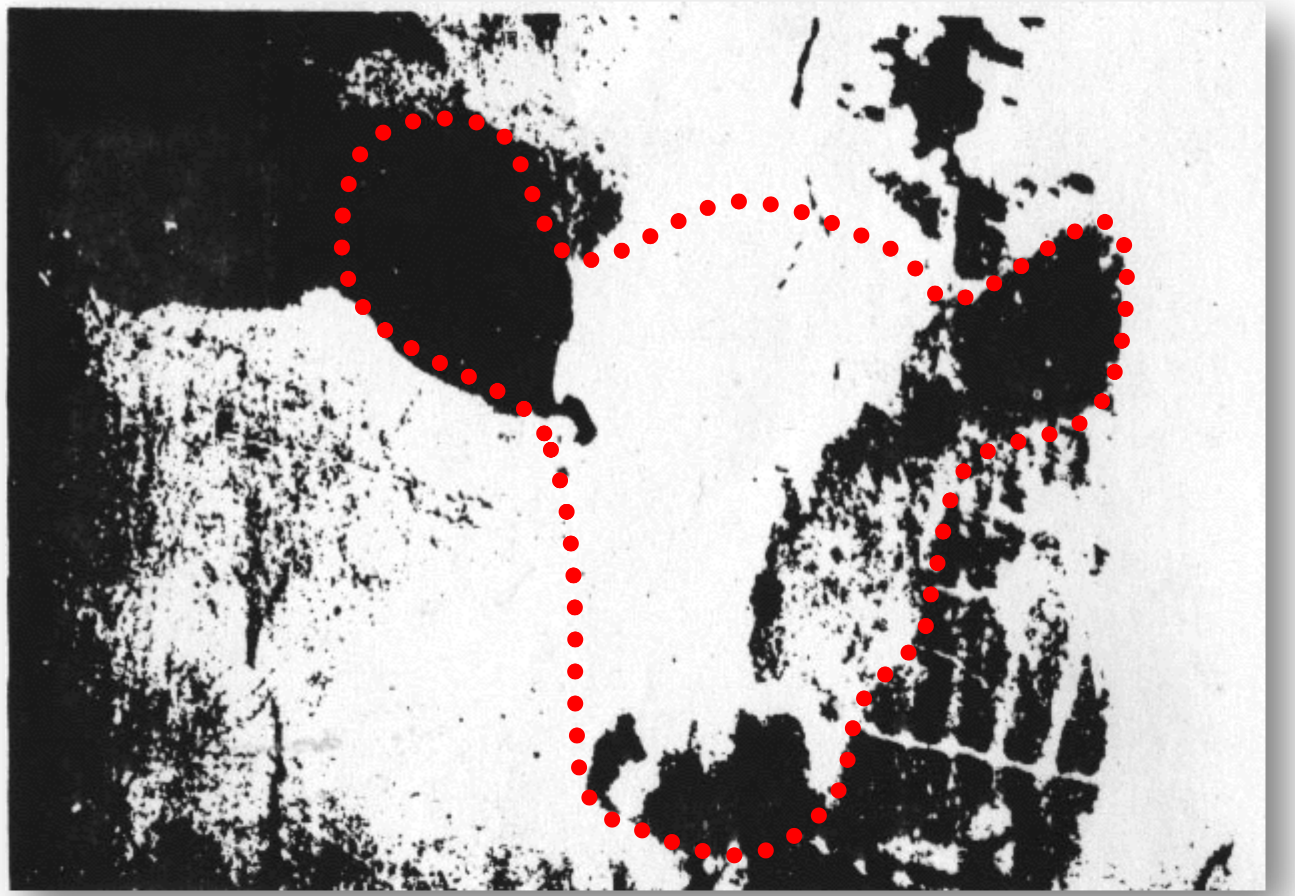
Institute for Machine Learning, D-INFK, ETH Zurich

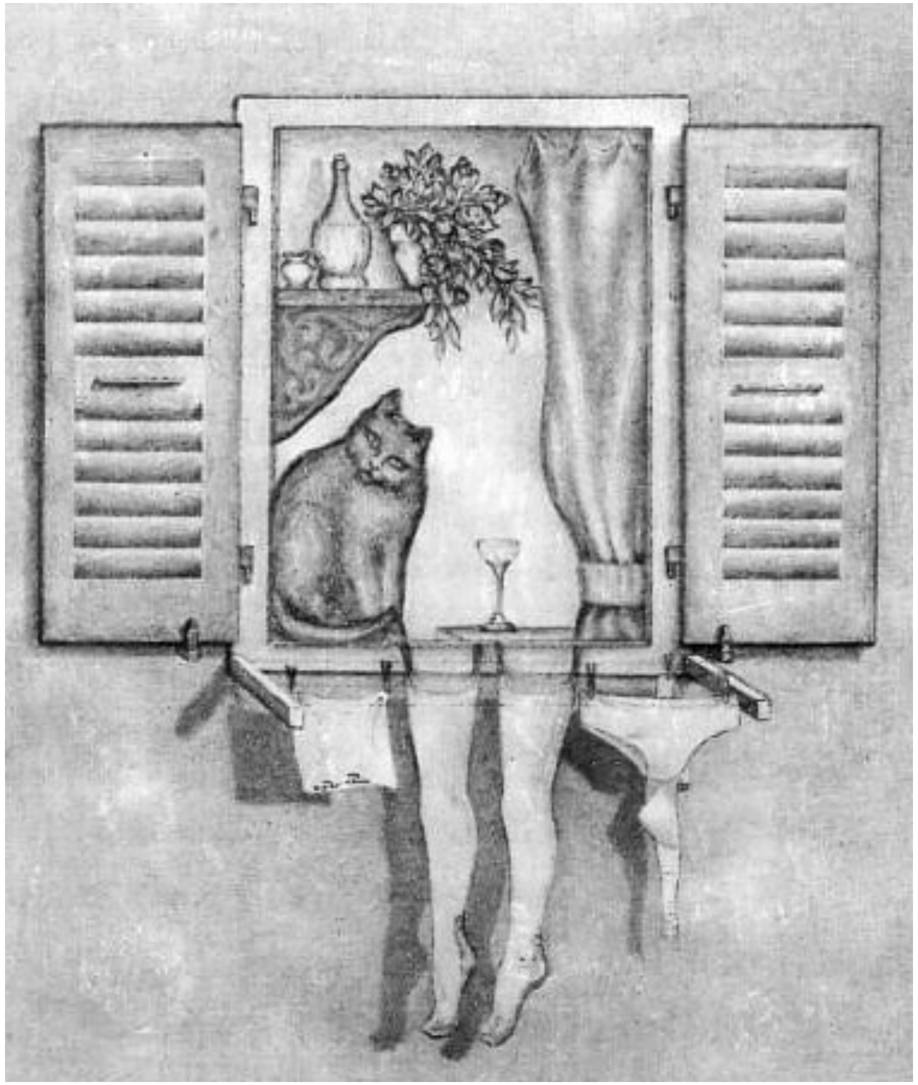
Our world, in which we live!





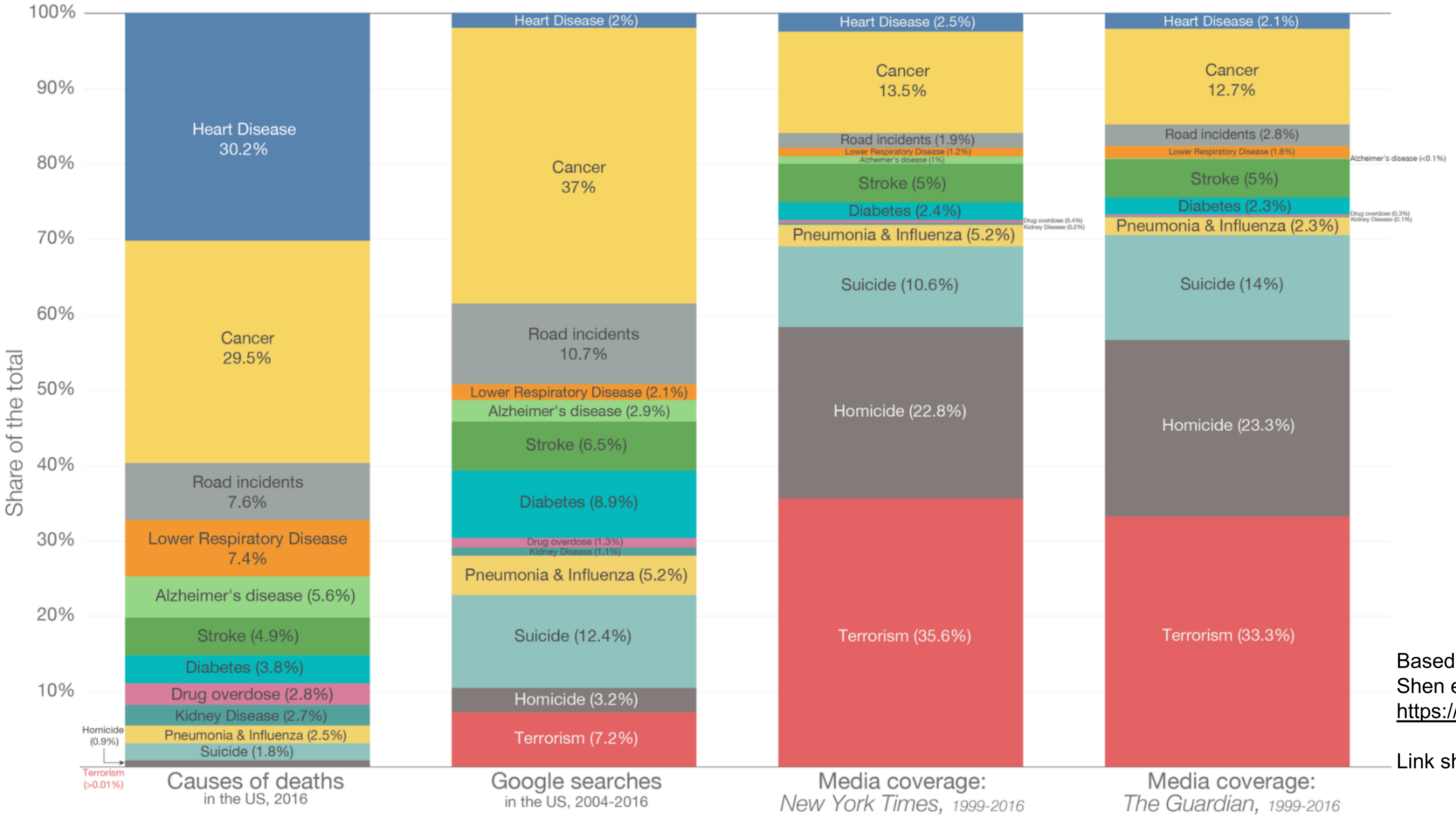
Seeing patterns in data (vision) is difficult!





Causes of death in the US

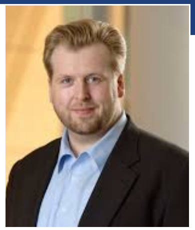
What Americans die from, what they search on Google, and what the media reports on

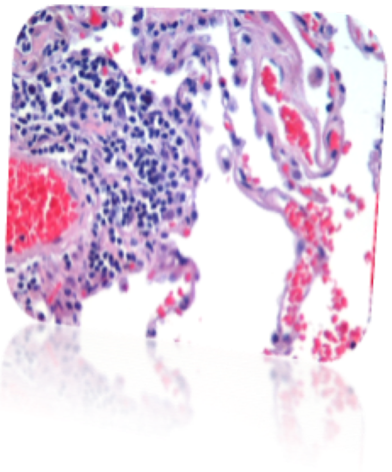


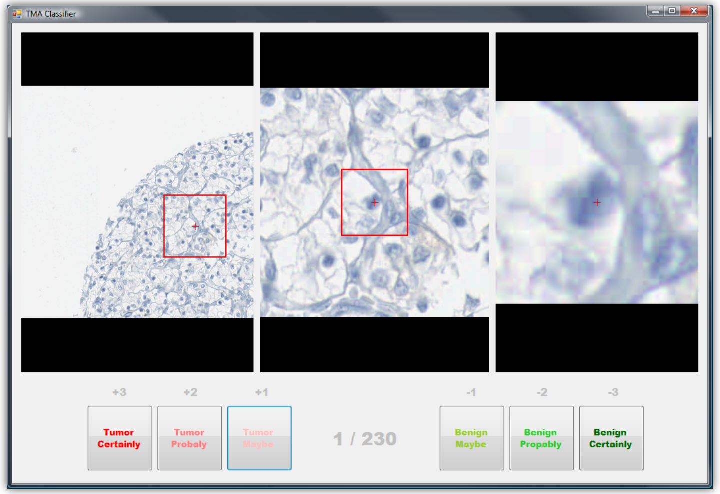
Based on data from
Shen et al. (2018) – Death: reality vs. reported.
https://owenshen24.github.io/charting-death

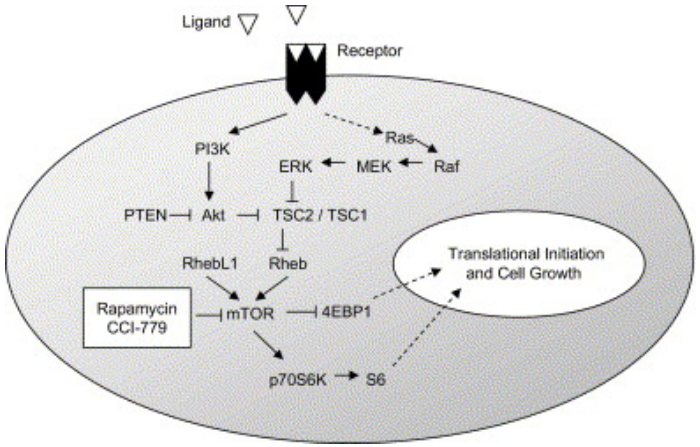
Link shared by **Alessandro Curioni**, IBM Research

IT value generation in personalized medicine



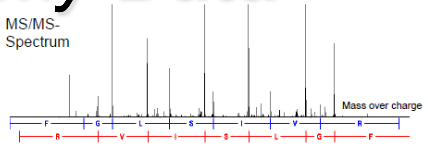


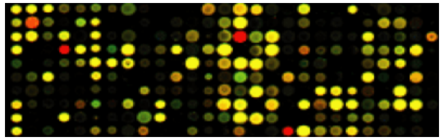


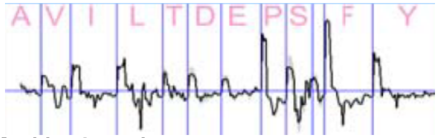


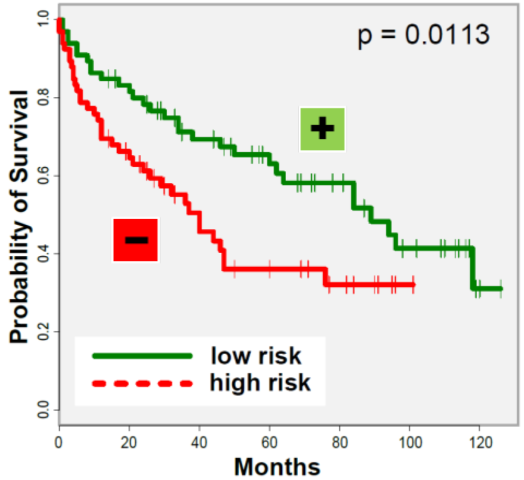
Activation of the mTOR Signaling Pathway in Renal Clear Cell Carcinoma. Robb et al., J Urology 177:346 (2007)

my **Data** → *my* **Information** → *our* **Knowledge** → *my* **Value**



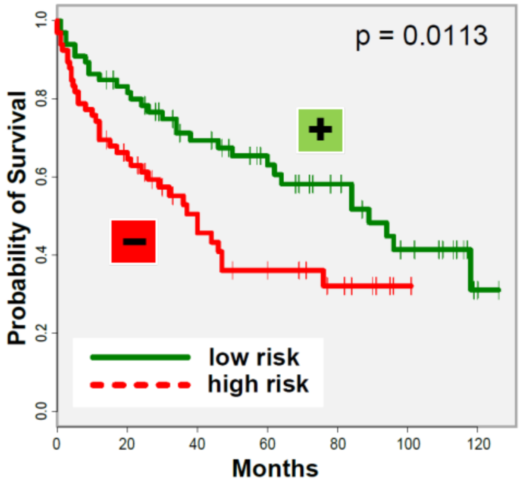




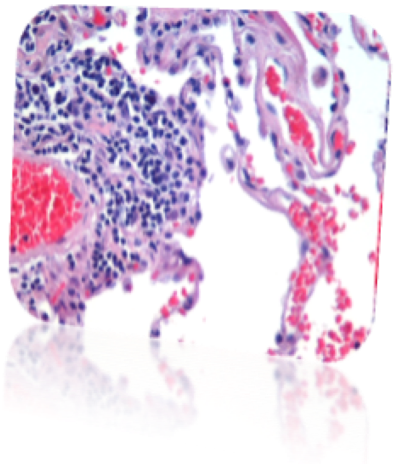


Fundamental data science questions - Which posterior distribution is encoder by algorithm $\mathcal{A}$?

$\mathbb{P}^{\mathcal{A}}\Big($



$\Big|$



$,$

$\Big)$

The Algorithm: Idiom of Modern Science

(Bernard Chazelle)

- Informally, an **algorithm** is any well-defined computational procedure, that takes some value as **input** and produces some value as **output**. (CLRS)
- **Analysis of algorithms**

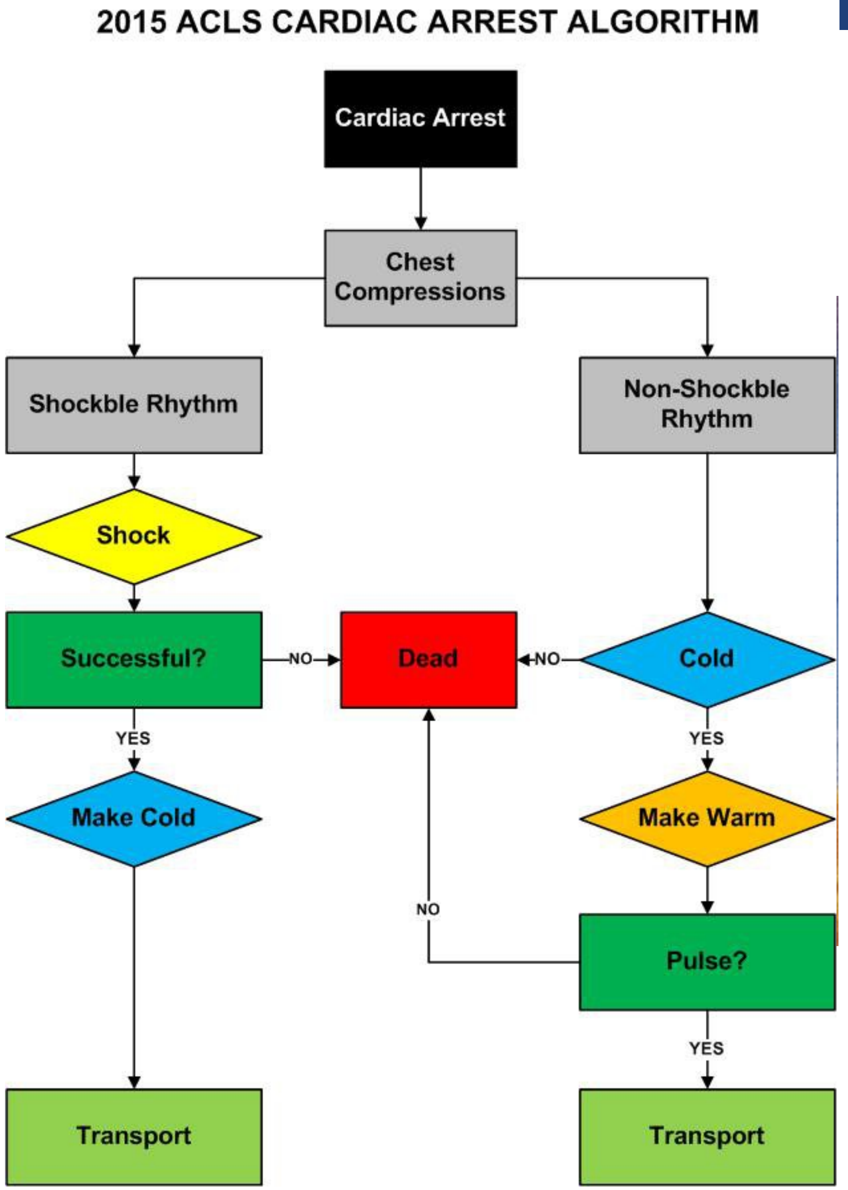
  Runtime

  Memory consumption

  ✗ **Robustness**

  ✗ **Generalization**

➢ **Learning algorithms „explore" a complex stochastic reality!**



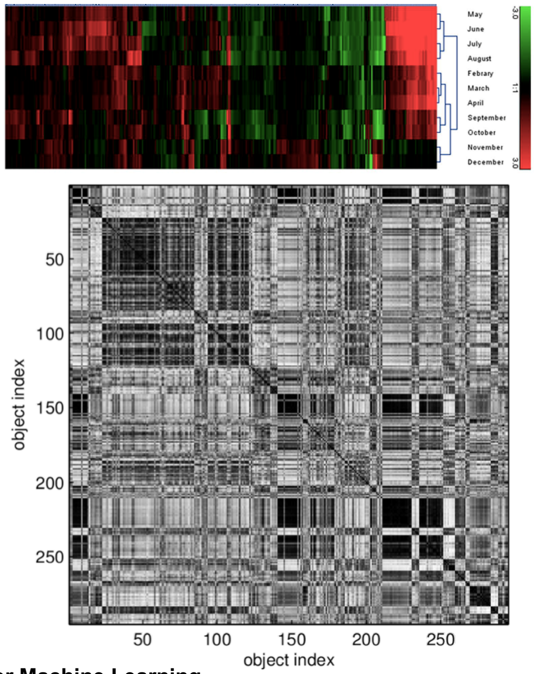
Roadmap

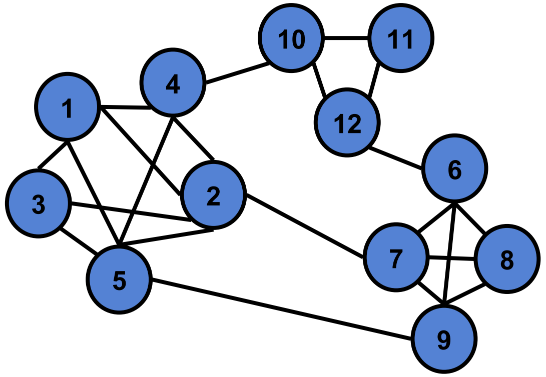
- **Algorithm design for Data Science**
  - What is the core problem? Lessons learned!
- **Algorithm validation** by information theory
  Learning optimal algorithms as open challenge!
- **Examples**
  - **Cortex parcellation**
  - **Sparse Minimum Bisection & Community Detection Problem**
- **Quo vadis – Artificial Intelligence?**

Algorithmics for Data Science – what is the problem?

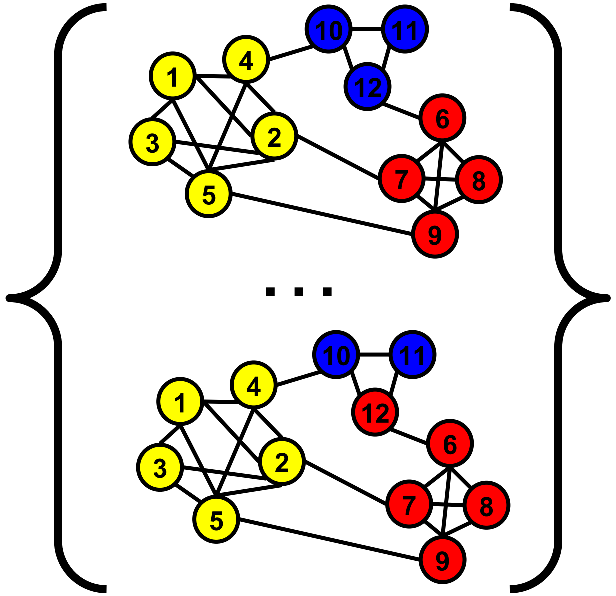
- **Random inputs imply random outputs**

$$\underbrace{\text{input } \mathbf{X}}_{\text{given}} \sim P(\mathbf{X}) \implies \underbrace{\mathcal{A}}_{\text{algorithm}} \implies \underbrace{\text{output } c \sim P(c|\mathbf{X})}_{\text{design}}$$





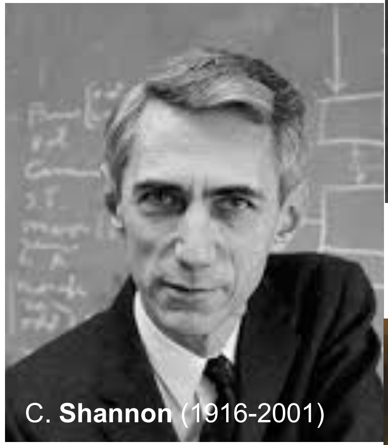
$\mathcal{A}$?



Core question for computer science:
How can we validate (data science) algorithms?

I. **Algorithms with random variables as input compute random variables as output!**
How can we prove correctness of such algorithms?


A. **Kolmogorov** (1903-1987)


C. **Shannon** (1916-2001)

II. **Algorithms** have to compute **typical solutions**!
What does this mean for algorithm design?

III. **When do algorithms generalize** **over noise/model mismatch?**


V. **Vapnik**
(1936 -)

IV. **How can algorithms autonomously improve performance?**

Typicality of solutions of random experiments

- Imagine the following **random coin flip experiment**
  $n = 1000$ coin flips of a biased coin $\forall i,\ P(\text{Head}) = P(\xi_i = 1) = p = 0.6$
- **Which sequence do you want to report?**
- **Minimizer of negative log-likelihood!**

$$\xi = \arg \min_{\xi_i \in \{0,1\}^n} \sum_{i=1}^{n} \Big(-\xi_i \log p - (1 - \xi_i) \log(1 - p) \Big)$$

$$= \underbrace{(1, 1, \ldots, 1)}_{1000 \text{ times}}$$

Machine Learning is not Optimization!

- **What you might want do, cannot be done** since we don't know $P(\mathbf{X})$.
- **What you can do, isn't most relevant** since it might yield atypical solutions.

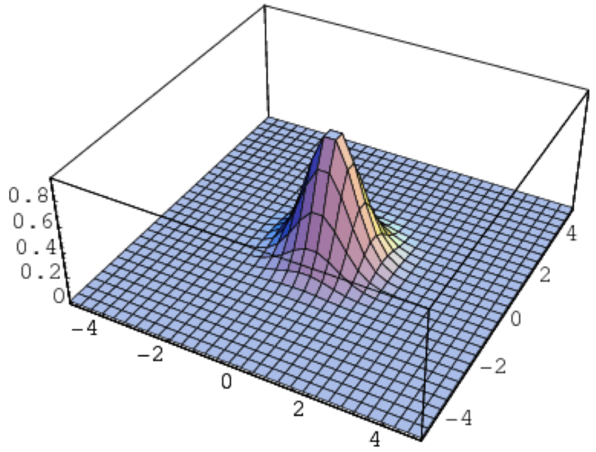
$$c^{\perp} \in \arg\min_{c \in \mathcal{C}} \mathbb{E}_{\mathbf{X}} R(c, \mathbf{X})$$

$$\hat{c}(\mathbf{X}') \in \arg\min_{\xi \in \mathcal{C}} R(c, \mathbf{X}')$$

- Machine **Learning algorithms localize solutions**!

We must **validate the metric** of the solution space

$$c \sim P_\theta(c|\mathbf{X}')$$



Model selection – What should we compare?

- **Standard setting**: Given are **training**, **validation** and **test** instance $\mathbf{X}', \mathbf{X}'', \mathbf{X}'''$.

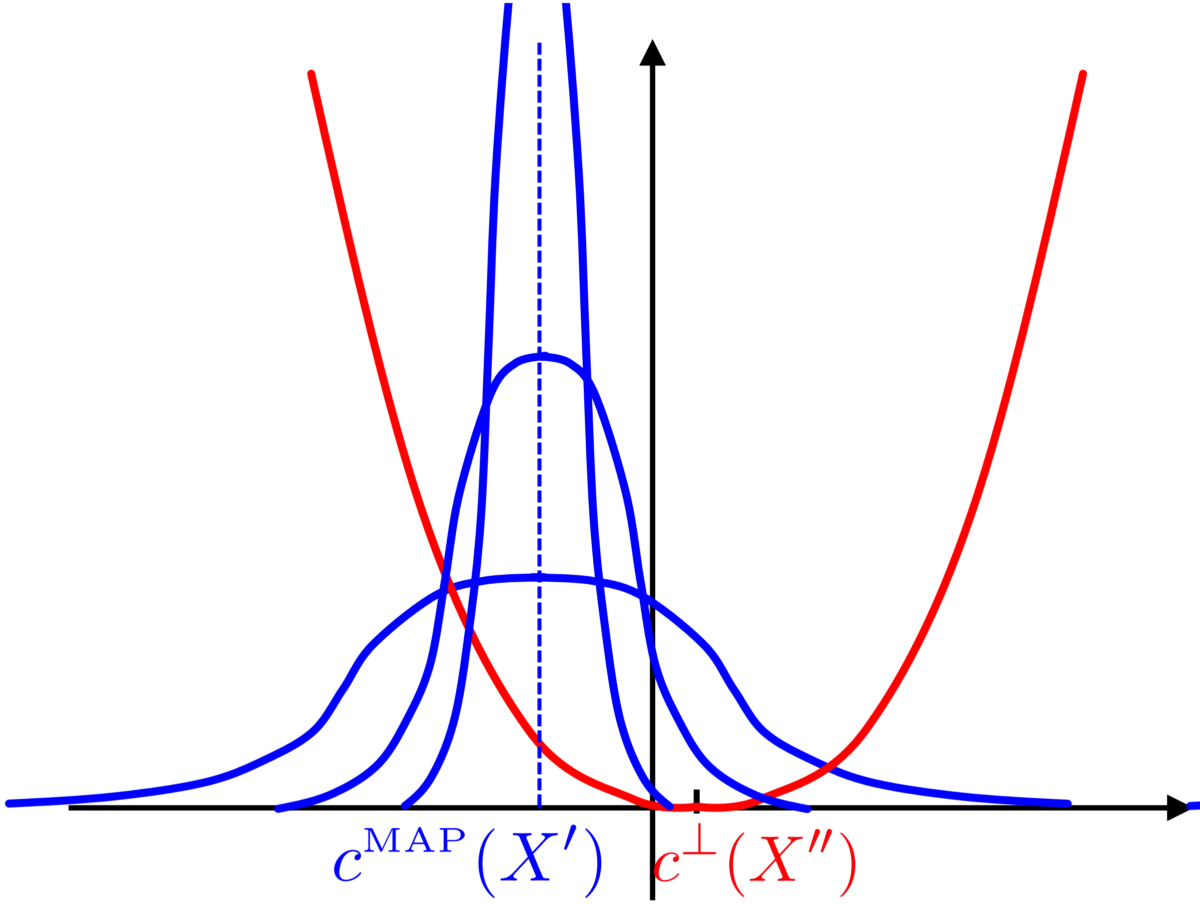
  We consider a set of possible models (risks) $\{R^1, \ldots, R^K \;:\; R : \mathcal{C} \times \mathcal{X} \to \mathbb{R}_+\}$

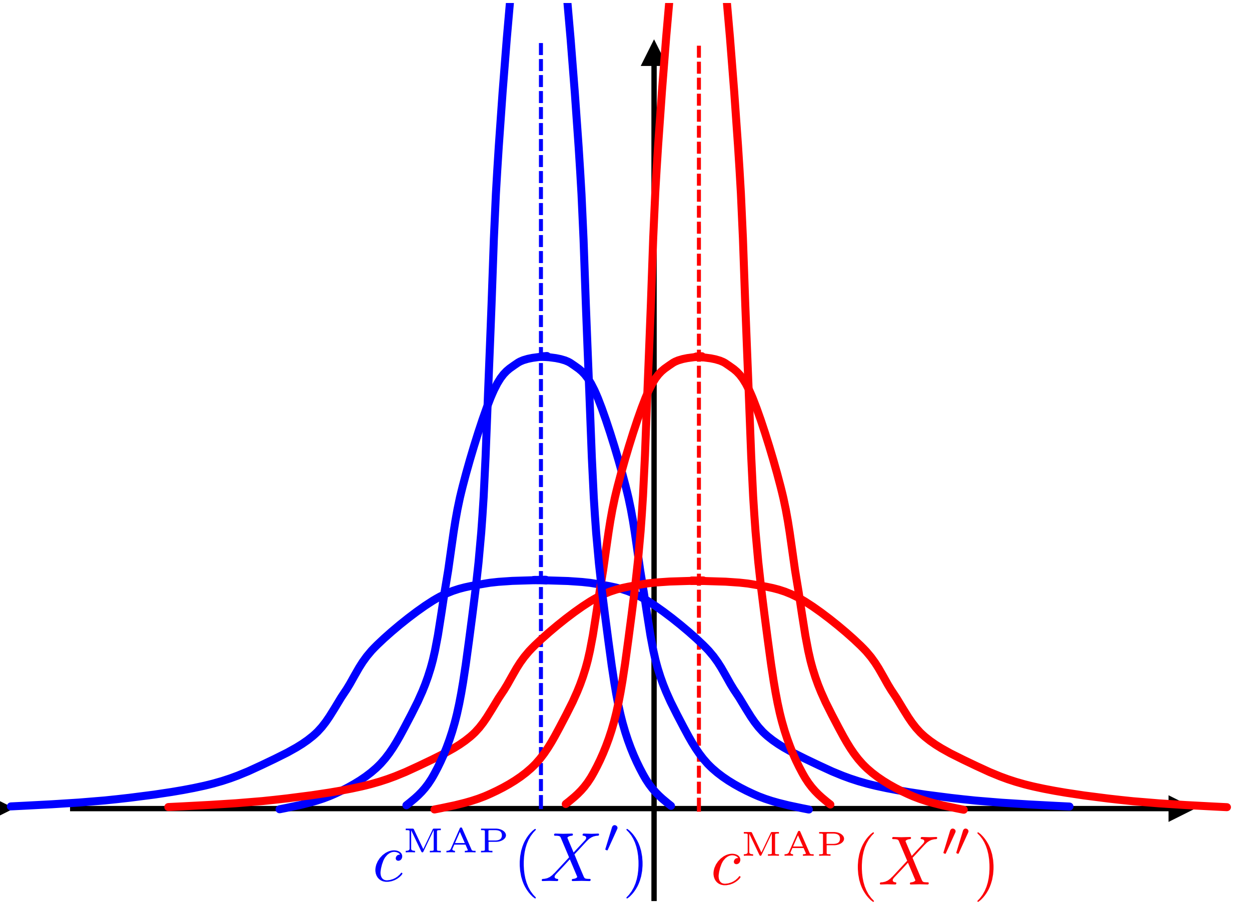
  Select the model $R^\star(.,.)$ with the lowest validation error on the training solutions

$$R^\star(.,.) = \arg \min_{1 \le \alpha \le K} \sum_{c \in \mathcal{C}} p(c|\mathbf{X}')\, R^\alpha(c, \mathbf{X}'')$$

- *Standard view:* **„Machine Learning is stochastic optimization“ of risk**:
  - Different risks with the same global minimum yield significantly different solutions under uncertainty, i.e., when the input contains noise.
- **Modeling wisdom**: Use small numbers when you encounter large uncertainties!

Risk Minimization versus Score Maximization

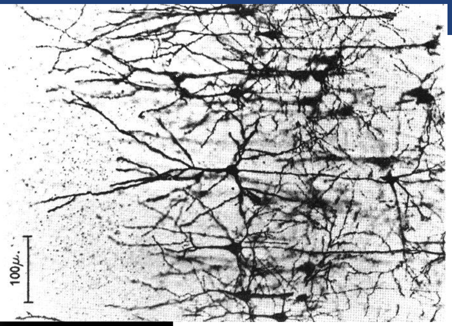




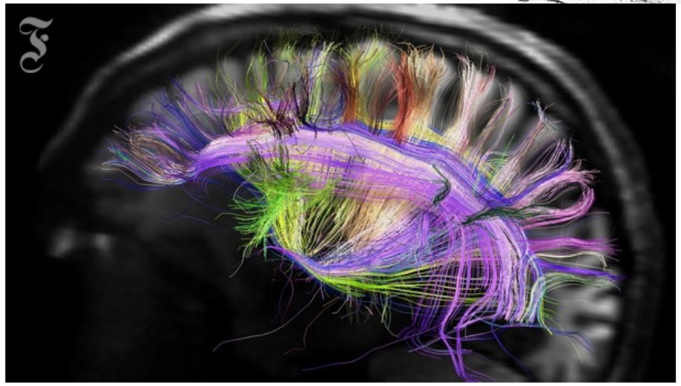
$$\text{validation error} = \sum_{c \in \mathcal{C}} P(c|X')\, R(c, X'')$$

$$k(X', X'') = \sum_{c \in \mathcal{C}} P(c|X')\, P(c|X'')$$

Learning machines master algorithmic induction and «imitate» humans

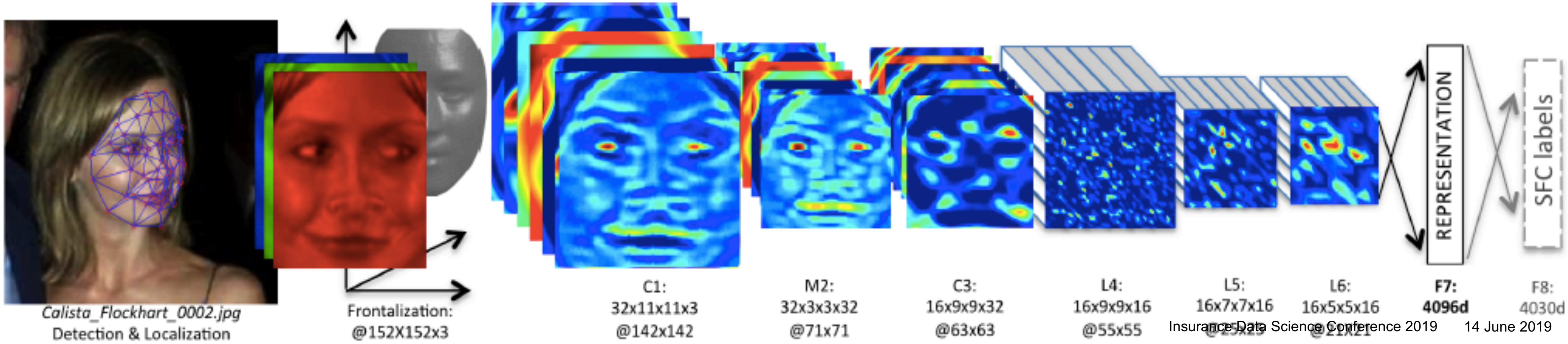


- **Biological neural networks** are adaptive and can learn.
- **Artificial neural networks** mimic these learning capabilities.



Neural networks visualized by brain scans. © VAN WEDEEN

- **DeepFace** network of FaceBook



ETH zürich

“Deep Network” Halluzinationen

(Courtesy of **Sebastian Nowozin, 2016**)

Image interpolation with neural networks

What is missing?

The Scientific Method

Step 1: **Ask questions**

A ?

Step 2:
Propose hypotheses

A ?

Step 3:
Conduct experiment

A ?

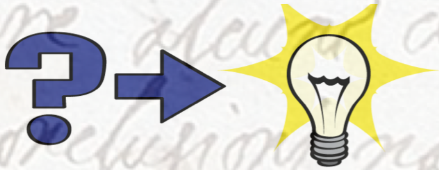
Step 4: **Analyze results**

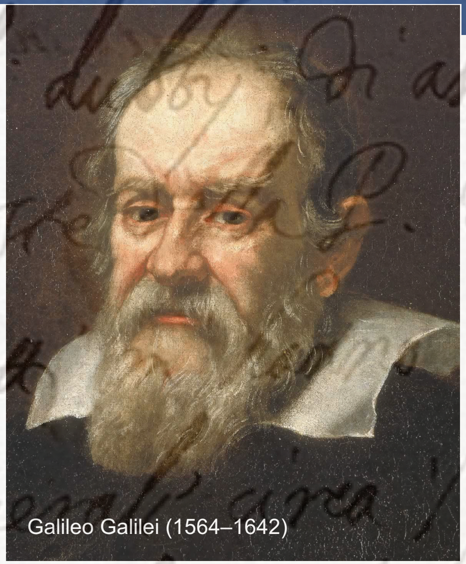
A ?

Step 5: Draw conclusions –
postulate a theory

A ?

A ?





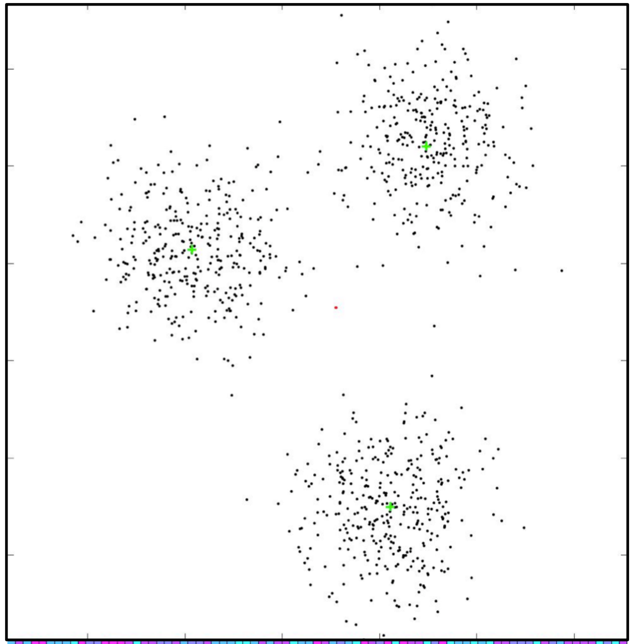
Galileo Galilei (1564–1642)

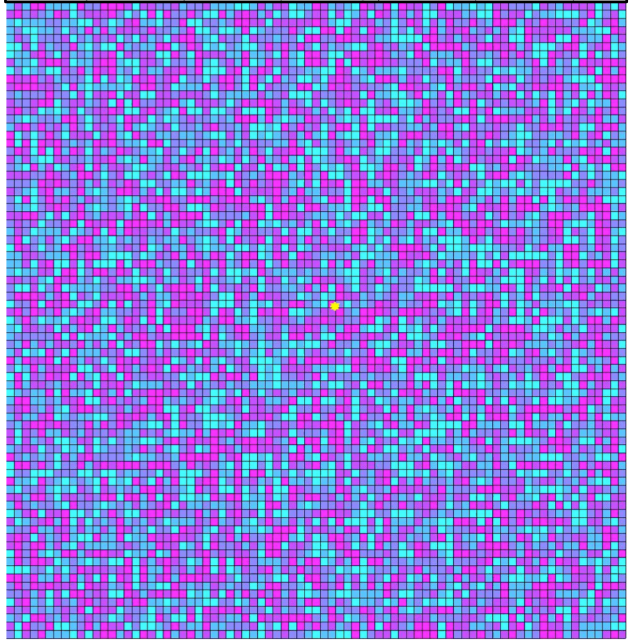
Gibbs distributions for optimization

- **Given a risk / cost function** $R : \mathcal{C} \times \mathcal{X} \to \mathbb{R}$
- **Gibbs posteriors maximize entropy** for expected costs $\mathbb{E}_{c|\mathbf{X}}\left[R(c, \mathbf{X})\right]$!

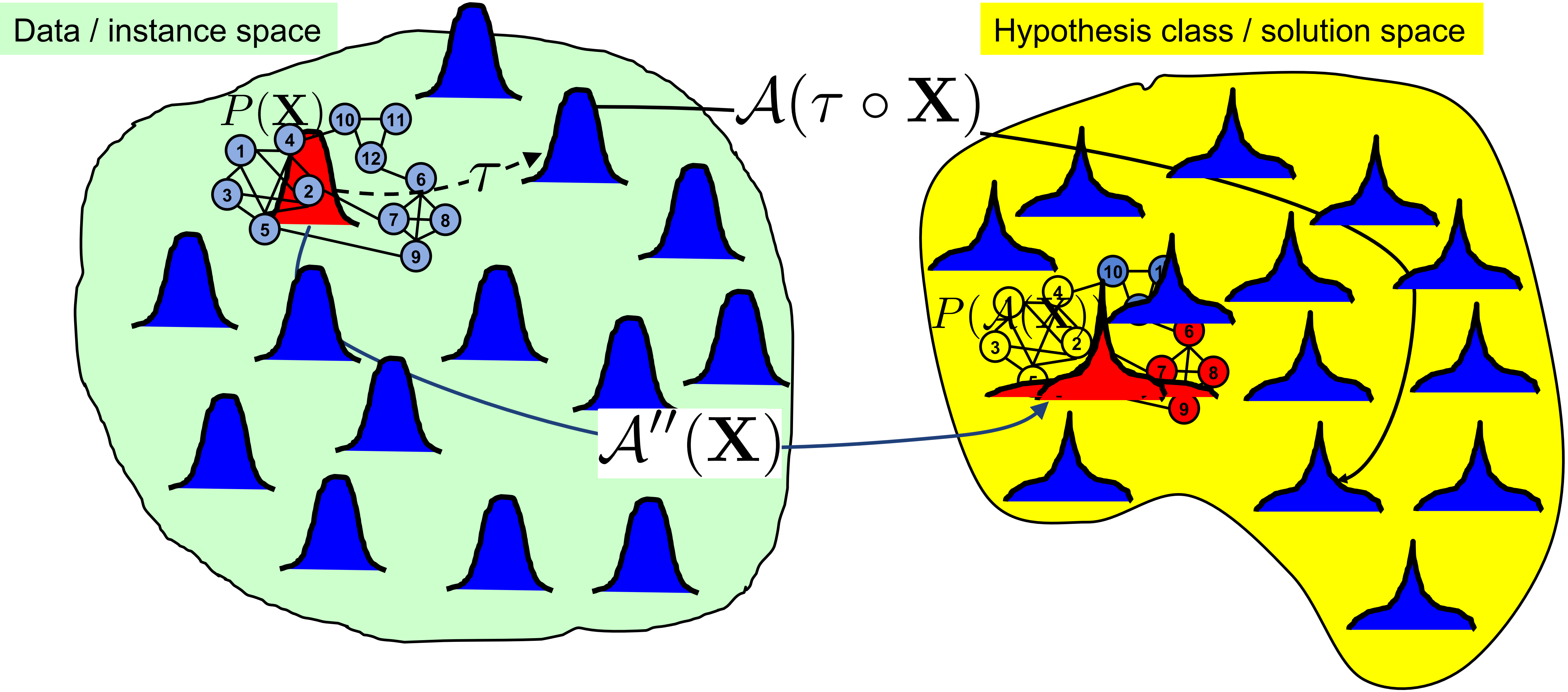
$$P_t(c|\mathbf{X}) = \frac{\exp\big(-\beta_t R(c, \mathbf{X})\big)}{\sum_{c' \in \mathcal{C}} \exp\big(-\beta_t R(c', \mathbf{X})\big)}$$

- **Robustness** by maximum entropy
- **Annealing**: increase iteratively β_t during algorithm execution

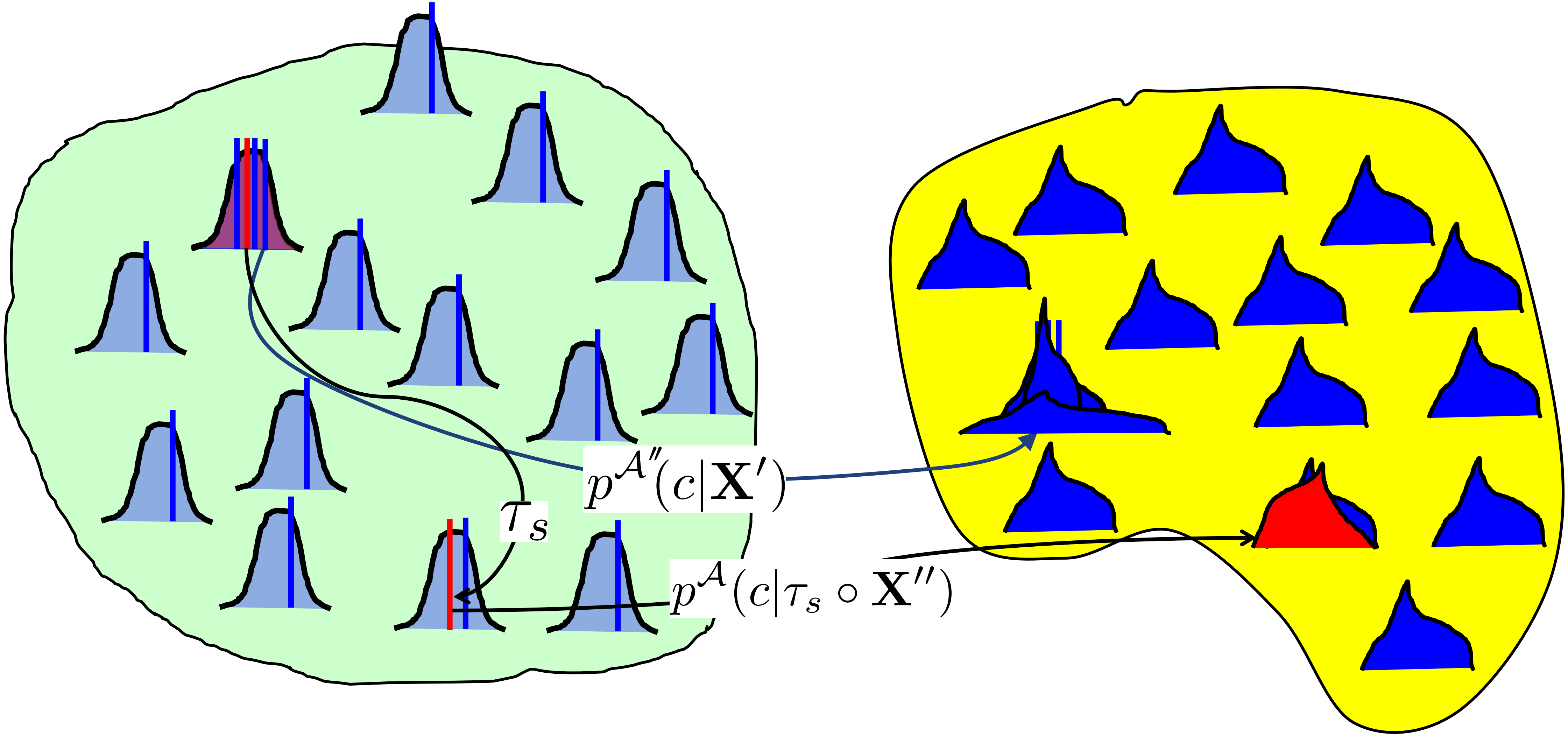




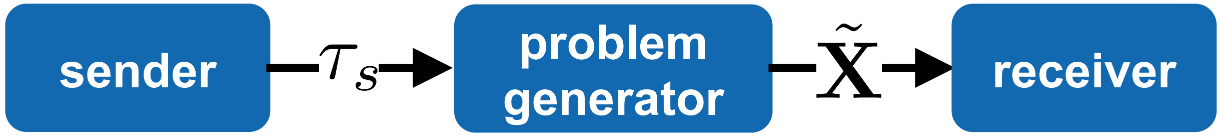
Noisy inputs of algorithms quantize hypothesis classes



Quantized hypothesis classes based on instances



Communication process and decoding



- Sender sends transformation τ_s
- Receiver accepts instance $\tilde{\mathbf{X}} := \tau_s \circ \mathbf{X}''$ with $\mathbf{X}', \mathbf{X}'' \sim P(\mathbf{X})$ and decodes the transformation by **maximizing expected posterior**

$$\hat{\tau} \in \arg\max_{\tau \in \mathcal{T}} \mathbb{E}_{c|\tau \circ \mathbf{X}'} \left(c | \tau_s \circ \mathbf{X}'' \right)$$

- **Error** events are decisions with $\hat{\tau} \neq \tau_s$

=> Calculate probability $P(\hat{\tau} \neq \tau_s | \tau_s)$

Error probability $P(\hat{\tau} \neq \tau_s | \tau_s)$

- Estimate error given random transformations $\tau \in \mathcal{T}$ and test data $\tilde{\mathbf{X}} := \tau_s \circ \mathbf{X}''$

$$k^{\mathcal{A}}(\mathbf{X}', \mathbf{X}'') := \sum_{c \in \mathcal{C}(\mathbf{X}'')} p^{\mathcal{A}}(c|\mathbf{X}')\, p^{\mathcal{A}}(c|\mathbf{X}'')$$

$$\begin{aligned}
\mathbb{P}\left(\hat{\tau} \neq \tau_s | \tau_s\right) &= \mathbb{P}\left(\max_{j \neq s} \mathbb{E}_{c|\tau_j \circ \mathbf{X}'} p^{\mathcal{A}}(c|\tilde{\mathbf{X}}) > \mathbb{E}_{c|\tau_s \circ \mathbf{X}'} p^{\mathcal{A}}(c|\tilde{\mathbf{X}}) \Big| \tau_s\right) \\
&\leq \sum_{j \neq s} \mathbb{P}\left(\mathbb{E}_{c|\tau_j \circ \mathbf{X}'} p^{\mathcal{A}}(c|\tilde{\mathbf{X}}) > k^{\mathcal{A}}(\mathbf{X}', \mathbf{X}'') \Big| \tau_s\right) \\
&\leq M\, \mathbb{P}\left(\mathbb{E}_{c|\tau_{\neq s} \circ \mathbf{X}'} p^{\mathcal{A}}(c|\tilde{\mathbf{X}}) > k^{\mathcal{A}}(\mathbf{X}', \mathbf{X}'') \Big| \tau_s\right) \\
&\leq M\, \mathbb{E}_{\mathbf{X}', \mathbf{X}''} \frac{\mathbb{E}_{\tau_{\neq s}} \mathbb{E}_{c|\tau_{\neq s} \circ \mathbf{X}'}\, p^{\mathcal{A}}(c|\tilde{\mathbf{X}})}{k^{\mathcal{A}}(\mathbf{X}', \mathbf{X}'')}
\end{aligned}$$

Generalization capacity from typicality

Theorem: Asymptotic error free ($\lim_{n\to\infty} P(\hat{\tau} \neq \tau_s|\tau_s) = 0$) *identification* of *hypotheses* is achievable if

$$
\begin{aligned}
P\left(\hat{\tau} \neq \tau_s|\tau_s\right) \;&\leq\; \exp\Big(-(\mathcal{I} - \log M)\Big) \;\to\; 0 \qquad \text{with} \\
\mathcal{I} \;&=\; \mathbb{E}_{\mathbf{X}',\mathbf{X}''} \log\Big(|\mathcal{C}|k^{\mathcal{A}}(\mathbf{X}',\mathbf{X}'')\Big) \\
k^{\mathcal{A}}(\mathbf{X}',\mathbf{X}'') \;&=\; \sum_{c\in\mathcal{C}} p^{\mathcal{A}}(c|\mathbf{X}')p^{\mathcal{A}}(c|\mathbf{X}'') \in [0,1]
\end{aligned}
$$

Learning algorithms localize typical solutions

- "**Posteriors**" for probable data $\mathbf{X}', \mathbf{X}''$ should agree!

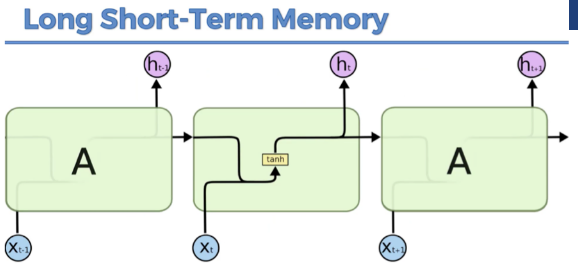
$$k^{\mathcal{A}}(\mathbf{X}', \mathbf{X}'') \;=\; \sum_{c \in \mathcal{C}} p^{\mathcal{A}}(c|\mathbf{X}')\, p^{\mathcal{A}}(c|\mathbf{X}'') \in [0,1]$$

A too broad or too narrow posterior $p^{\mathcal{A}}(.|\mathbf{X})$ yields a small kernel value $k^{\mathcal{A}}(\mathbf{X}', \mathbf{X}'')$! Optimize width of $p^{\mathcal{A}}(.|\mathbf{X})$.

- **Optimal posterior**

$$P^{\star} \in \arg\max_{t} \mathbb{E}_{\mathbf{X}', \mathbf{X}''} \log\left(|\mathcal{C}| k_t^{\mathcal{A}}(\mathbf{X}', \mathbf{X}'')\right)$$

Learning an algorithm: open challenge!



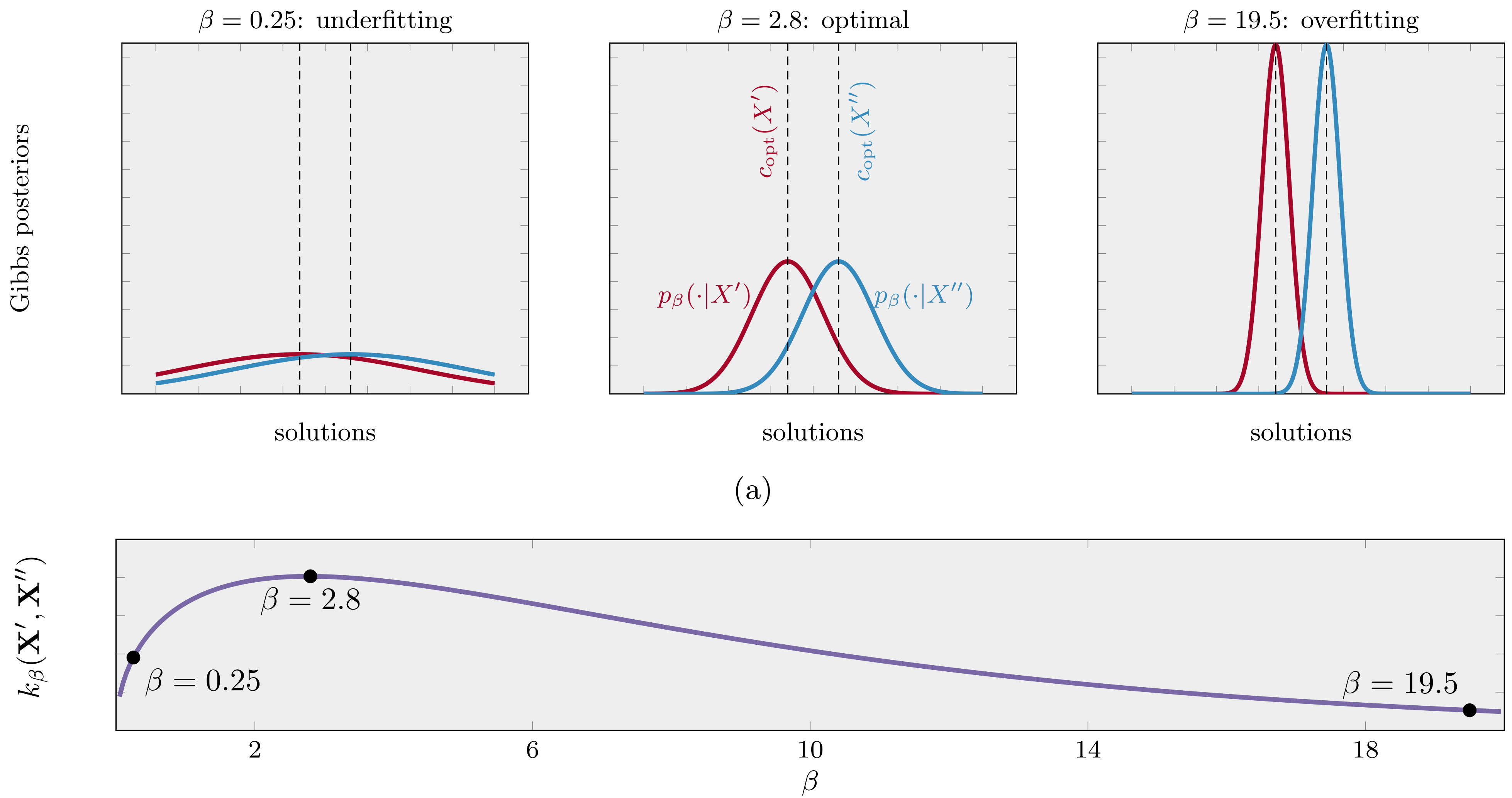
- **Given a set of algorithms** $\left\{\mathcal{A}^{(\alpha)}(\mathbf{X}) = \langle P_0^{(\alpha)}(c|\mathbf{X}), \ldots, P_{t^\star}^{(\alpha)}(c|\mathbf{X})\rangle\right\}$
- **Select posterior** $\mathcal{A}^{(\alpha)}(\mathbf{X})$ s.t. generalization capacity is maximized

$$P^\star \in \arg\max_{\{\mathcal{A}\}} \max_t \mathbb{E}_{\mathbf{X}',\mathbf{X}''} \log\left(|\mathcal{C}|k_t^{\mathcal{A}}(\mathbf{X}',\mathbf{X}'')\right)$$

- **Problem**: We cannot evaluate $\mathbb{E}_{\mathbf{X}',\mathbf{X}''} \log \ldots$ since $P(\mathbf{X}',\mathbf{X}'')$ is unknown!
- **Statistical Learning Theory**: bound expectation by sample average

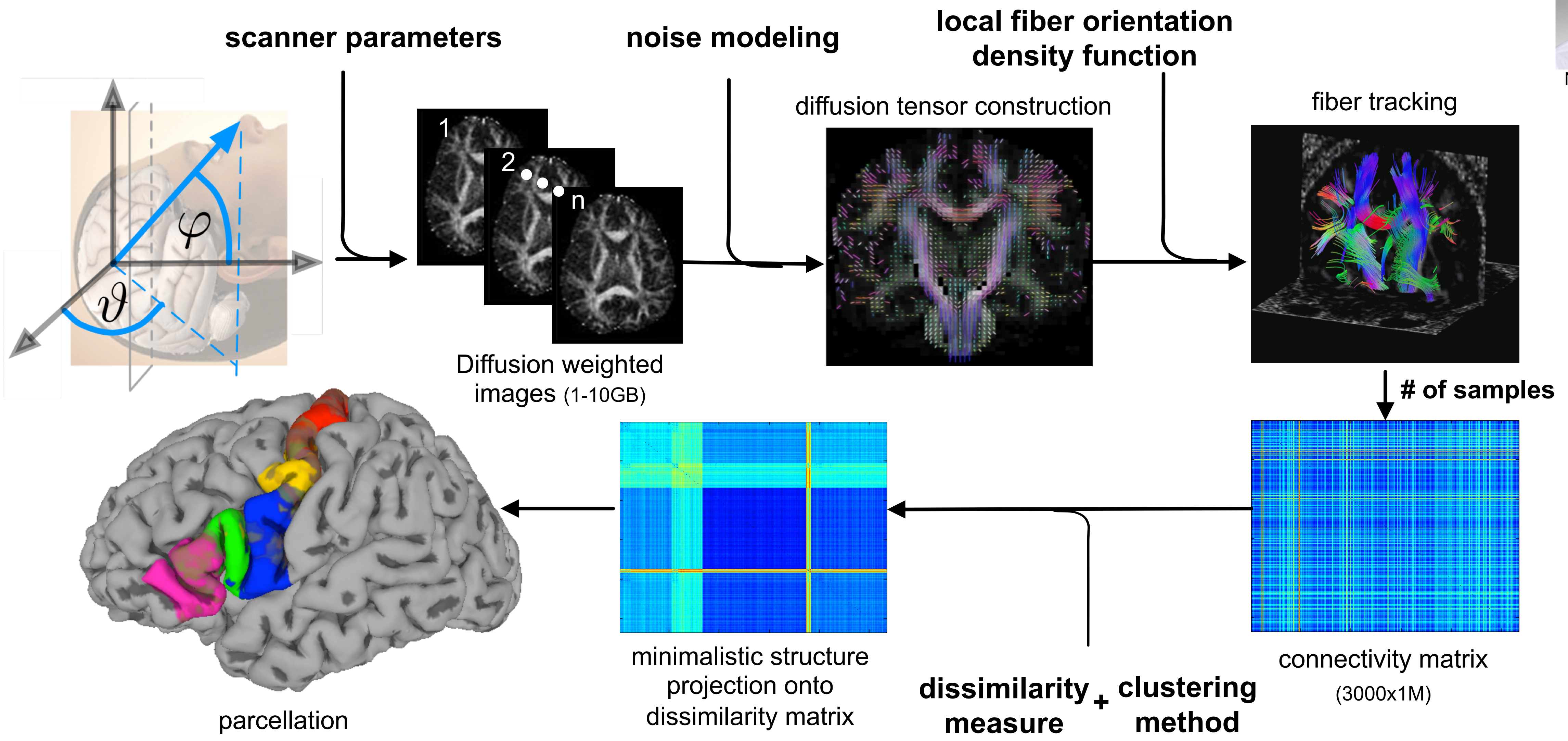
$$\mathbb{E}_{\mathbf{X}',\mathbf{X}''} \log\left(|\mathcal{C}|k_t^{\mathcal{A}}(\mathbf{X}',\mathbf{X}'')\right) \geq \frac{1}{L}\sum_{l\leq L} \log\left(|\mathcal{C}|k_t^{\mathcal{A}}(\mathbf{X}_l',\mathbf{X}_l'')\right) - \text{penalty}$$

Maximal score at finite β



Cortex Parcellation with diffusion weighted tensor imaging

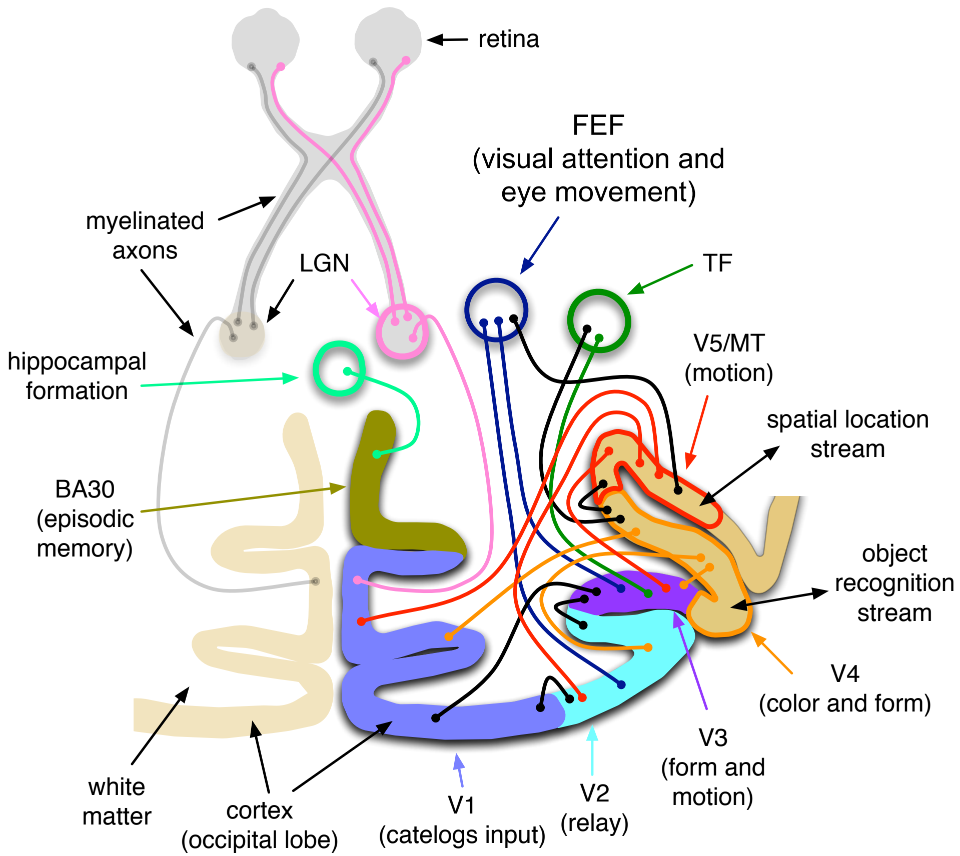

Nico Gorbach

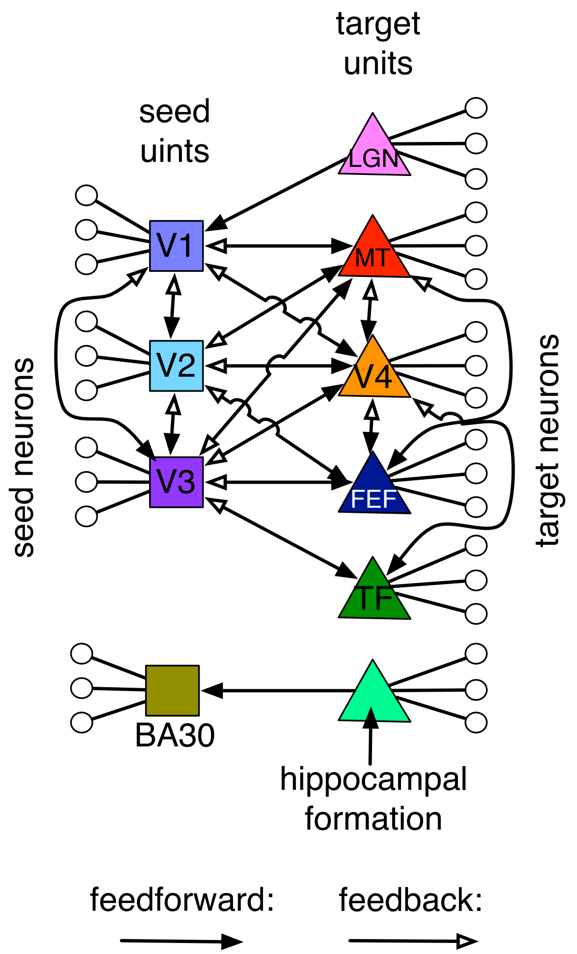


Systems Neuroscience

Subset of the visual system in the macaque monkey.

Target connections are limited for illustration purposes.

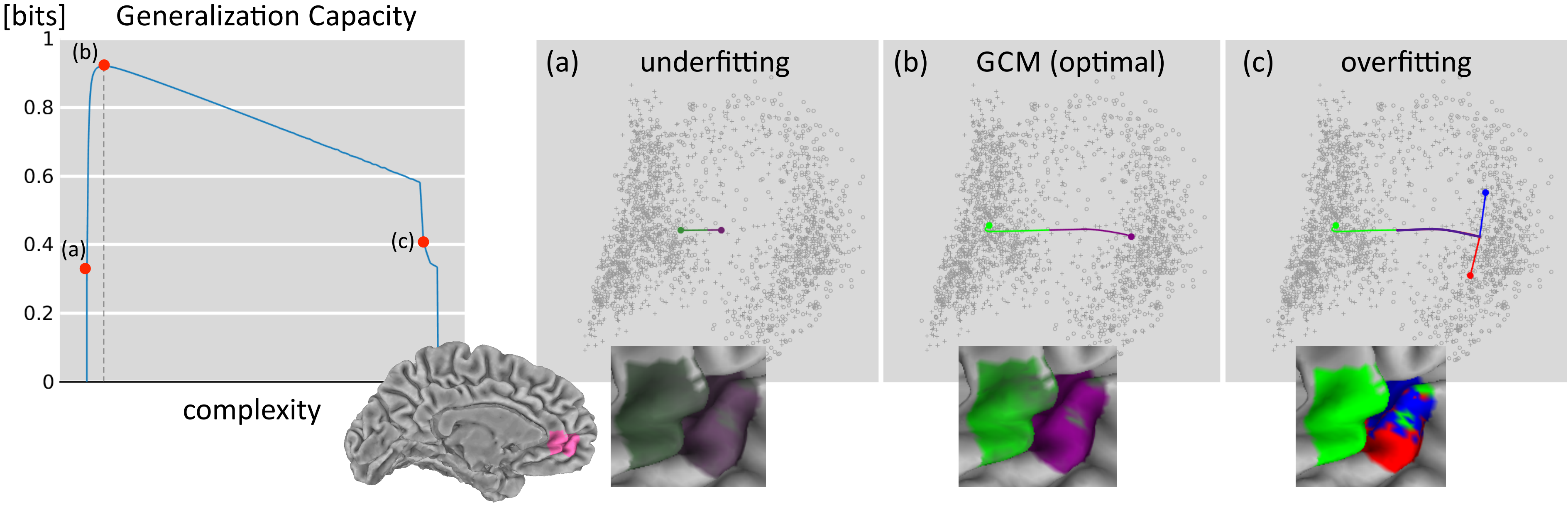




- The brain is considered as an ensemble of functionally specialized units coupled together in a modulatory fashion (Friston, 2002).

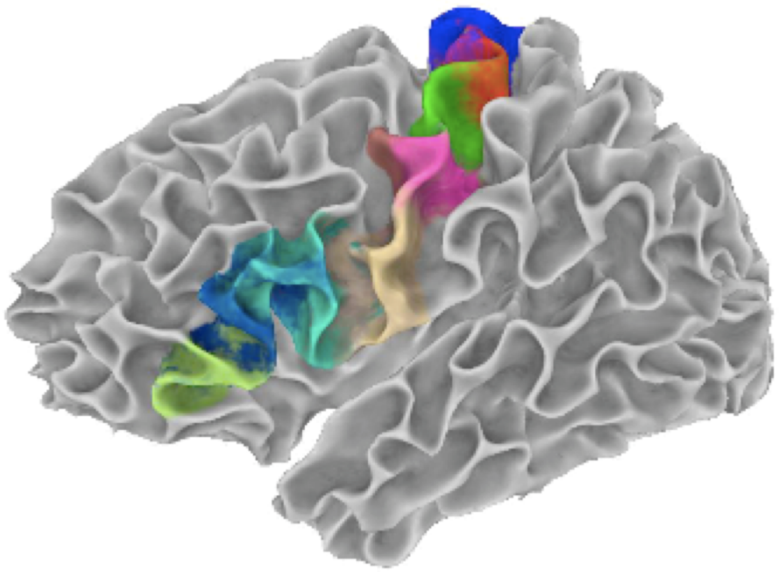
Under- and overfitting in parcellation

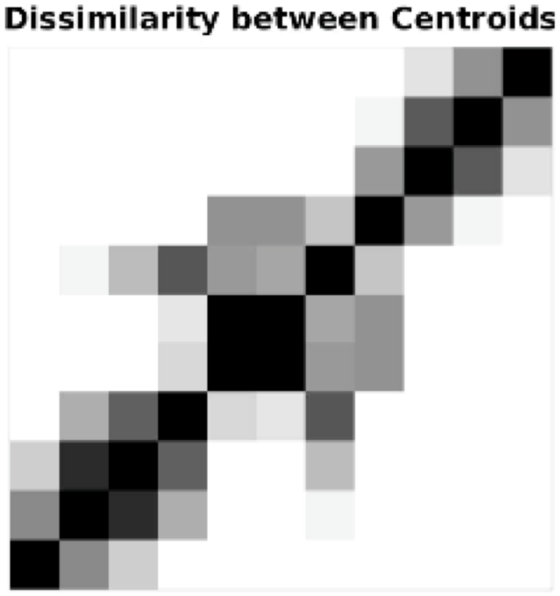
- Connectivity of two brain regions is analyzed
- Generalization capacity maximizer (GCM) outperforms empirical risk minimizer

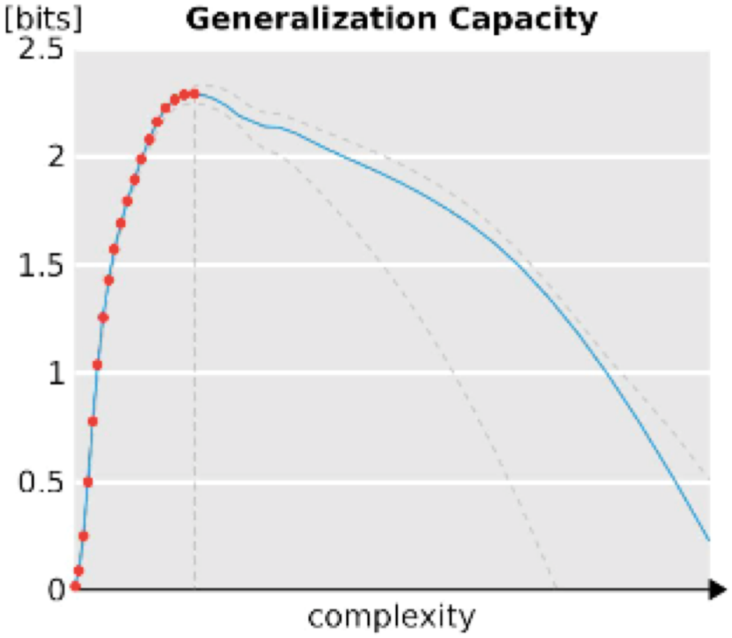


Dynamics of cortex parcellation

- Start at low resolution
- Estimate parcellations with higher resolution
- Stop at maximal generalization capacity







What is missing?

The scientific method

Step 1: **Ask questions**

$\mathcal{A}$?

Step 2:
Propose hypotheses

$\mathcal{A}$?

Step 3:
Conduct experiments

$\mathcal{A}$?

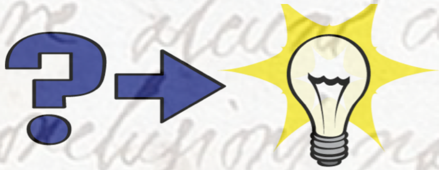
Step 4: **Analyze results**

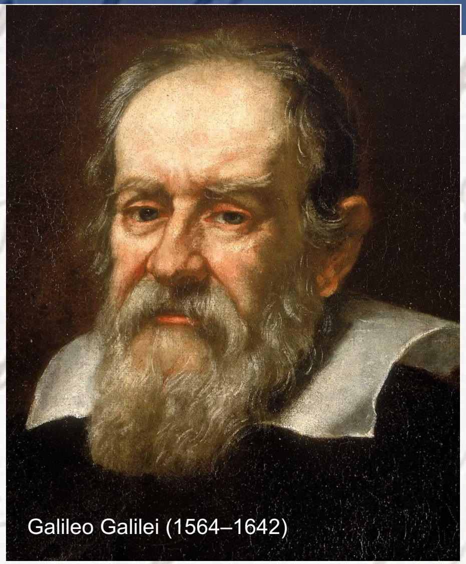
$\mathcal{A}$?

Step 5: Draw conclusions –
postulate a theory

$\mathcal{A}$?

$\mathcal{A}$?




Galileo Galilei (1564–1642)

Roadmap

- **Algorithm design for Data Science**
  - What is the core problem? Lessons learned!
- **Algorithm validation** by information theory
  Learning optimal algorithms as open challenge!
- **Examples**
  - **Cortex parcellation**
  - **Sparse Minimum Bisection & Community Detection Problem**
- **Quo vadis – Artificial Intelligence?**

Outlook and Lessons learned

1. **Algorithms are models of posteriors and localize in solution spaces.**
2. **Learning requires validation of algorithms,** not “only” verification.
3. Conditioned on inputs, **algorithms** are characterized by a **generalization capacity, i.e., an optimal resolution of the hypothesis class!**

$\Rightarrow$ **structure specific information** in data.

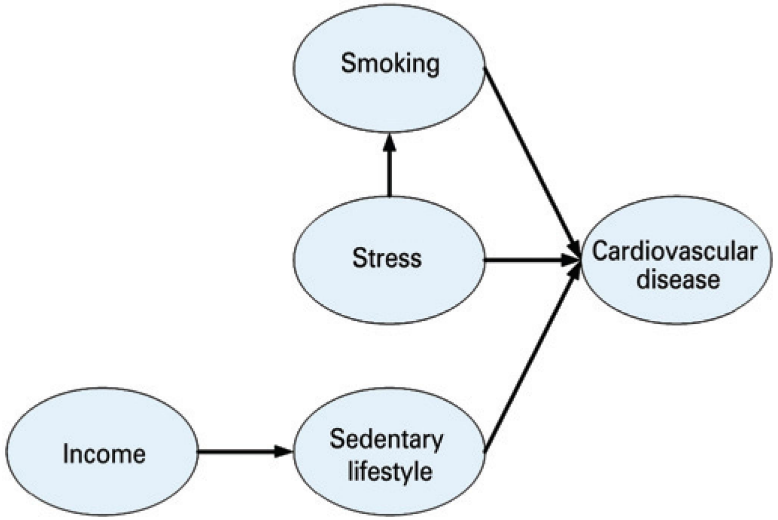
$\Rightarrow$ Relate ***statistical*** **complexity** to ***computational*** **complexity**!

Expert systems enable Artificial Intelligence !?

(Strategy of AI researchers in 60th to 80th)

Intelligent behavior as a programming problem

+ **Inference** by **rule systems** and **logic calculus**
– **Problem**: *Knowledge Engineering* via experts



- **Experts invent symbols**
- **Learning algorithms discover relations**, i.e. conditional probabilities