ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

inf | Informatik
Computer Science

Decision making under uncertainty:
How informative is your algorithm with noisy inputs and internal computation errors?

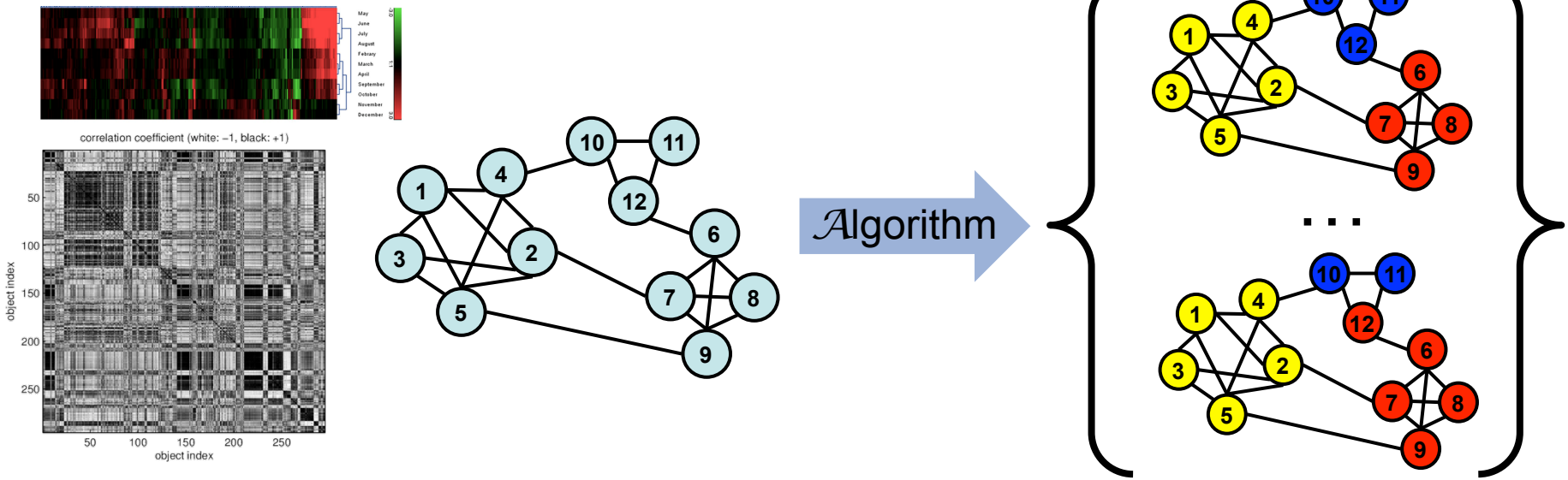
Joachim M. Buhmann
Computer Science Department, ETH Zurich

Roadmap

- **Robust computation versus learning**
  Measuring the information content of algorithms
- **Algorithm/Model validation** by information theory
- **Learning optimal algorithms**: open challenge!
  - Graph cut for gene expression analysis
- Outlook & vision: Low power computing, resilient programming

What is learning?

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



- **Modeling in pattern recognition** requires
  - **quantization**: identify a set of "good" hypotheses,
  - **learning**: find an algorithm that specifies such a set!

Robust algorithms for pattern recognition

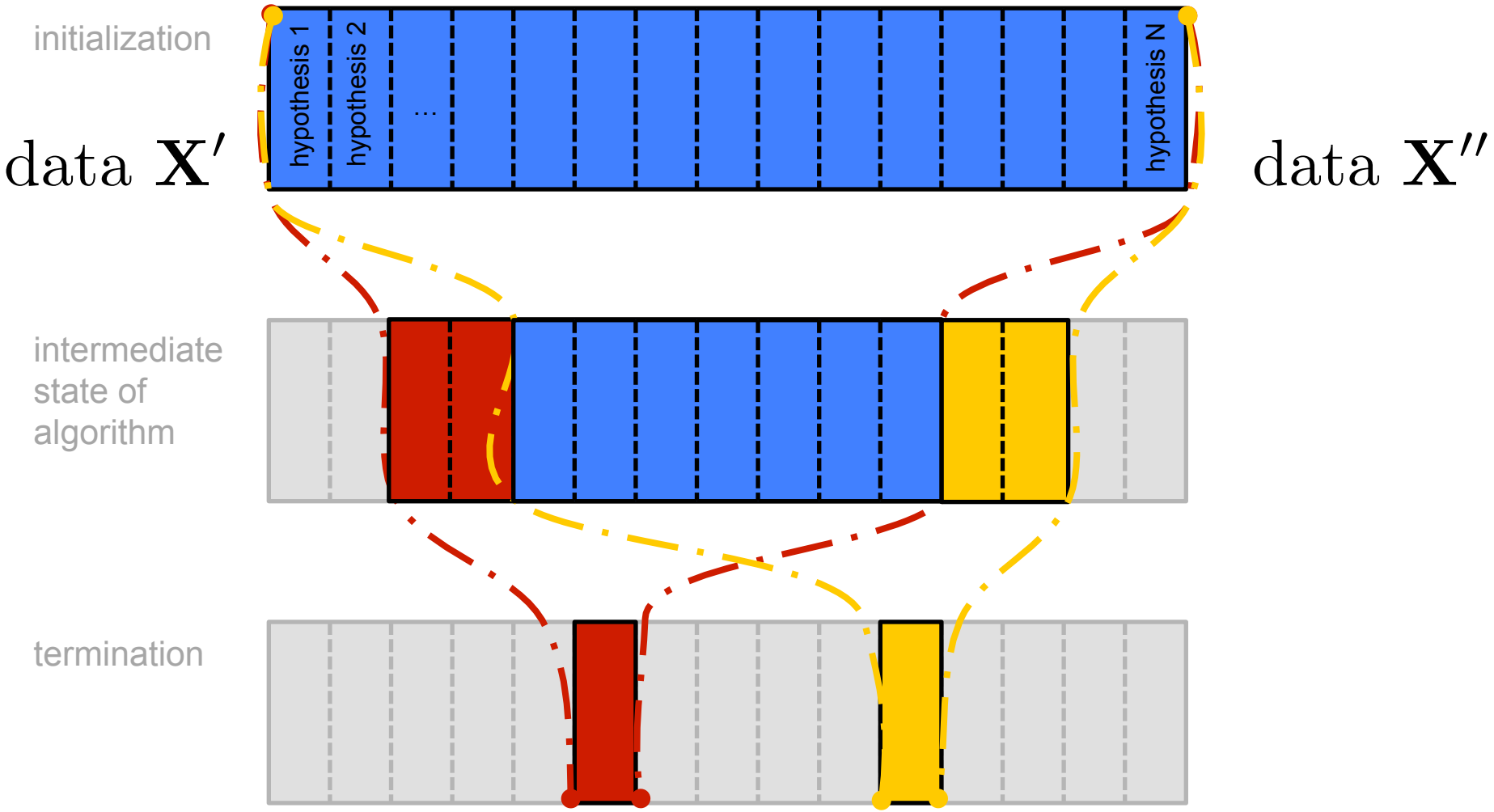
- **Classical view**: algorithm $\mathcal{A}$ maps (stochastic) data (input) to a solution / hypothesis (output).

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

- **Discriminative probabilistic view**: algorithm $\mathcal{A}$ maps stochastic data (input) to weights over solutions / hypotheses (output)

$$\text{input } (X, \gamma) \to \mathcal{A} \to \text{output } w_\gamma^{\mathcal{A}}(c, X) \in [0, 1]$$

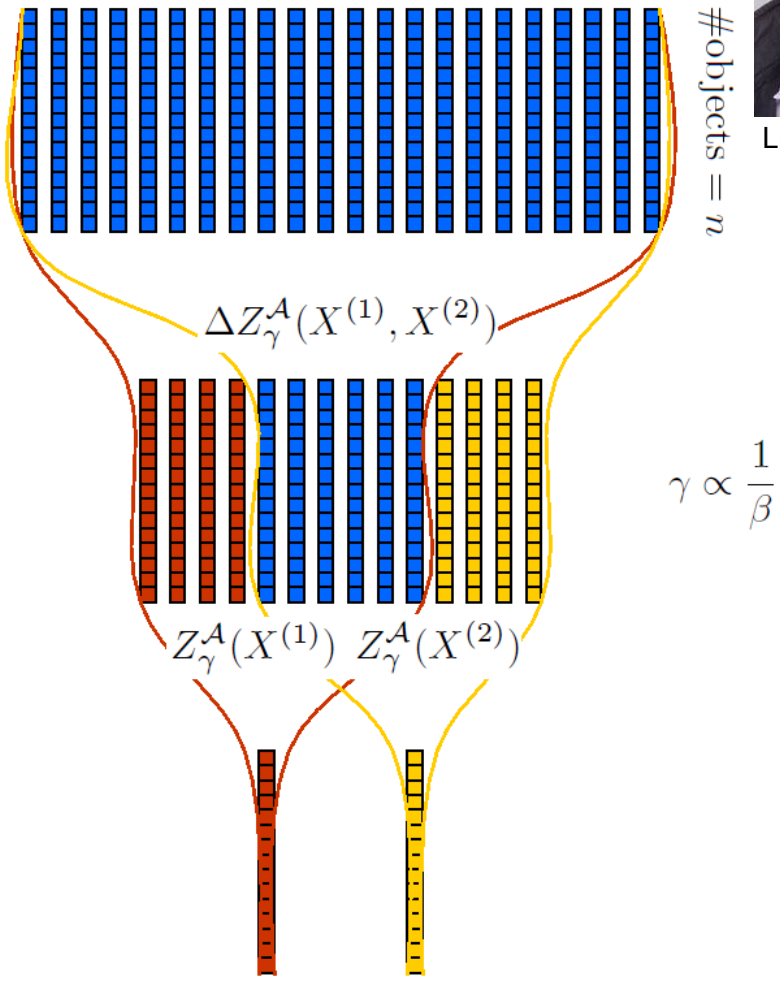
Hypotheses explored by an algorithm $\mathcal{A}$



Contractive dynamics of an algorithm

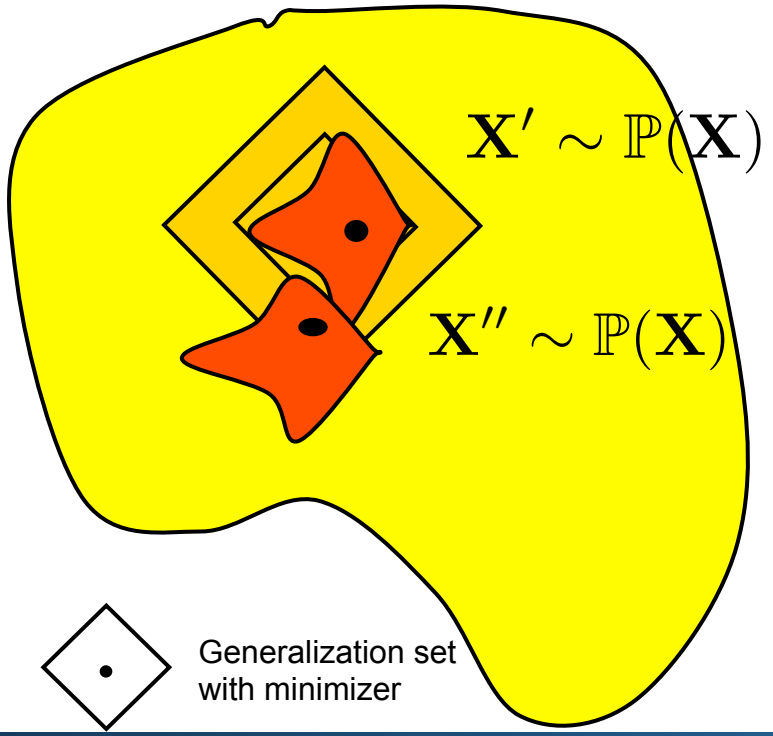

Ludwig Busse

1. **INIT**: All hypotheses have equal weights
2. **WHILE** not converged **{**
   shrink weights of poor hypotheses
   }
3. **RETURN** hypotheses with large weights



Coarsening of hypothesis classes and the two instances test

- Quantize **hypothesis class** by generalization sets



- Size: partition function

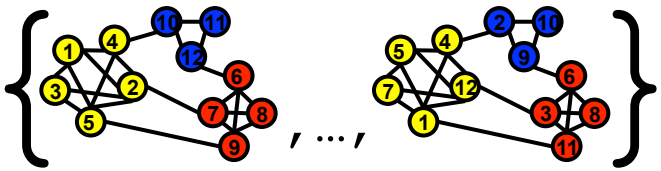
$$Z(\mathbf{X}') = \sum_{c \in \mathcal{C}(\mathbf{O}')} w_\beta(c, \mathbf{X}')$$

- Weight overlap models joint approximations

$$\Delta Z(\mathbf{X}', \mathbf{X}'') = \sum_{c \in \mathcal{C}(\mathbf{O}')} w_\beta(c, \mathbf{X}')\, w_\beta(c, \mathbf{X}'')$$

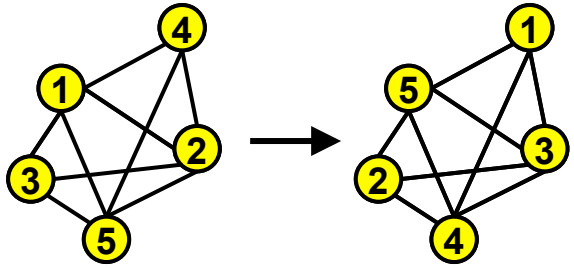
Generalization sets as symbols for coding

- **Set of generalization sets**

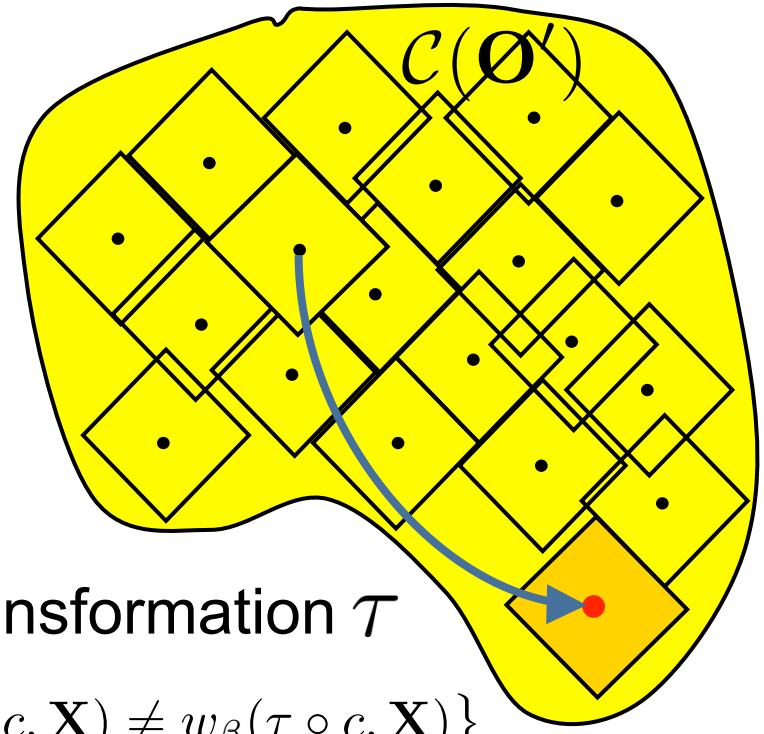


- How to generate sets of generalization sets?

  Random transformations



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



$$\mathbb{T} = \{\tau : \forall \beta, w_\beta(c, \mathbf{X}) = w_\beta(\tau \circ c, \tau \circ \mathbf{X}) \ \wedge \ w_\beta(c, \mathbf{X}) \neq w_\beta(\tau \circ c, \mathbf{X})\}$$

Communication Process

- Receiver **compares sets of hypothesis weights**
- **Decoding by maximizing weight overlap**

$$\hat{\tau} := \arg\max_{\tau} \#\left(\diamond_{\tau} \cap \diamond \right)$$

- **Error** event: $\hat{\tau} \neq \tau_s$
- Calculate $\lim_{n\to\infty} P(\hat{\tau} \neq \tau_s | \tau_s) = 0$

Error Probability

- Estimate error given random transformations $\tau \in \mathcal{T}$

$$\Delta Z_j := \sum_{c \in \mathcal{C}(\mathbf{X}'')} w_\beta(c, \tau_j \circ \mathbf{X}')\, w_\beta(c, \tau_s \circ \mathbf{X}'')$$

$$P\left(\hat{\tau} \neq \tau_s | \tau_s\right) = P\left(\max_{j \neq s} \Delta Z_j > \Delta Z_s \middle| \tau_s\right)$$

Union bound

$$\leq \sum_{j \neq s} P\left(\Delta Z_j > \Delta Z_s | \tau_s\right)$$

$$\leq M P\left(\Delta Z_{\neq s} > \Delta Z_s | \tau_s\right)$$

Markov ineq.

$$\leq M\, \mathbb{E}_{\mathbf{X}', \mathbf{X}''} \frac{\mathbb{E}_{\tau_{\neq s}} \Delta Z_{\neq s}}{\Delta Z_s} = M \mathbb{E} \frac{Z(\mathbf{X}') Z(\mathbf{X}'')}{|\mathbb{T}| \Delta Z_s}$$

Generalization capacity from typicality

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

$$P\left(\hat{\tau} \neq \tau_s | \tau_s\right) \leq M\, \mathbb{E}_{\mathbf{X}',\mathbf{X}''} \frac{Z(\mathbf{X}')\, Z(\mathbf{X}'')}{|\mathbb{T}|\, \Delta Z_s(\mathbf{X}', \mathbf{X}'')}$$

$$\overset{\text{typicality}}{\leq} M \exp\Big(-\mathbb{E}_{\mathbf{X}',\mathbf{X}''} \log |\mathbb{T}| \underbrace{\sum_{c\in\mathcal{C}} p(c|\mathbf{X}')p(c|\mathbf{X}'')}_{\in[0,1]}\Big)$$

$$\text{with posterior } p(c|\mathbf{X}) = \frac{w_\beta(c, \mathbf{X})}{Z_\beta(\mathbf{X})}$$

Learning an algorithm: open challenge!

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

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

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

Consistent learning of an algorithm

- **Learning** requires that an empirical estimate of the capacity should be close to its expectation

$$\sum_{l} \log |\mathbb{T}| \sum_{c \in \mathcal{C}} \hat{p}^*(c|\mathbf{X}'_l)\, \hat{p}^*(c|\mathbf{X}''_l) \approx$$

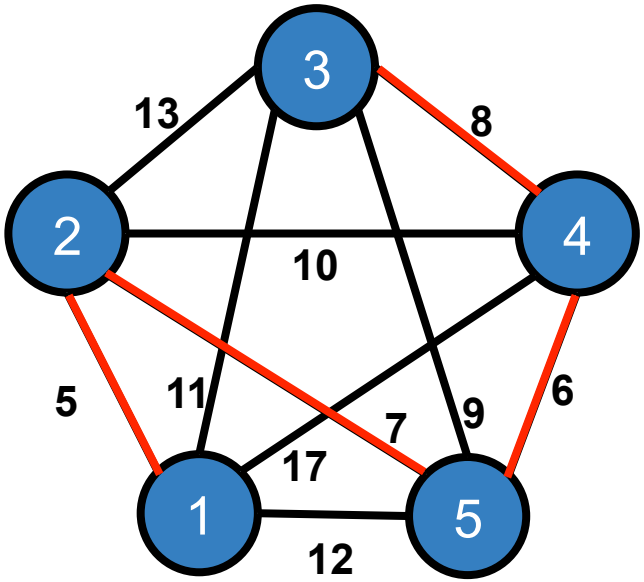
$$\mathbb{E}_{\mathbf{X}',\mathbf{X}''} \log |\mathbb{T}| \sum_{c \in \mathcal{C}} p^*(c|\mathbf{X}')\, p^*(c|\mathbf{X}'')$$

$$\text{with optimal (empirical,expected) posterior } \left(\hat{p}^*, p^*\right)$$

Learning to span a graph

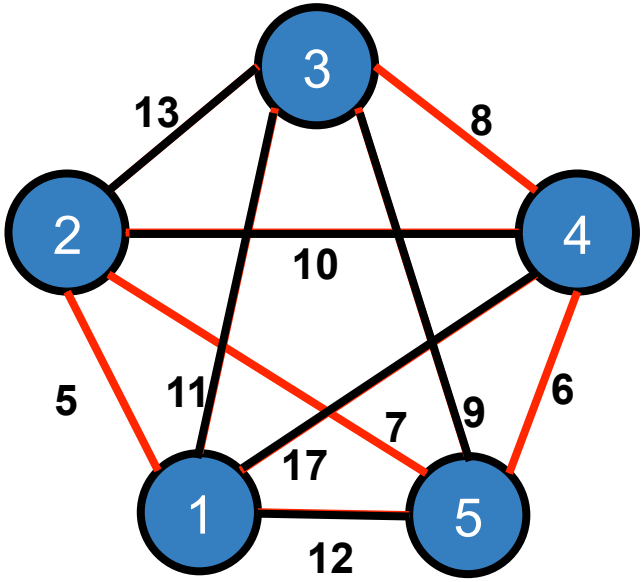
- Consider **Minimum Spanning Tree** algorithms
- **Prim**'s "Growing tree" strategy: add minimal edge to tree.
- **Kruskal**'s "Joining trees" strategy: add minimal edge connecting two trees in a forest.
- **Reverse-Delete**: "Reducing graph" strategy: delete maximal edge without destroying connectivity.

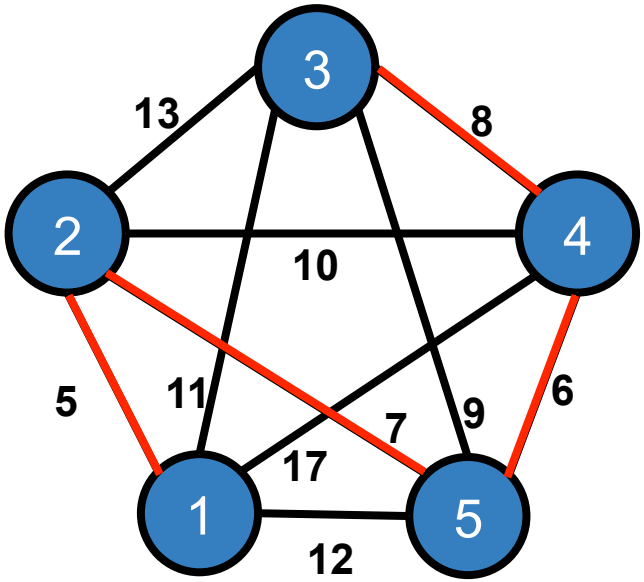
Flow of MST algorithms



Prim

Reverse - delete

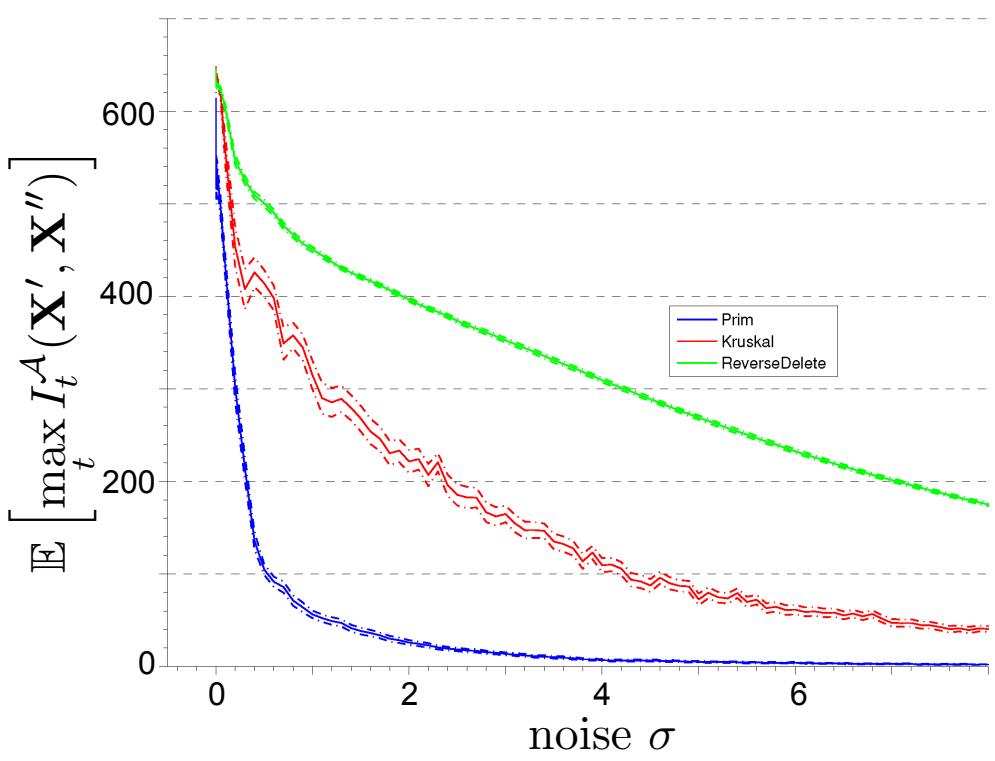


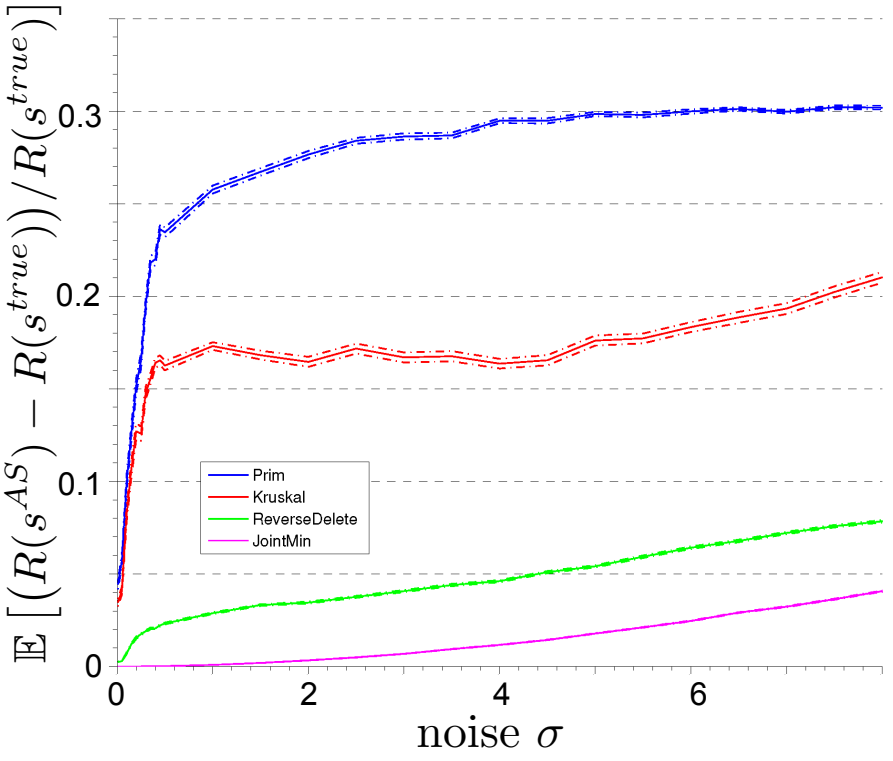


Kruskal

Algorithmic informativeness

100 vertices, uniform random i.i.d. weights, Gaussian noise



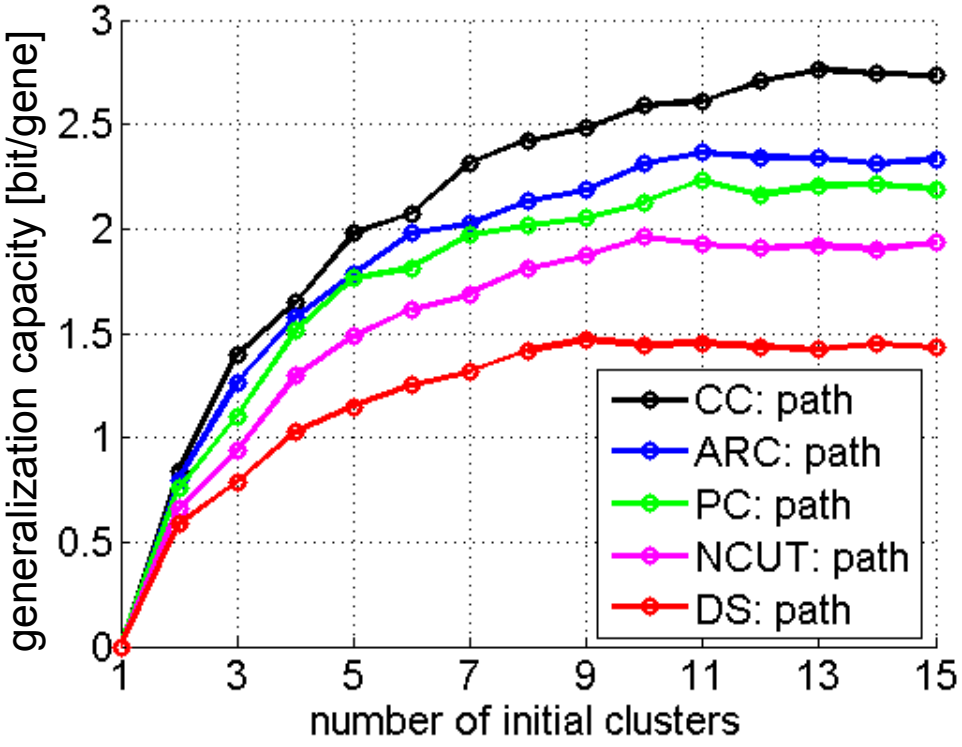


Approximation capacity of clustering yeast gene expression data

Morteza H. Chehreghani

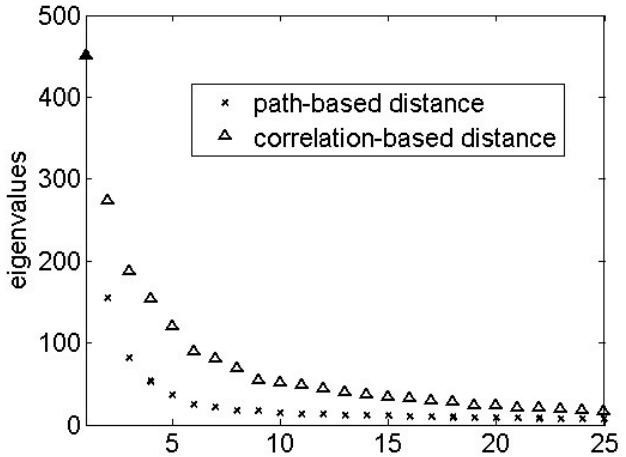
Ranking of spectral clustering methods for different dissimilarity measures [Eisen et al. (1998) PNAS 95,14863]

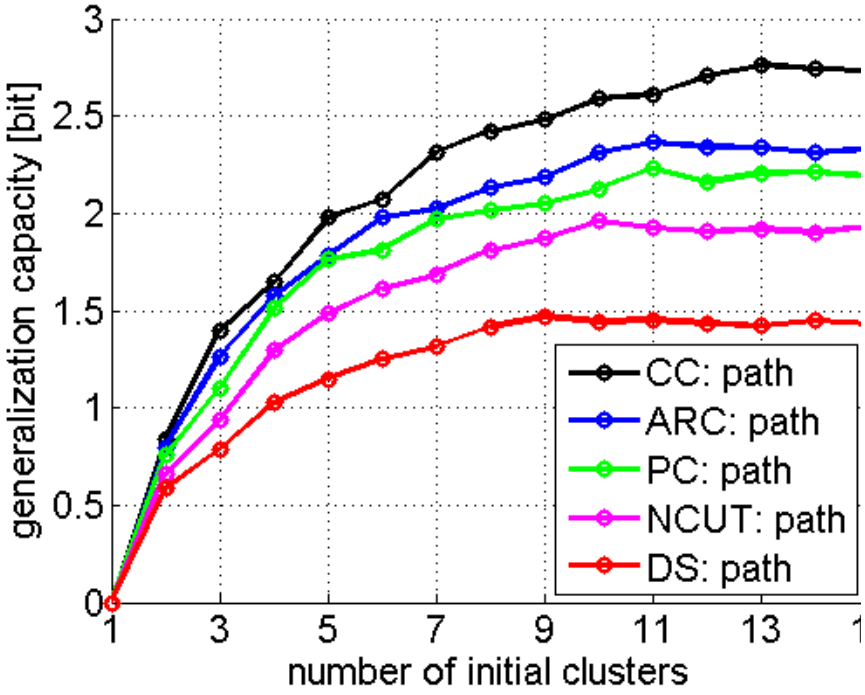
1. shifted correlation clustering (CC)
2. adaptive ratio cut (ARC)
3. pairwise clustering (PC)
4. normalized cut (Ncut)
5. dominant set clustering (DS) ($\theta = 10^{-4}$)

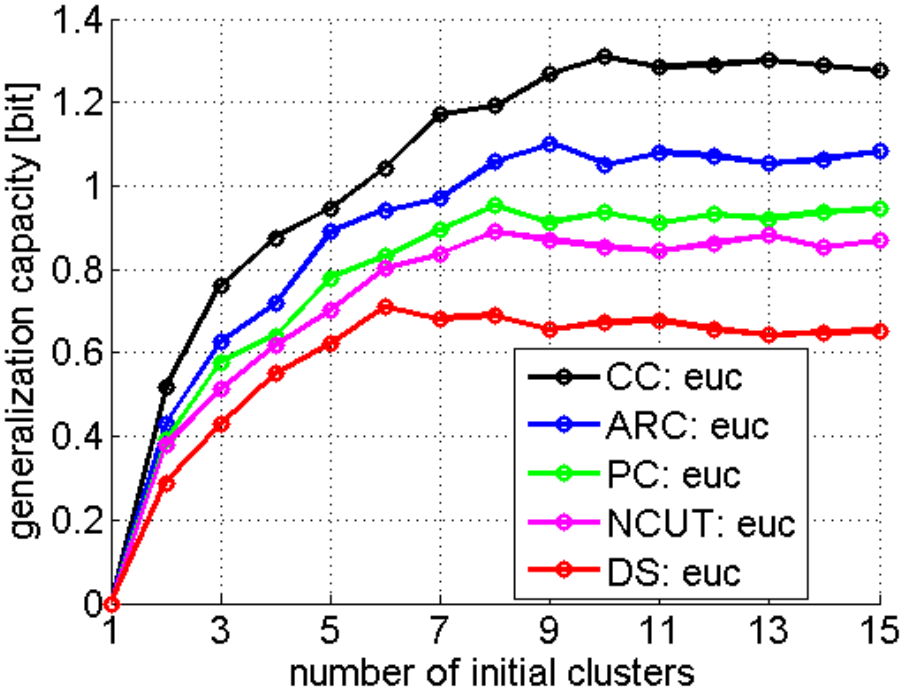


Influence of metric on AC

- Path-based distance
- Correlation dissimilarity
- Euclidean distance

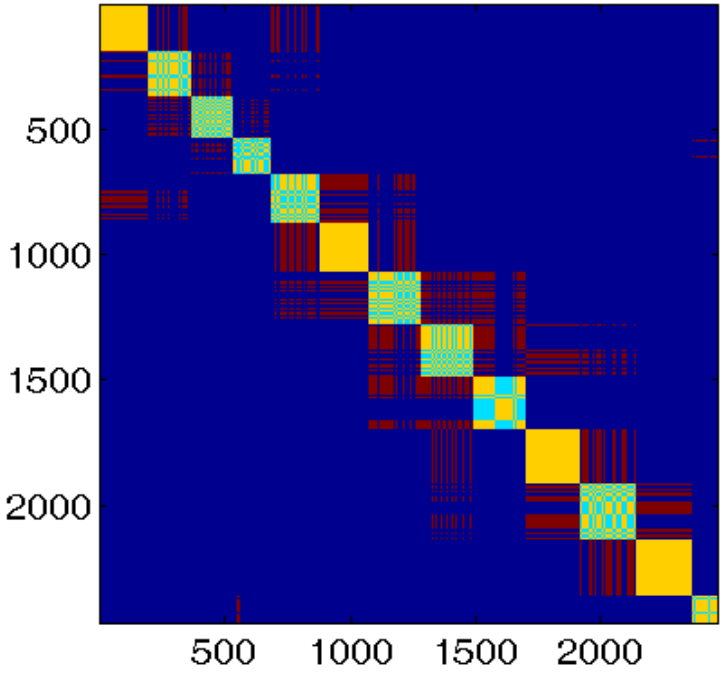




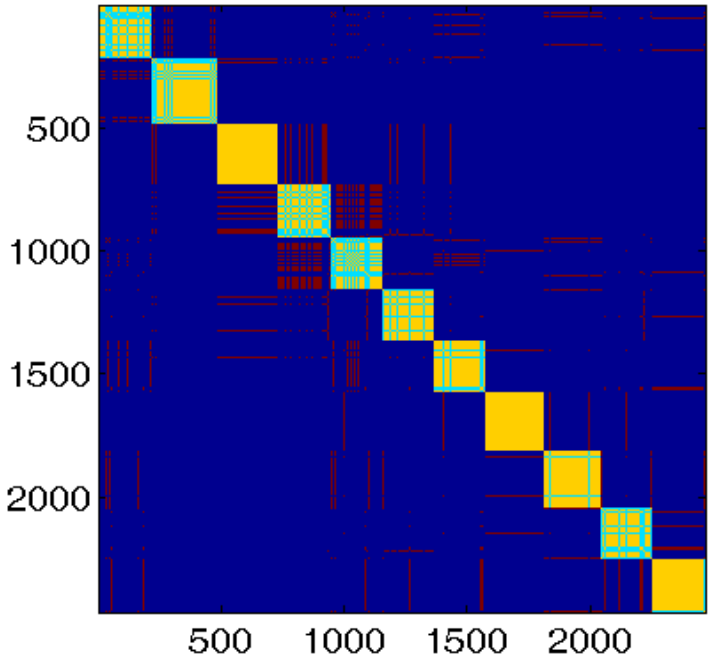


Consistency of clusterings

- CC & PC cluster objects (i,j) together
- CC & PC cluster objects (i,j) separately
- CC clusters objects (i,j) together, PC clusters separately
- CC clusters objects (i,j) separately, PC clusters together



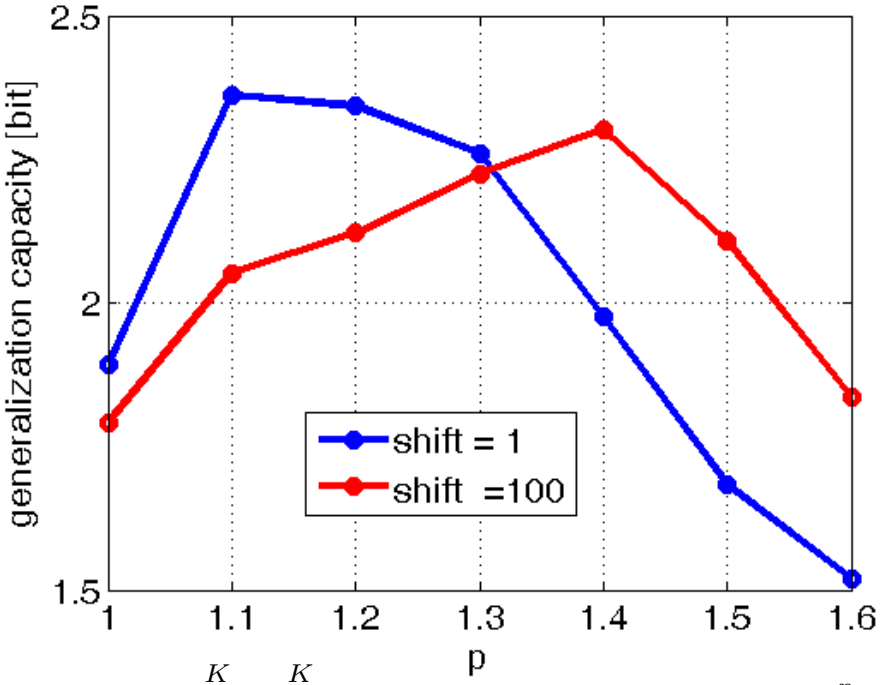
CC vs. PC (path distance)



CC vs. PC (correlation distance)

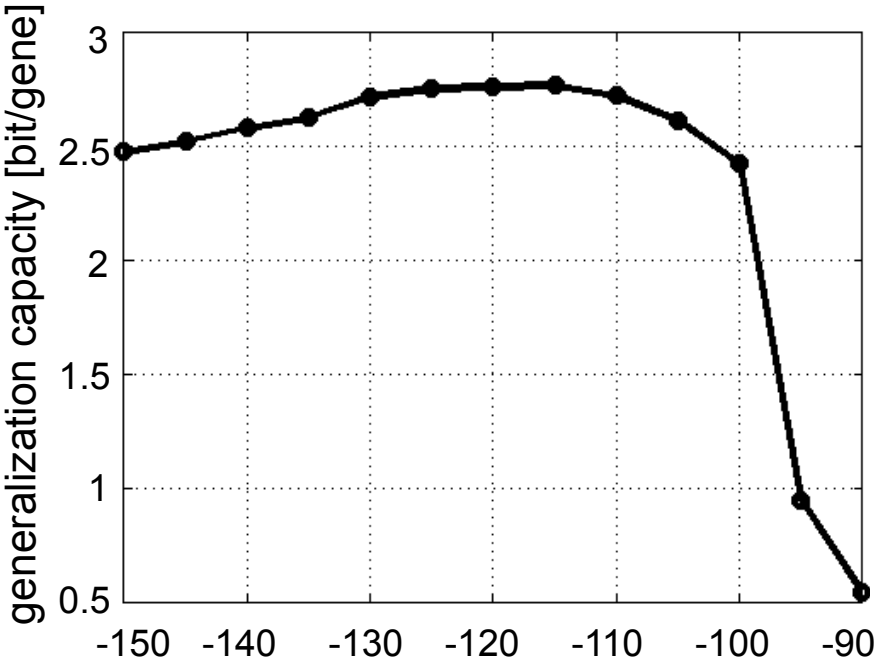
Parameter adaptation of clustering costs

Generalization capacity vs. exponent for adaptive ratio cut



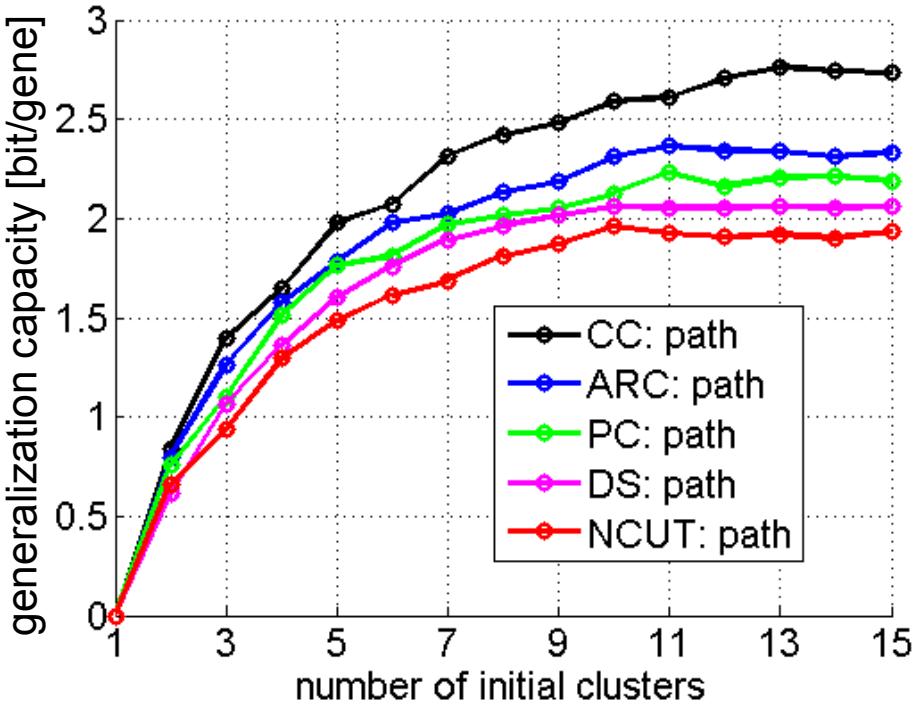
$$R(c, \mathbf{X}) = \sum_{\alpha=1}^{K} \sum_{\beta=\alpha+1}^{K} cut(V_\alpha, V_\beta) \left(|V_\alpha|^{\frac{1}{p}} + |V_\beta|^{\frac{1}{p}} \right)^{-p}$$

Generalization capacity vs. shift for correlation clustering

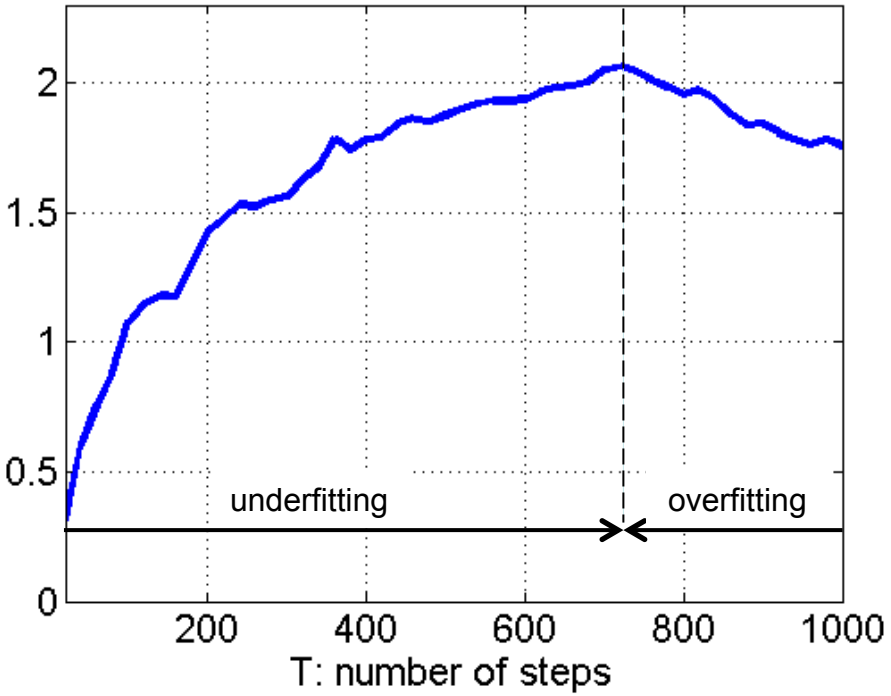


$$R(c, \mathbf{X}, K, s) = \frac{1}{2} \sum_{1 \le u \le K} \sum_{(i,j) \in E_{uu}} (|X_{ij} + s| - X_{ij} - s)$$
$$+ \frac{1}{2} \sum_{1 \le u \le K} \sum_{1 \le v < u} \sum_{(i,j) \in E_{uv}} (|X_{ij} + s| + X_{ij} + s).$$

Generalization capacity of Dominant Set Clustering with adaptive threshold



Evolution of DS generalization capacity



Conclusion

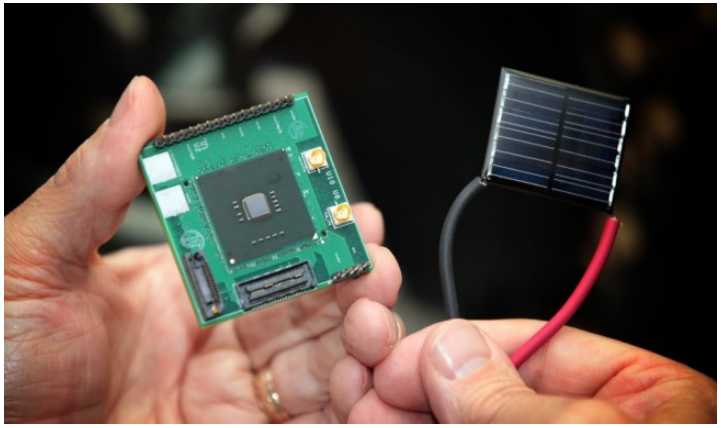
- **Quantization**: Noise quantizes mathematical structures (hypothesis classes) **=>** symbols
- These symbols can be used for **coding**!
- Optimal error free coding scheme determines **approximation capacity** of a model class.

$\Rightarrow$Bounds for robust optimization.

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

Low-Energy Architecture Trends

- Novel low-power architectures operate **near transistor threshold voltage** (NTV)
  - e.g., Intel Claremont
  - 1.5 mW @10 MHz (x86)

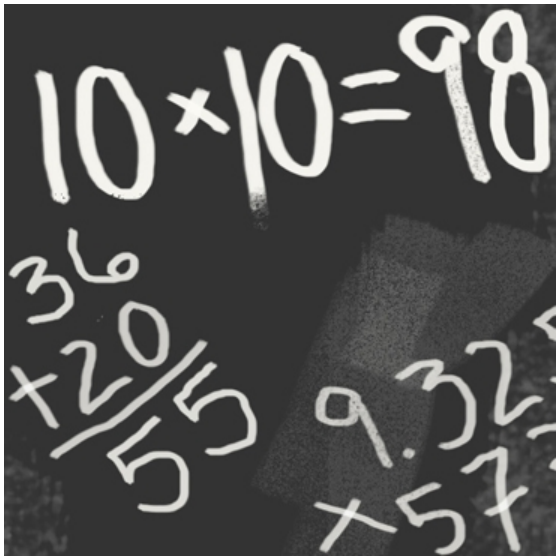


source: Intel

- NTV promises 10x more energy efficiency at 10x more parallelism!
  - 10^5 times more soft errors (bits flip stochastically)
  - Hard to correct in hardware → expose to programmer?

@ Thorsten Höffler

Resilient Programming



Source: MIT, CHRISTINE DANILOFF

- Distorted computations
  - Some parts of the program work
    - e.g., Conjugate Gradient while computing the new direction
  - Others cannot
    - e.g., the target address for a jump-table
  - NTV is very intriguing for large-scale computing
- Interesting model for developing algorithms, programming systems, and architectures

@ Thorsten Höffler