ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

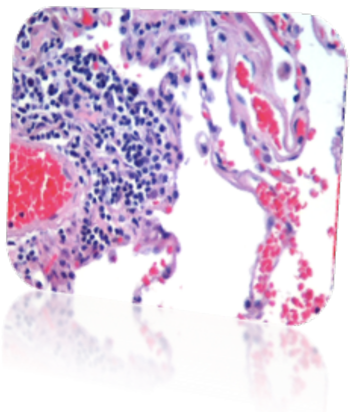
inf | Informatik
Computer Science

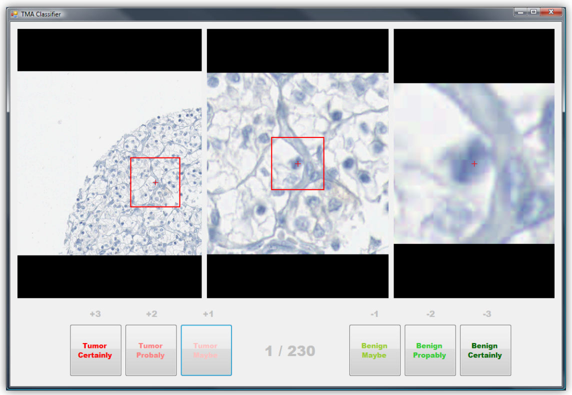
Information theory of algorithms

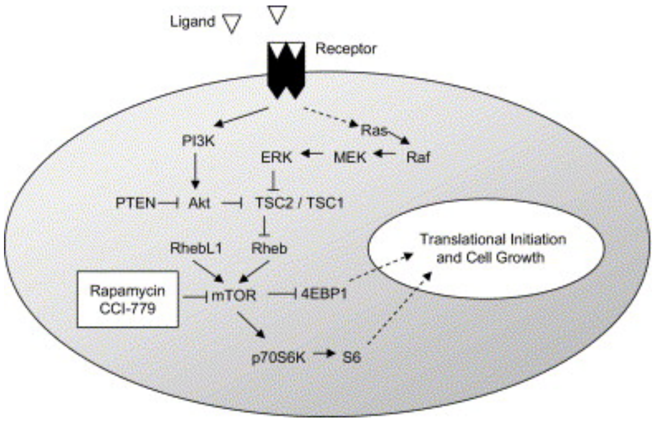
Joachim M. Buhmann

Computer Science Department, ETH Zurich

IT value for personalized medicine

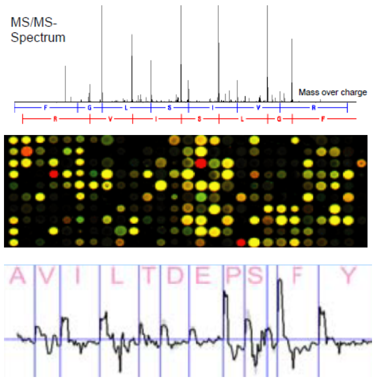


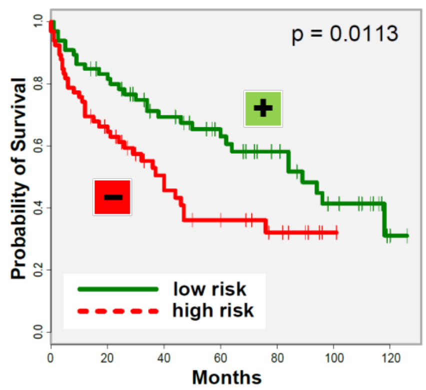




Activation of the mTOR Signaling Pathway in Renal Clear Cell Carcinoma. Robb et al., J Urology 177:346 (2007)

my **Data** → *my* **Information** → *my* **Knowledge**





my **Value**

Roadmap

- **Robust computation versus learning**
  Measuring the information content of algorithms
- **Algorithm/Model validation** by information theory
- **Learning optimal algorithms**: open challenge!
  - Validating approximate spanning trees
  - Graph cut for gene expression analysis
  - Robust sorting

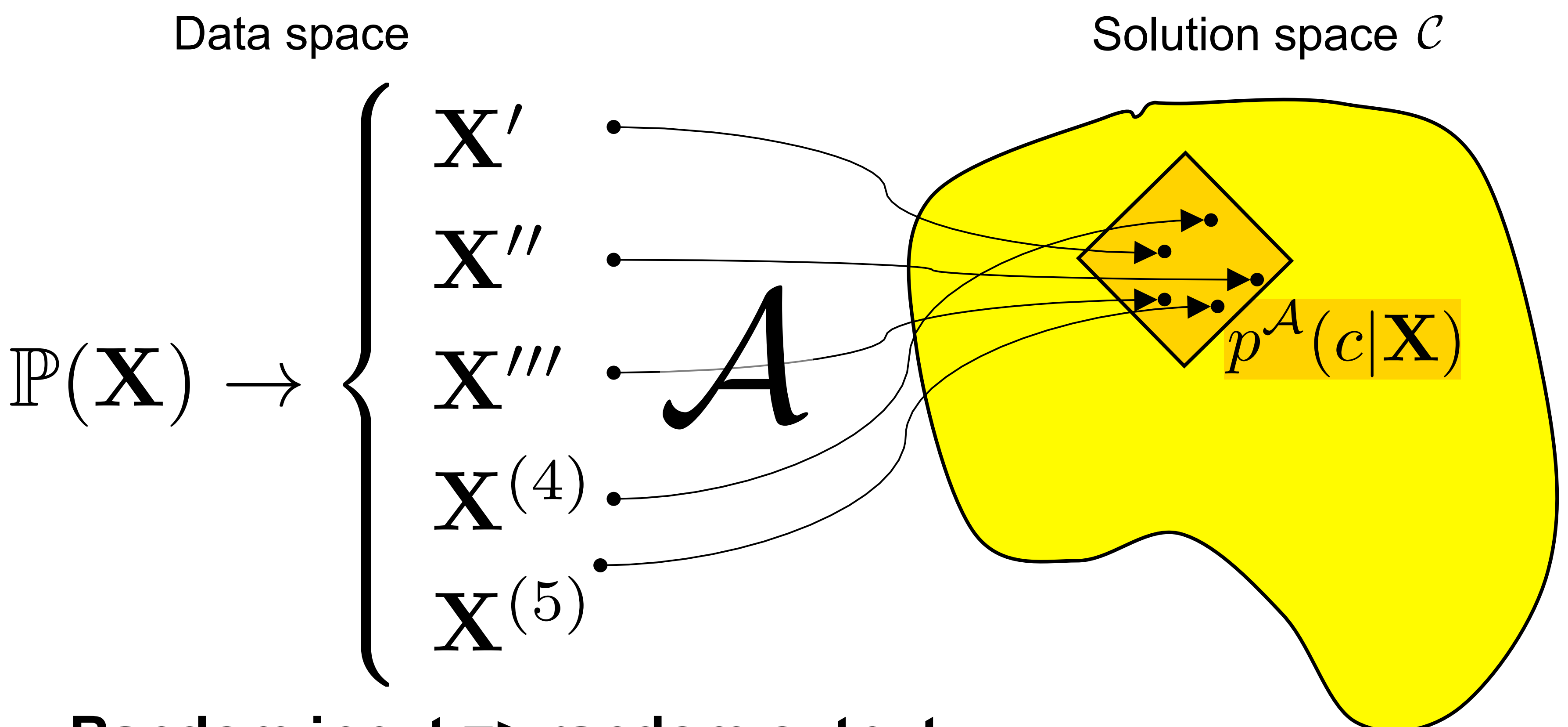
What is an algorithms?

- Informally, an **algorithm** is any well-defined computational procedure that takes some value(s) as **input** and produces some value(s) as **output** (CLRS' 01)
- **Classical view**: algorithm $\mathcal{A}$ maps (stochastic) data (input) to a solution / hypothesis (output).

$$\text{input } \mathbf{X} \to \mathcal{A} \to \text{output } c \in \mathcal{C}$$

Challenge of robust algorithm design

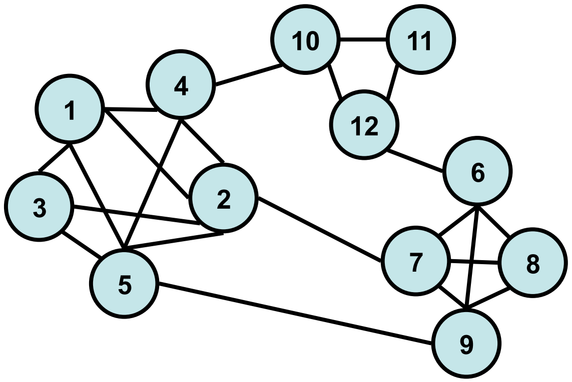
Algorithmic processing: $\text{input } \mathbf{X} \to \mathcal{A} \to \text{output } c \in \mathcal{C}$

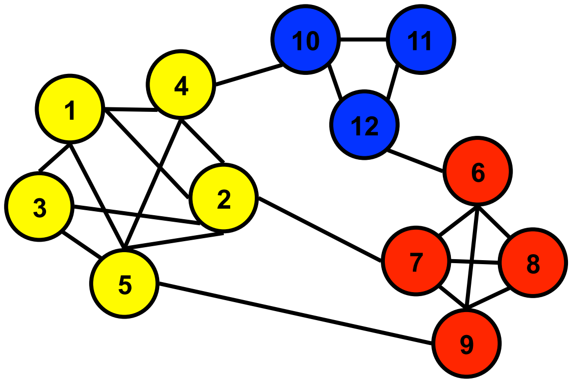


Random input => random output

Core problem in discriminative learning?

- Often, the data space is much larger than the solution space





Space of graphs $\mathfrak{G}_n = \{(\mathcal{V}, \mathcal{E}, \mathcal{W})\}$ with n vertices

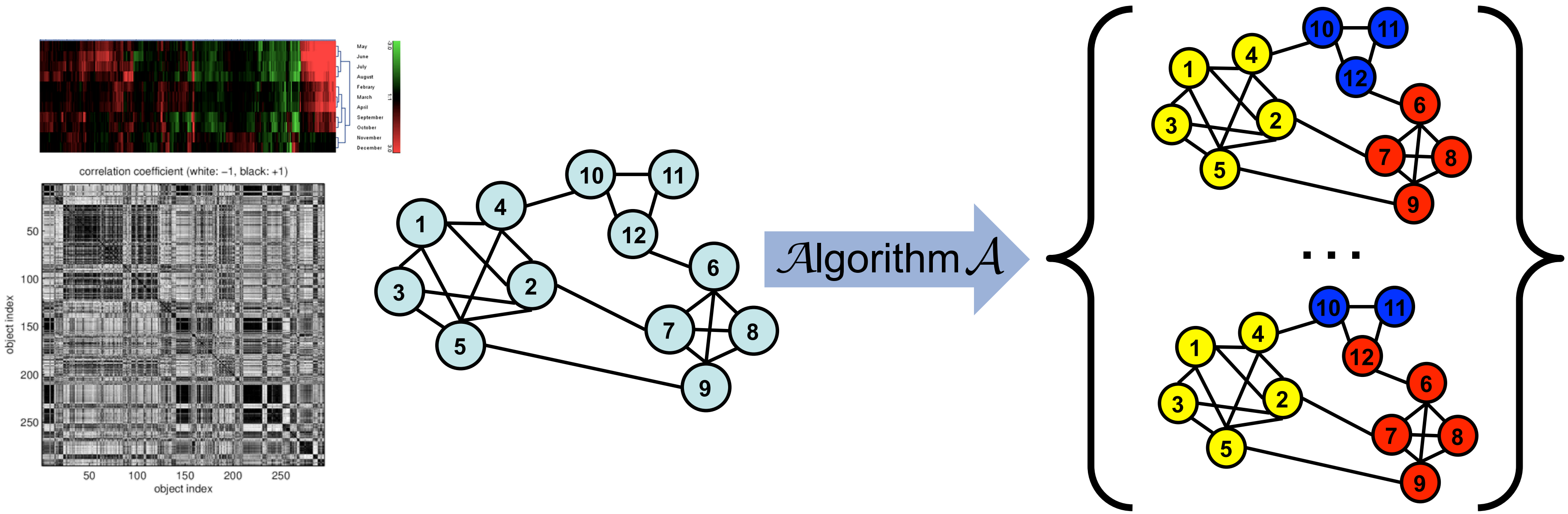
$$|\mathfrak{G}_n| = \mathbb{R}^{\binom{n}{2}}$$

Space of k colorings $\mathcal{C}$

$$|\mathcal{C}| \leq k^n$$

What is learning?

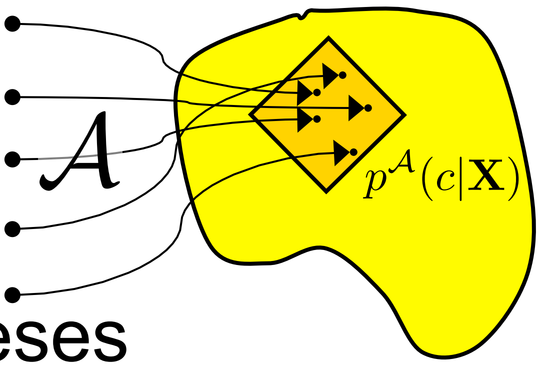
- **Given**: data $\mathbf{X} \sim \mathbb{P}(\mathbf{X})$ and a hypothesis class $\mathcal{C}$



- **Modeling in data analysis** requires
  - **quantization**: given $\mathcal{A}$, identify a set of *good* hypotheses;
  - **learning**: find an $\mathcal{A}$ that specifies an informative set!

Robust algorithms

- For **stochastic input**, an algorithm $\mathcal{A}$ returns a **stochastic output**; we interpret $\mathcal{A}$ as a trajectory in the powerset of the solution space $\mathcal{C}$.



- **Discriminative probabilistic view**: algorithm $\mathcal{A}$ maps stochastic data (input) to a set of solutions / hypotheses $A_t(\mathbf{X})$ (output) still considered at time t
  input $(\mathbf{X}, t^*) \to \mathcal{A} \to$ output posterior $p^{\mathcal{A}}(c|\mathbf{X})$

Algorithms as sets of feasible solutions

- Let X denote data and $A_t(\mathbf{X})$ the set of feasible solutions at iteration t

$$
\begin{aligned}
\text{algorithm } \mathcal{A}(\mathbf{X}) &= \langle A_0(\mathbf{X}), \ldots, A_T(\mathbf{X}) \rangle, \\
\mathbf{init}\ \ A_0(\mathbf{X}) &= \mathcal{C}, \\
A_t(\mathbf{X}) &\subseteq \mathcal{C}, \quad 0 \le t \le T, \\
\mathbf{return}\ \ A_T(\mathbf{X}) &= \{c^\perp(\mathbf{X})\}.
\end{aligned}
$$

- Monotonically contractive algorithms

$$\mathcal{A}(\mathbf{X}) = \langle A_0(\mathbf{X}) \supseteq \ldots \supseteq A_{t^*}(\mathbf{X}) \supseteq \ldots \supseteq A_T(\mathbf{X}) \rangle$$

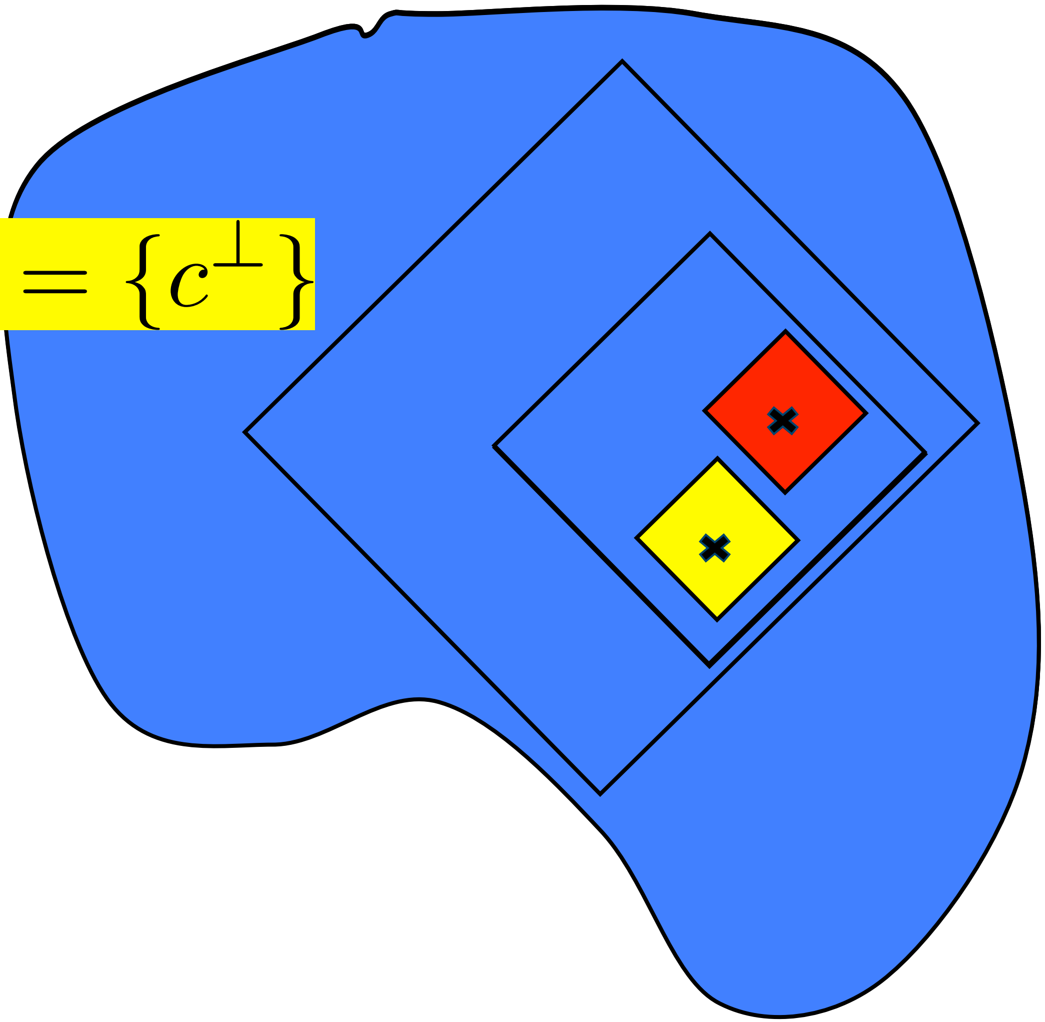
Hypotheses explored by an algorithm $\mathcal{A}$

data $\mathbf{X}'$

data $\mathbf{X}''$

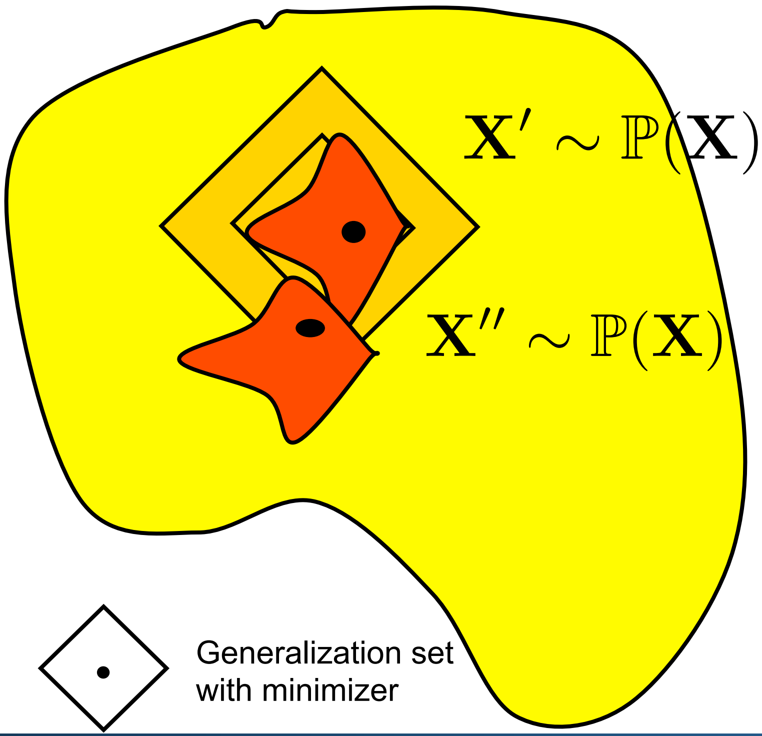
$A_T(\mathbf{X}') = \{c^\perp\}$

$A_T(\mathbf{X}'')$



Coarsening of hypothesis classes and the two instances test

- Quantize **hypothesis class** by generalization sets



- Size: "partition function" at iteration t is $|A_t(\mathbf{X}')|$
- Weight overlap models joint approximations $|A_t(\mathbf{X}') \cap A_t(\mathbf{X}'')|$
- Posterior of hypothesis c

$$p^{\mathcal{A}}(c|\mathbf{X}') = \begin{cases} \dfrac{1}{|A_t(\mathbf{X}')|} & \text{if } c \in A_t(\mathbf{X}') \\ 0 & \text{otherwise} \end{cases}$$

Information theory: structures as symbols

1) sample a hypothesis $\tilde{c} \sim p^{\mathcal{A}}(c|\mathbf{X})$

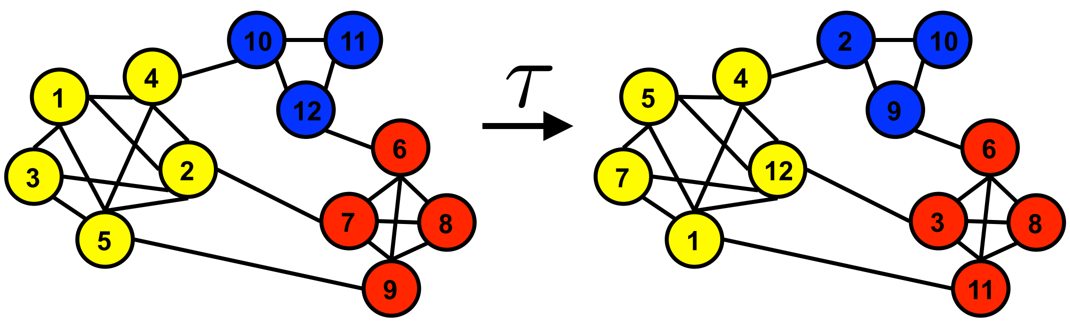
2) for $j = 1 \ldots M$

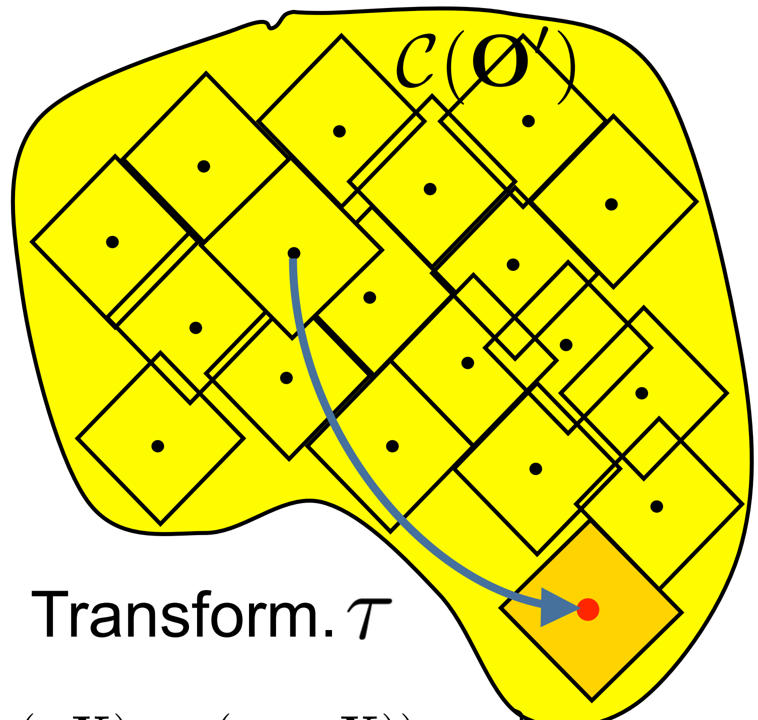
Select a random transf. $\tau_j \in \mathbb{T}(\tilde{c})$

define code vector $\tilde{c}_j := \tau_j \circ \tilde{c}$

3) return codebook $\mathcal{T} := \{\tilde{c}_1, \ldots, \tilde{c}_M\}$

Cover **hypothesis class** densely, but identifiably!



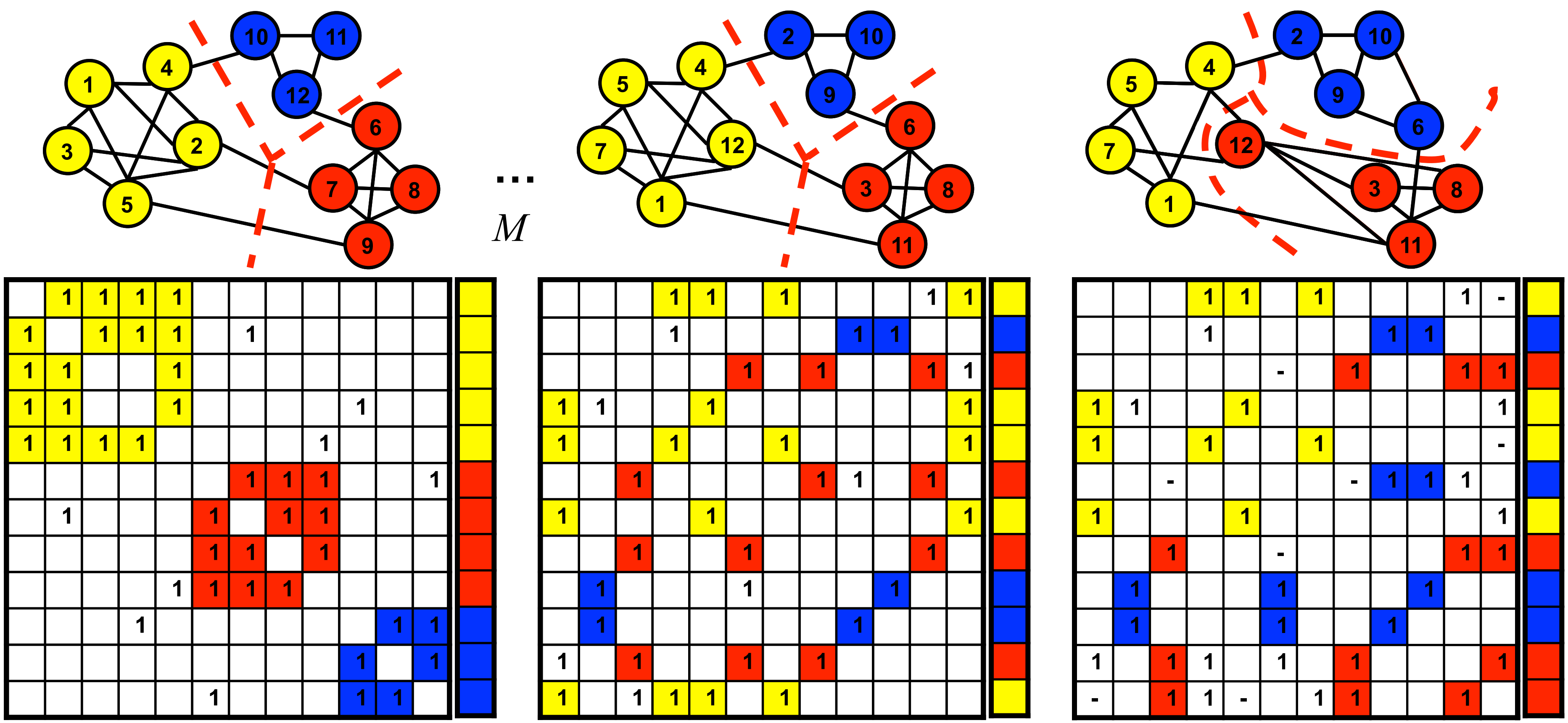


$$\mathbb{T}(\tilde{c}) = \{\tau : \forall \beta, c, w_\beta(c, \mathbf{X}) = w_\beta(\tau \circ c, \tau \circ \mathbf{X}) \ \wedge \ D(w_\beta(., \mathbf{X}), w_\beta(., \tau \circ \mathbf{X})) > \rho\}$$

Code problem generation for graph cut

graph cut *code* problems

graph cut *test* problem



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**

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

- **Error** event: $\hat{\tau} \neq \tau_s$

Generalization capacity from typicality

- **Theorem**: Asymptotic error free *identification* $\lim_{n\to\infty} P(\hat{\tau} \neq \tau_s | \tau_s) = 0$ of *code structures* is possible for

$$P\left(\hat{\tau} \neq \tau_s | \tau_s\right) \;\leq\; M\, \mathbb{E}_{\mathbf{X}',\mathbf{X}''} \Big(|\mathbb{T}| k^{\mathcal{A}}(\mathbf{X}', \mathbf{X}'') \Big)^{-1}$$

$$\overset{\text{typicality}}{\leq} \;\exp\Big(-(\mathcal{I} - \log M) \Big) \qquad \text{with}$$

$$\mathcal{I} \;=\; \mathbb{E}_{\mathbf{X}',\mathbf{X}''} \log\Big(|\mathbb{T}| k^{\mathcal{A}}(\mathbf{X}', \mathbf{X}'') \Big)$$

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

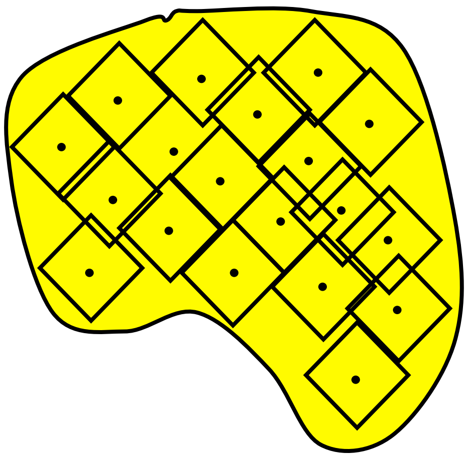
Behavior of the generalization capacity

- Let us assume that $\mathbf{X}', \mathbf{X}''$ are very similar.

$$\mathbf{X}' \approx \mathbf{X}'' \approx \mathbf{X}$$

$$k^{\mathcal{A}}(\mathbf{X}, \mathbf{X}) = \sum_{c \in \mathcal{C}} p^{\mathcal{A}}(c|\mathbf{X})^2 = \frac{|A_{t^*}(\mathbf{X}) \cap A_{t^*}(\mathbf{X})|}{|A_{t^*}(\mathbf{X})|^2}$$

$$\mathcal{I} = \mathbb{E}_{\mathbf{X}} \log\Big(\frac{|\mathbb{T}|}{|A_{t^*}(\mathbf{X})|}\Big)$$



- 2nd order terms yield maximum of GC $\mathcal{I}$ for parameter selection

Learning an algorithm: open challenge!

- **Statistical behavior of an algorithm** is described by its posterior $p^{\mathcal{A}}(c|\mathbf{X}')$
- **Adapt posterior** $p^{\mathcal{A}}(c|\mathbf{X}')$ s.t. generalization capacity is maximized

$$p^* \in \arg \max_{\substack{p^{\mathcal{A}}:\mathcal{X}\times\mathcal{C}\to[0,1] \\ \sum_c p^{\mathcal{A}}(c|\mathbf{X})=1}} \mathbb{E}_{\mathbf{X}',\mathbf{X}''} \log\Big(|\mathbb{T}|\, k^{\mathcal{A}}(\mathbf{X}',\mathbf{X}'')\Big)$$

- **Problem**: We cannot evaluate $\mathbb{E}_{\mathbf{X}',\mathbf{X}''} \log \ldots$ since $P(\mathbf{X}',\mathbf{X}'')$ is unknown!

Roadmap

- **Robust computation versus learning**
  Measuring the information content of algorithms
- **Algorithm/Model validation** by information theory
- **Learning optimal algorithms**: open challenge!
  - Validating approximate spanning trees
  - Graph cut for gene expression analysis
  - Robust sorting

Example: Robust Spanning Trees

- **Given** are graphs with stochastic edge weights.
- **Question**: How robust are MST algorithms in this stochastic setting?
- Measure **robustness of algorithms** like Prim's, Kruskal's algorithm or the Reverse-Delete algorithm.
- Determine a **stable set of approximate spanning trees** by early stopping of an MST algorithm.

Learning to span a graph

Alexey Gronskiy

Consider **Minimum Spanning Tree** algorithms

- **Prim**’s “Growing tree” strategy: add minimal edge to tree.
- **Kruskal**’s “Joining trees” strategy: add minimal edge connecting two trees in a forest.
- **Reverse-Delete**: “Reducing graph” strategy: delete maximal edge without destroying connectivity.

MST Algorithm as a sequence of approximate spanning tree sets

- Let $\mathbf{X}$ be a graph and $A_t(\mathbf{X})$ the set of spanning trees at iteration t

$$\begin{aligned} \mathcal{A}(\mathbf{X}) &= \langle A_0(\mathbf{X}), \ldots, A_T(\mathbf{X}) \rangle, \\ A_t(\mathbf{X}) &\subseteq \mathcal{C}, \quad 0 \leq t \leq T, \\ A_0(\mathbf{X}) &= \mathcal{C}, \quad A_T(\mathbf{X}) = \{c^\perp\}. \end{aligned}$$

- Monotonically contractive algorithms

$$\mathcal{A}(\mathbf{X}) = \langle A_0(\mathbf{X}) \supseteq A_1(\mathbf{X}) \supseteq \ldots \supseteq A_T(\mathbf{X}) \rangle$$

Cardinality of AST sets

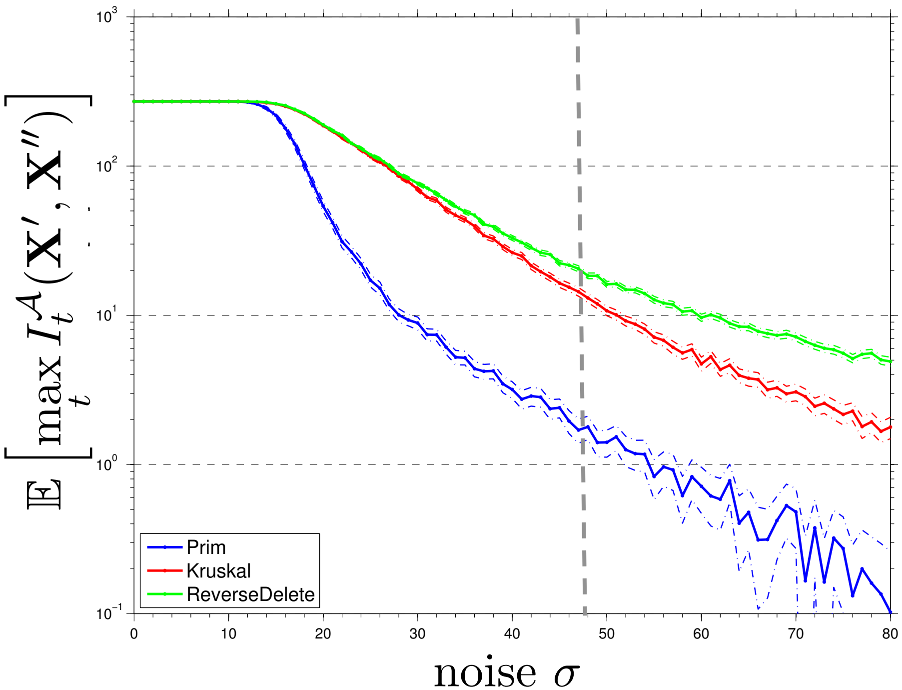
- **Matrix tree theorem**: the number of spanning trees is equal to (any) cofactor of the matrix

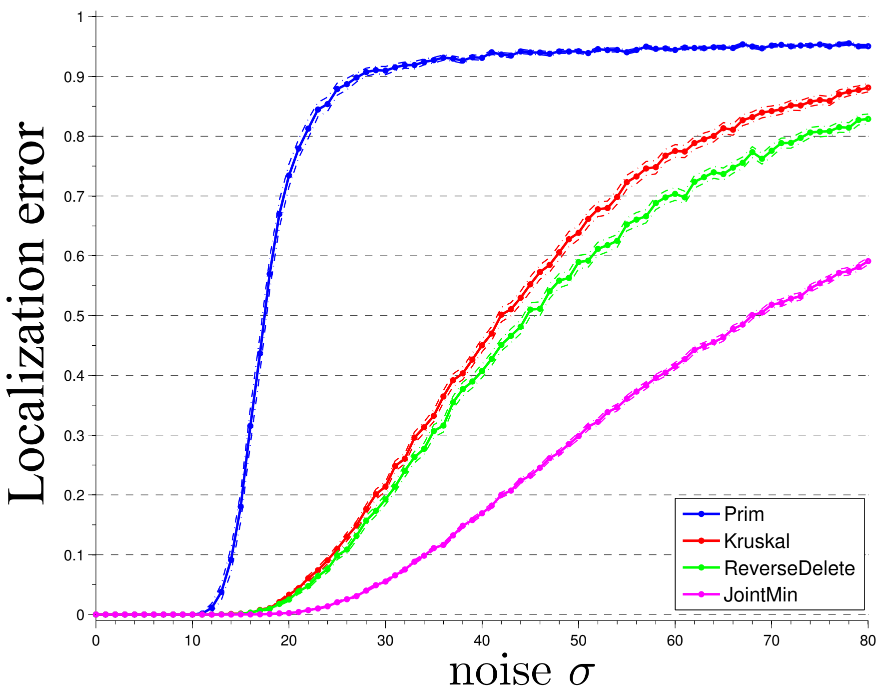
$$L = M_{\text{deg}}^{X} - M_{\text{adj}}^{X}$$

- Calculate the cofactor of L after t steps for the effective $X(t)$ where selected edges are contracted (Prim, Kruskal) or removed (Reverse-Delete).

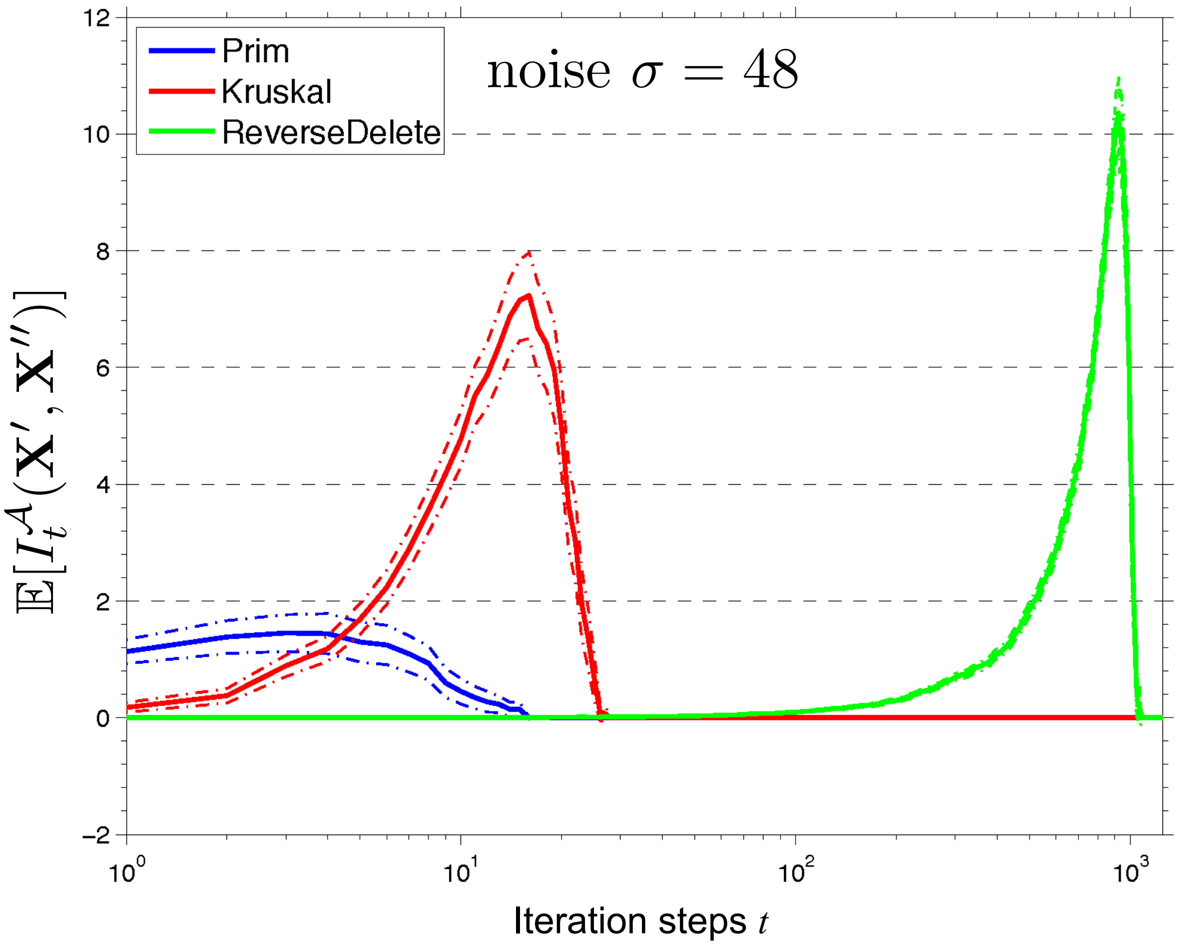
Algorithmic informativeness

- Hierarchical graph generation:
  - ground truth graph: 50 vertices, i.i.d. normal weights $\mathcal{N}(100, 100)$
  - Additive Gaussian noise $\mathcal{N}(0, \sigma^2) \;\Rightarrow\; \mathbf{X}', \mathbf{X}''$



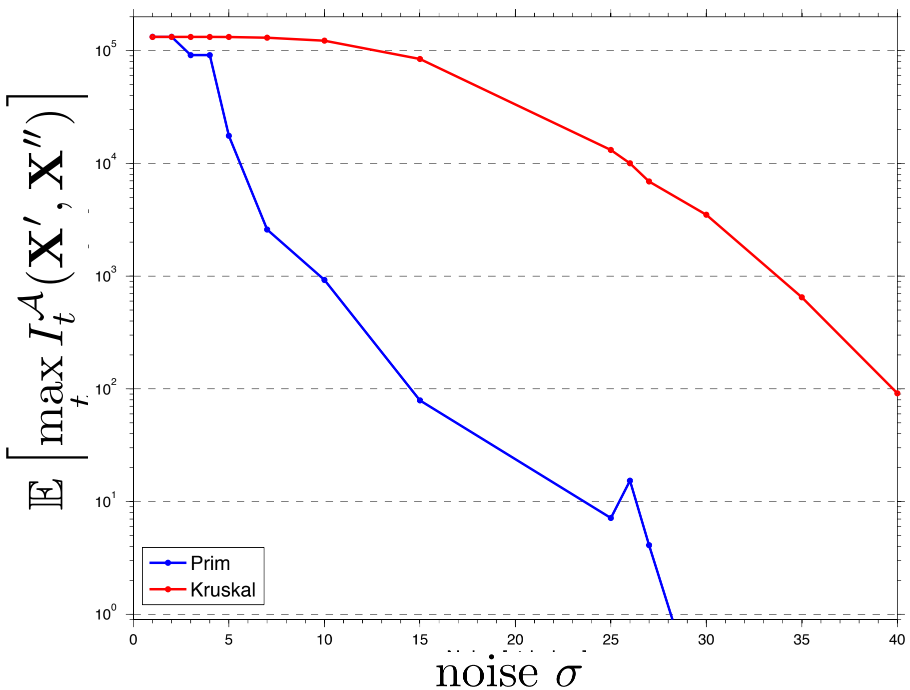


Dynamics of algorithmic informativeness



Informativeness of ASTs with 10^4 vertices

- Hierarchical graph generation:
  - ground truth graph: 10^4 vertices, i.i.d. normal weights $\mathcal{N}(100, 100)$
  - Additive Gaussian noise $\mathcal{N}(0, \sigma^2)$



Conclusion

- **Quantization**: Noise quantizes mathematical structures (hypothesis classes) **=>** symbols
- **Coding** with these symbols defines a **generalization capacity** for **algorithms**

$\Rightarrow$ **Quantization** of hypothesis class measures **structure specific information** in data.

- How to relate **statistical complexity** to algorithmic or **computational complexity** ?