ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

inf | Informatik
Computer Science

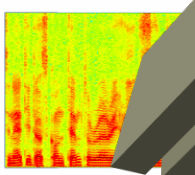
Context sensitive information: Model validation by information theory

Joachim M. Buhmann

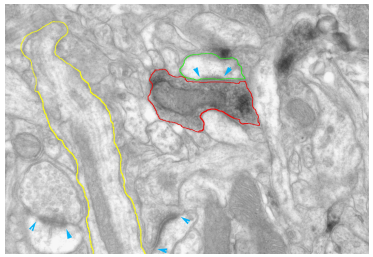
Computer Science Department, ETH Zurich

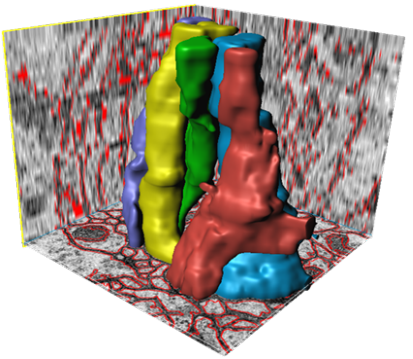
ETH Machine Learning Laboratory

adaptive signal processing

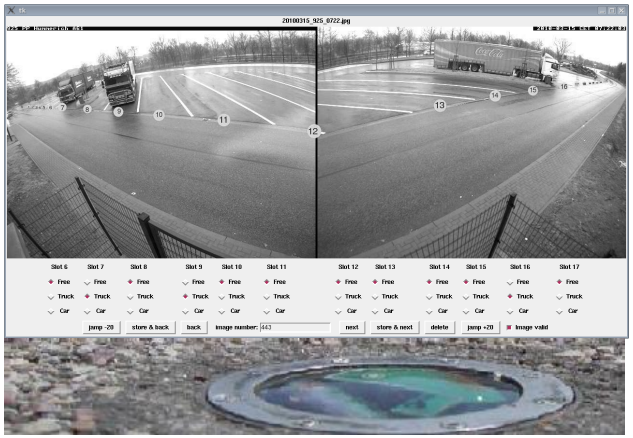


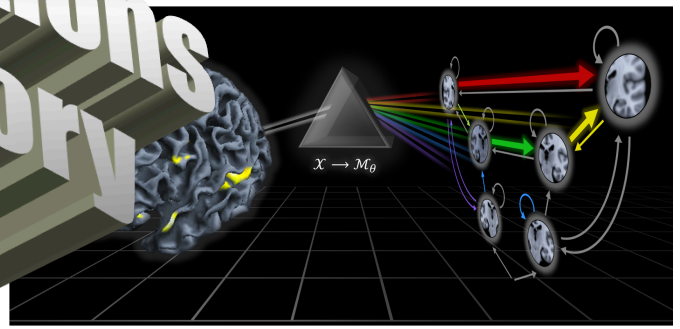
Applications & Theory



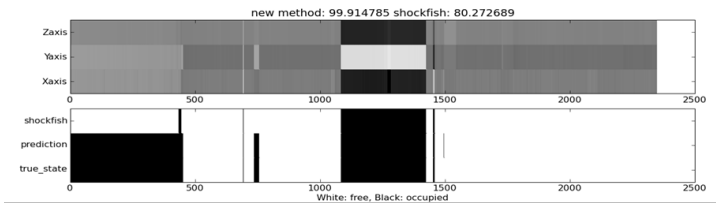


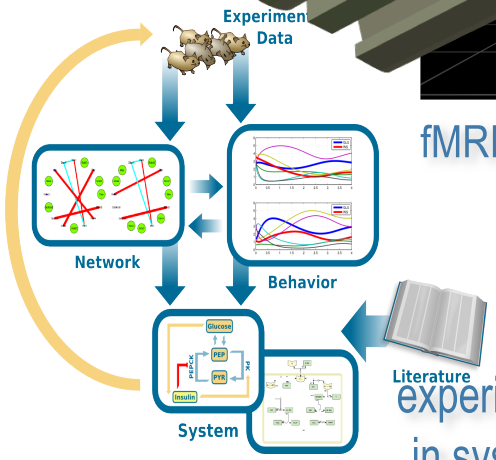
neuron reconstruction





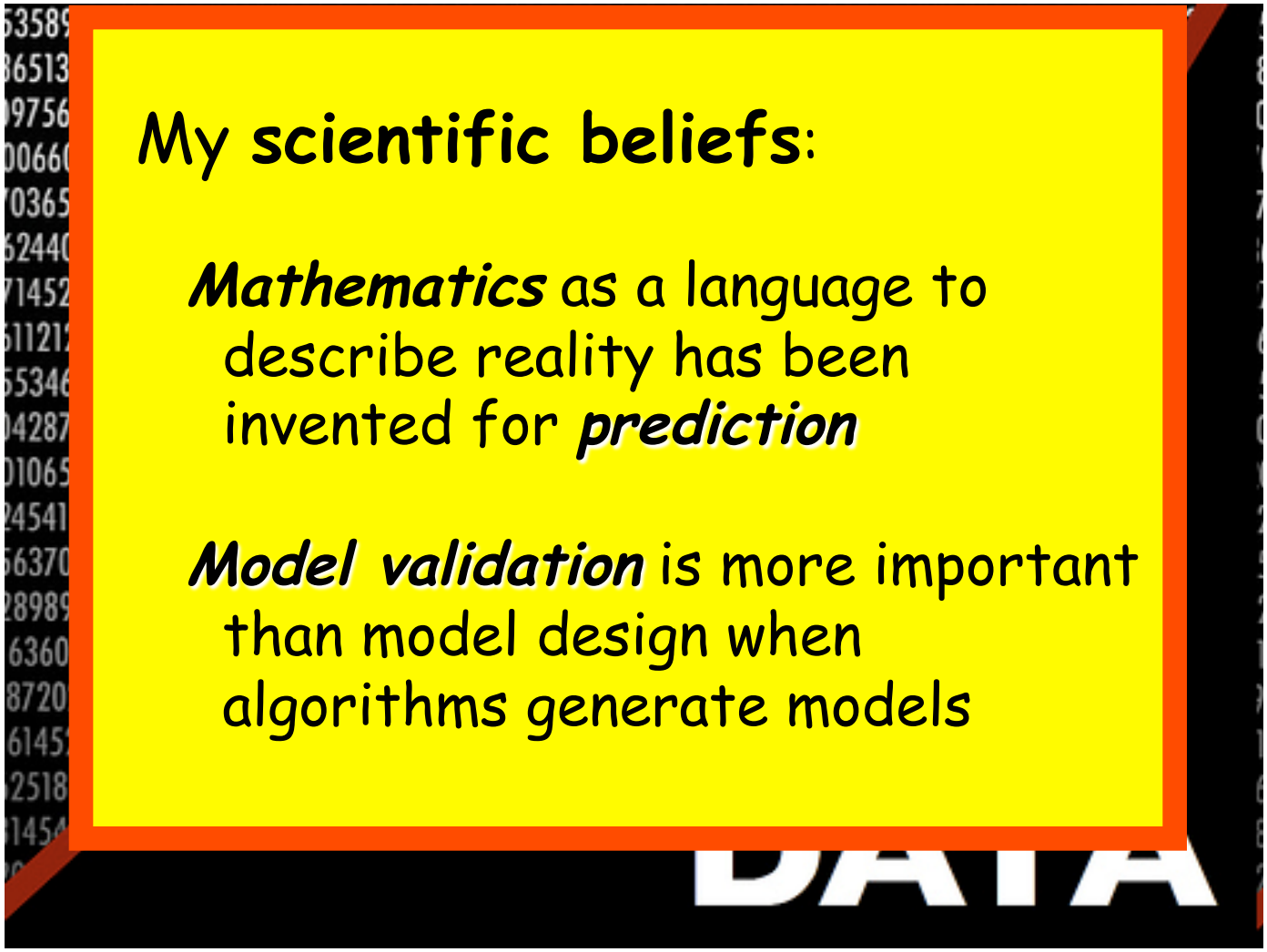
fMRI analysis for e.g. disease diagnosis



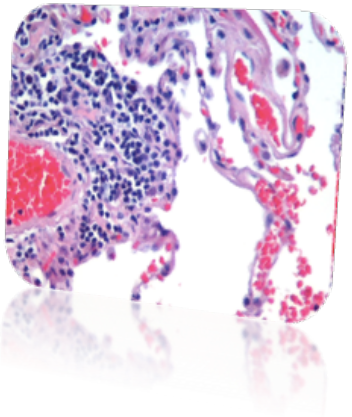


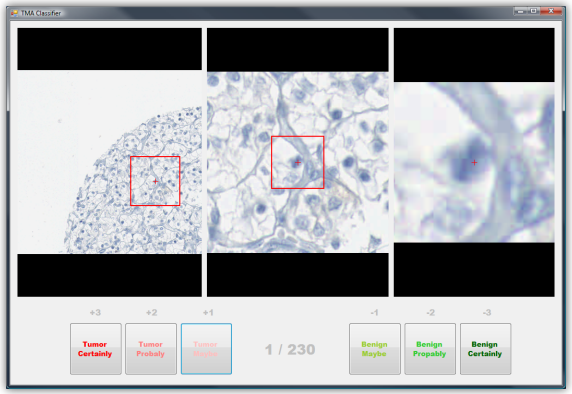
experimental design in systems biology

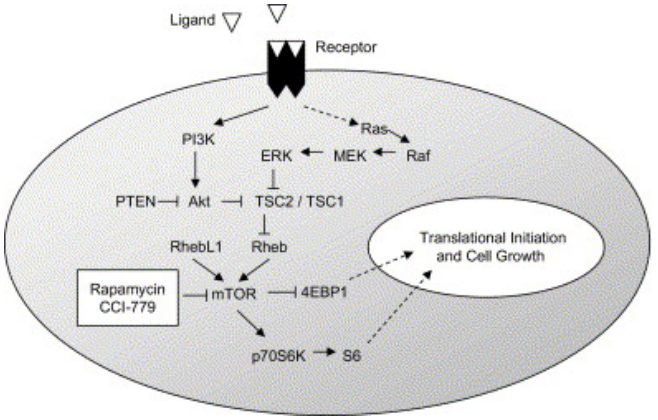
The data deluge of the Zetta Byte (2^{70}) Age



Modern information society and the Information Technology value chain

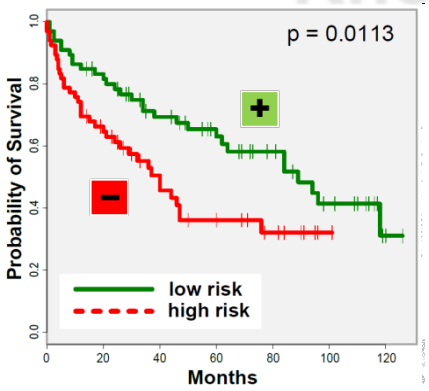




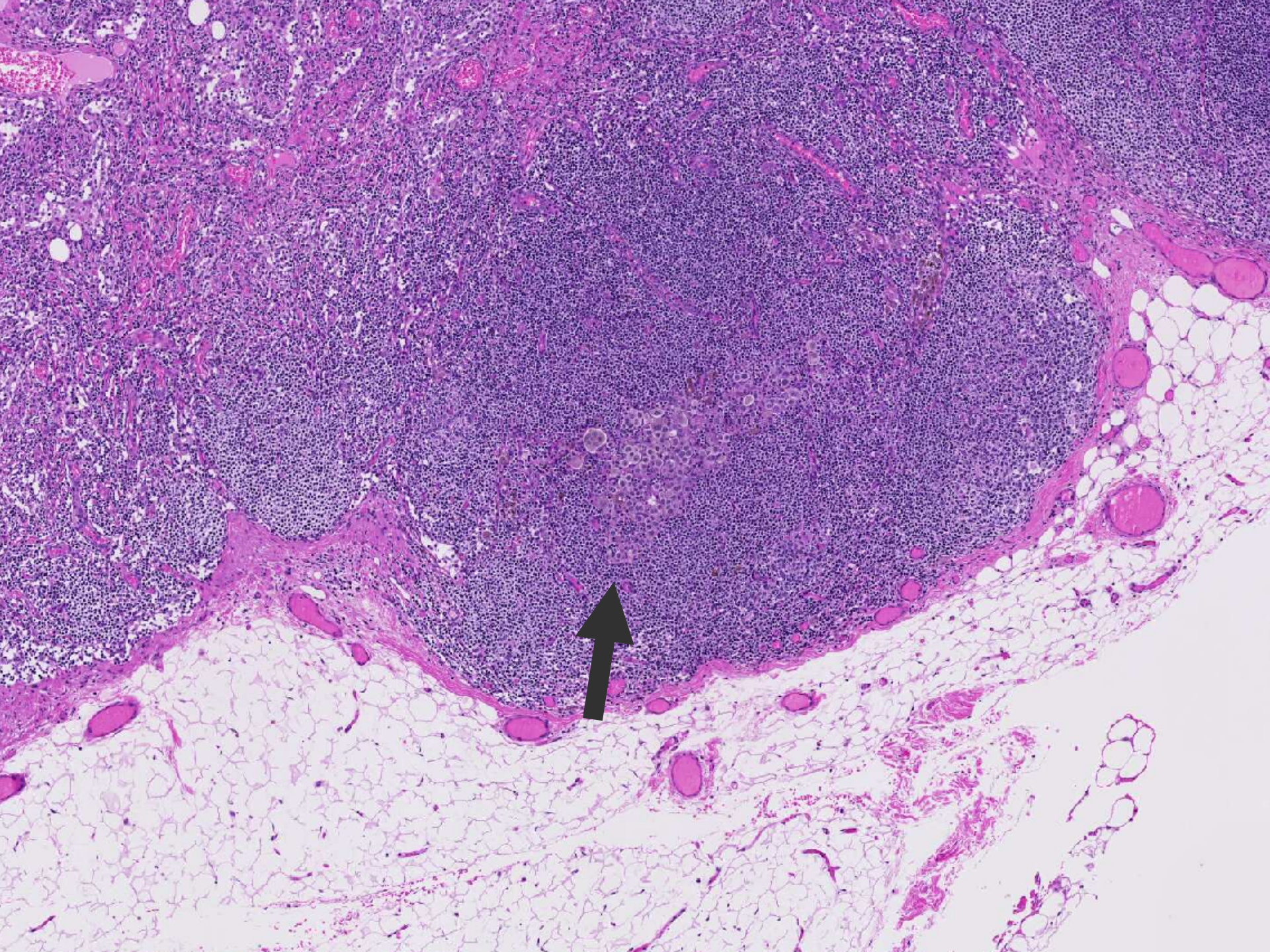


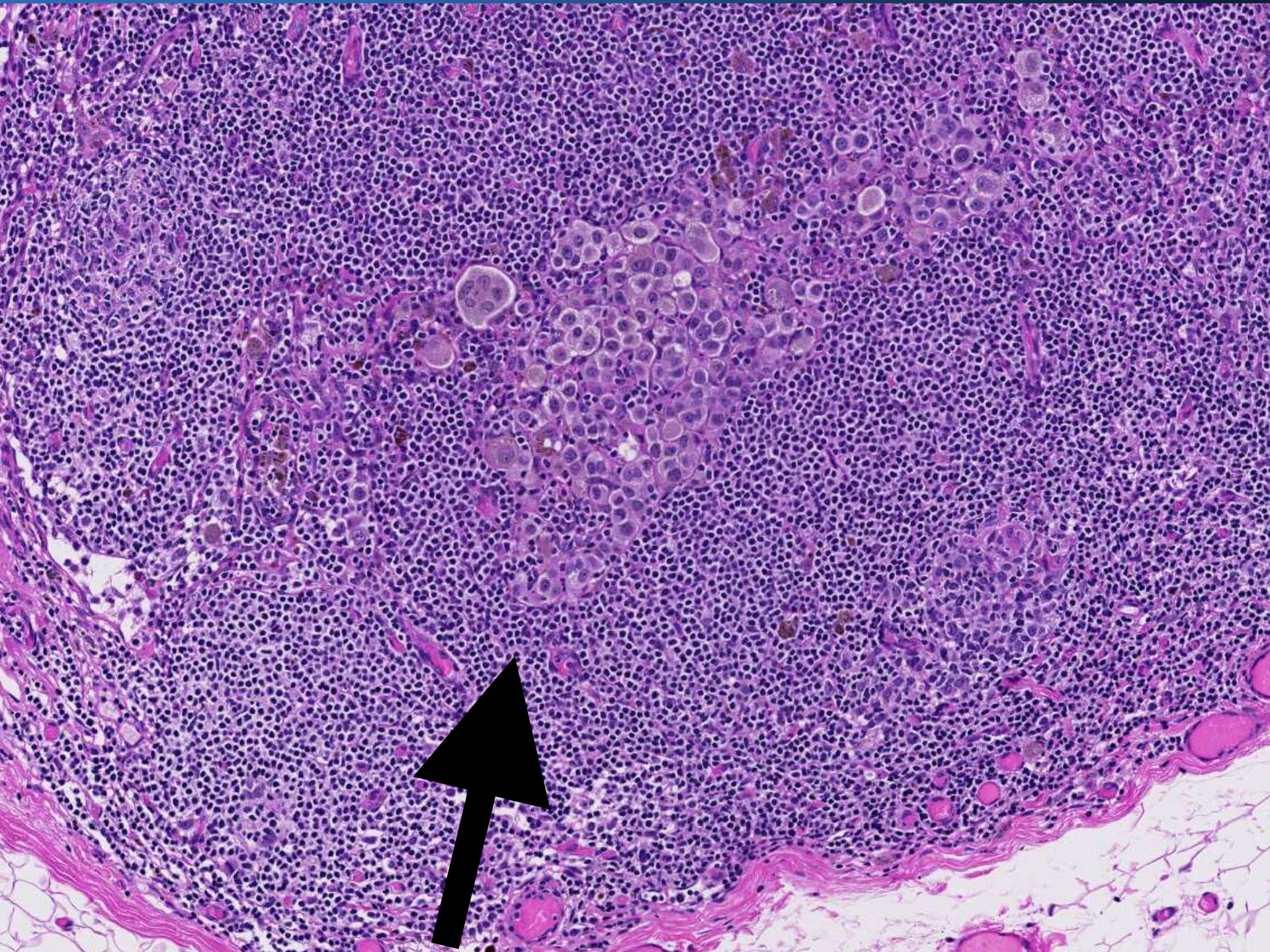
Activation of the mTOR Signaling Pathway in Renal Clear Cell Carcinoma. Robb et al., J Urology 177:346 (2007)

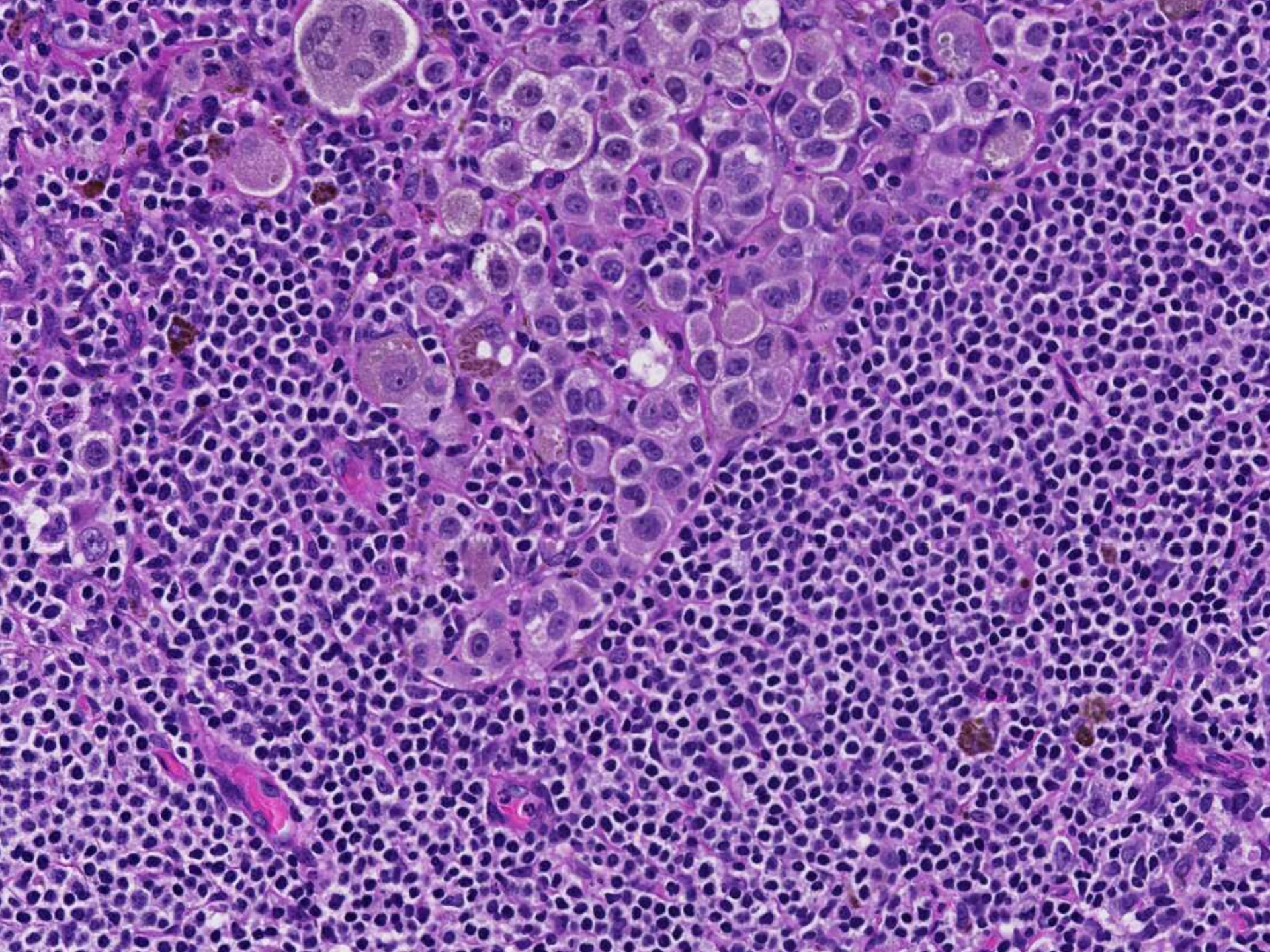
Data → **Information** → **Knowledge** → **Value**



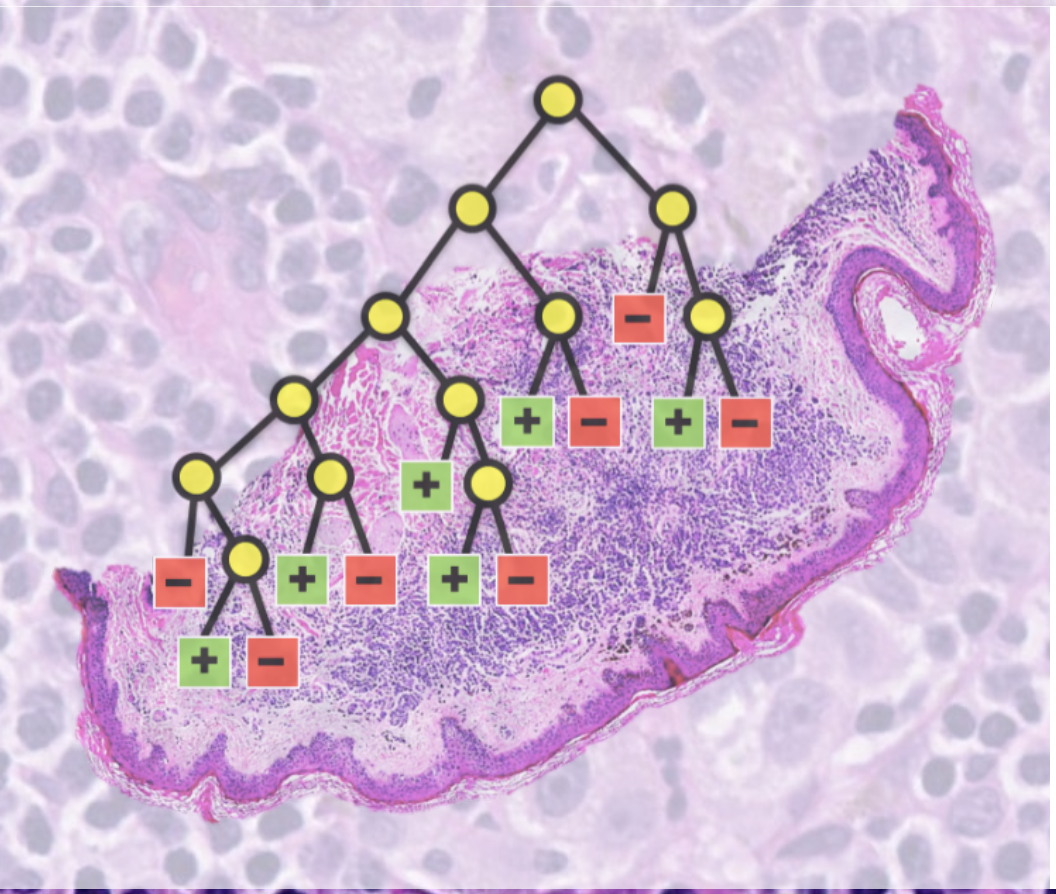
What and where is the valuable information?

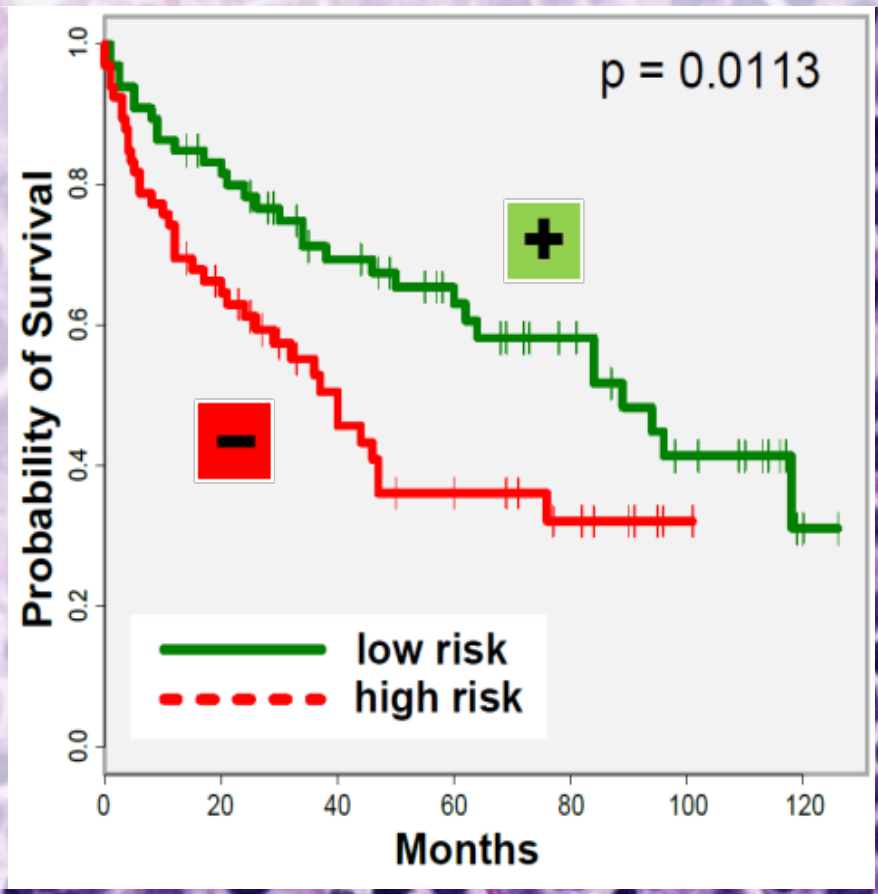






Survival risk: Where is this information?



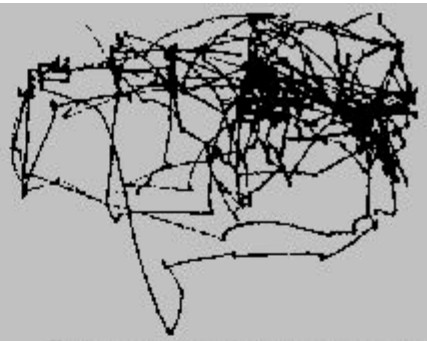


Relevant information in data – where can we find it ?

The relevance of **information** is determined by the task !


Yarbus (1967)


1
Free examination.


2
Estimate material circumstances of the family


3
Give the ages of the people.


4
Surmise what the family had been doing before the arrival of the unexpected visitor.


5
Remember the clothes worn by the people.

Outline

- Relation between **information theory** and **machine learning**
- ***Context sensitive information***
- Coding by approximate optimization: ***Error free communication enables model selection!***
- Examples from cluster model selection
  - Validating spectral clustering
  - Approximate sorting

Scientific strategy for model validation

1. ***Information processing*** *under* ***uncertainty***
   (noisy inputs) yields indistinguishable solutions!
   => ***approximation sets***
2. **Approximation sets** quantize the solution space.
   Consequently, **quantization creates symbols**.
3. **Symbols** define a **code** with an error rate.

=> **Optimality criterion**: Find the maximally informative symbols with zero error rate!

Context and task sensitive information!

- What do we look for?

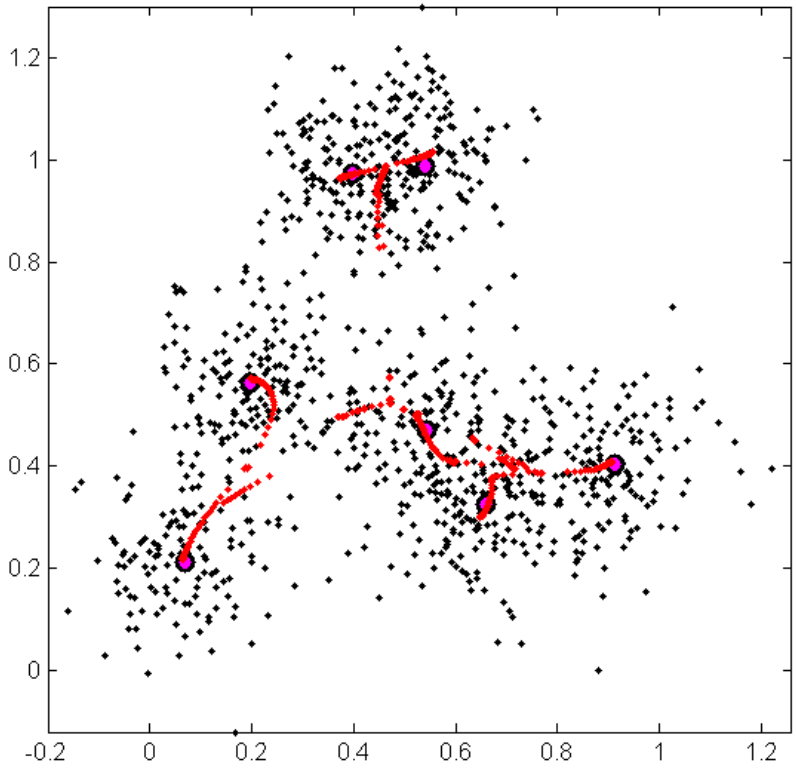
  Solutions out of a **hypothesis class**

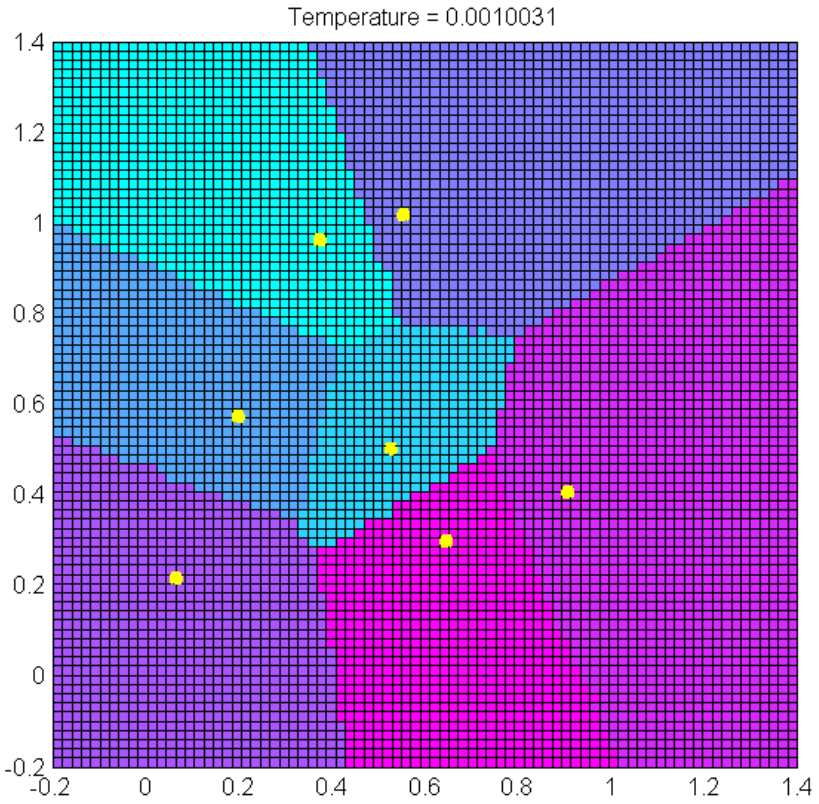
- *Select* **models** (cost functions) to rank hypotheses how well they interpret/explain data.
- ***Validate*** the selected model(s)

⇒***Robust*** and ***expressive*** models are prefered over brittle ones! (***stability vs. informativeness***)

⇒**# of stable bits = context sensitive information**

K-means hypotheses of increasing quality





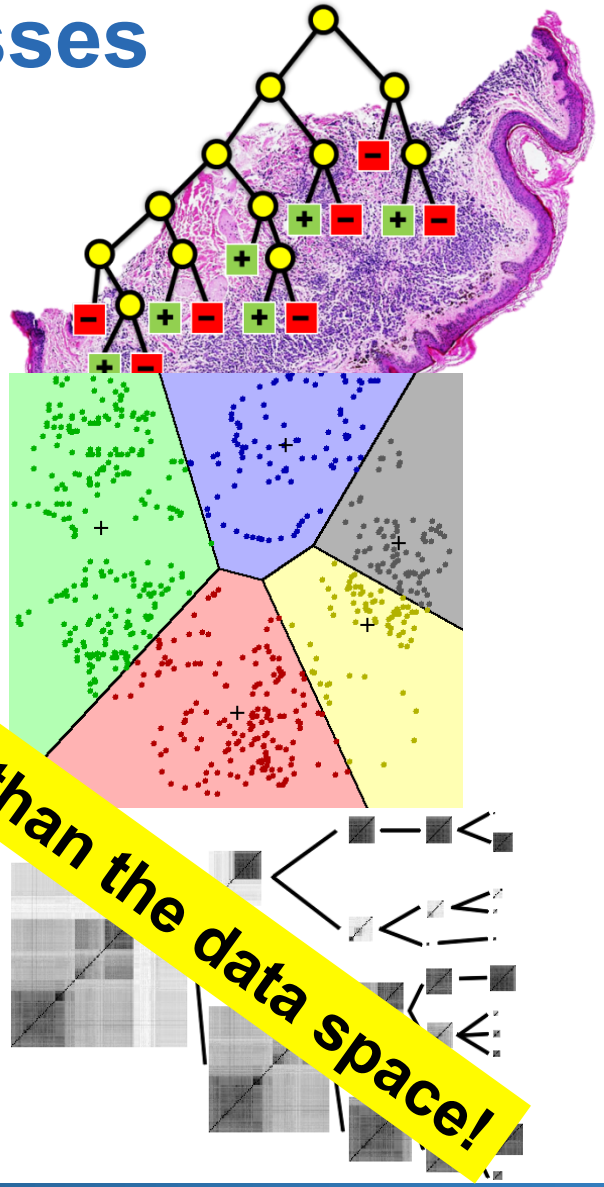
Examples of hypothesis classes

The hypothesis class is often much smaller than the data space!

- **Classifi**

- **Partitions or clust gs**:
  compactness/connectiv osts

- **Trees or dendrograms**:
  partitions with ultrametricity;
  Tree depth? # leaves?



Information Theory & Machine Learning

model validation based on guaranteed approximations
⇒ maximize **mutual information**

- IT-Co…
  - **Code bo… s}** = hypothesis cl…

0000000	0100101	10…	
0001111	0101010	10011…	
0010110	0110011	1010101	
0011001	0111100	1011010	111…

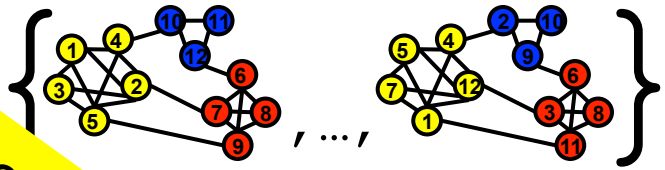
  - Noisy **channel**

    1100110 –> **0**100110

  - Decoder: minimize **Hamming distance**

    |**0**100110–1100110| <
    |**0**100110–1010101|

- Pattern Recognition elements
  - **Set of approximation sets (hypothesis class**



  - **…optimization** problem

  - Decoding b… **…ximate optimization** of …est instance

Approximate Optimization

- Define weights for hypotheses with low costs

$$w_\beta(c, \mathbf{X}) = \exp(-\beta R(c, \mathbf{X})) \quad c \in \mathcal{C}(\mathbf{X})$$

- Total weight of low cost hypotheses (partition fct.)

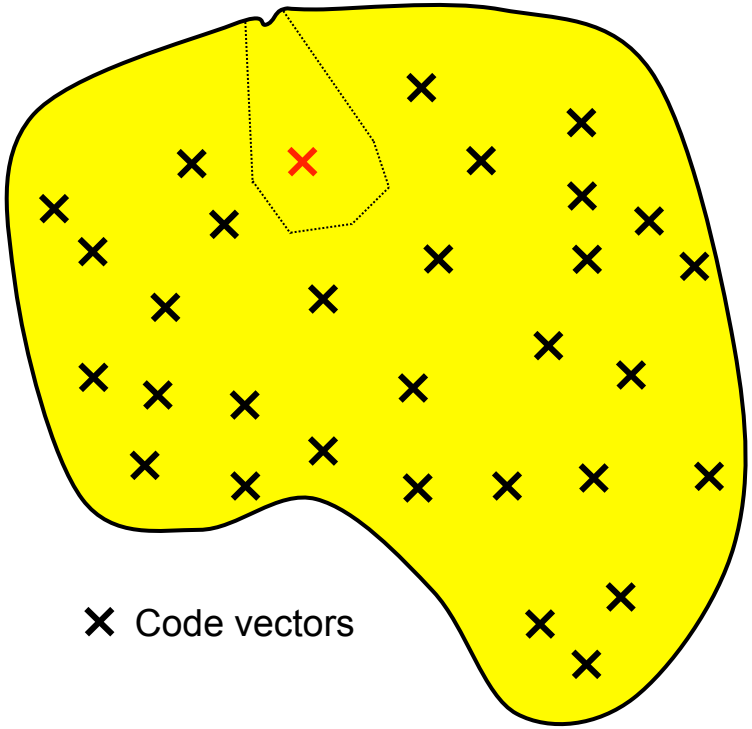
$$Z_\beta(\mathbf{X}) = \sum_{c \in \mathcal{C}(\mathbf{X})} \exp(-\beta R(c, \mathbf{X}))$$

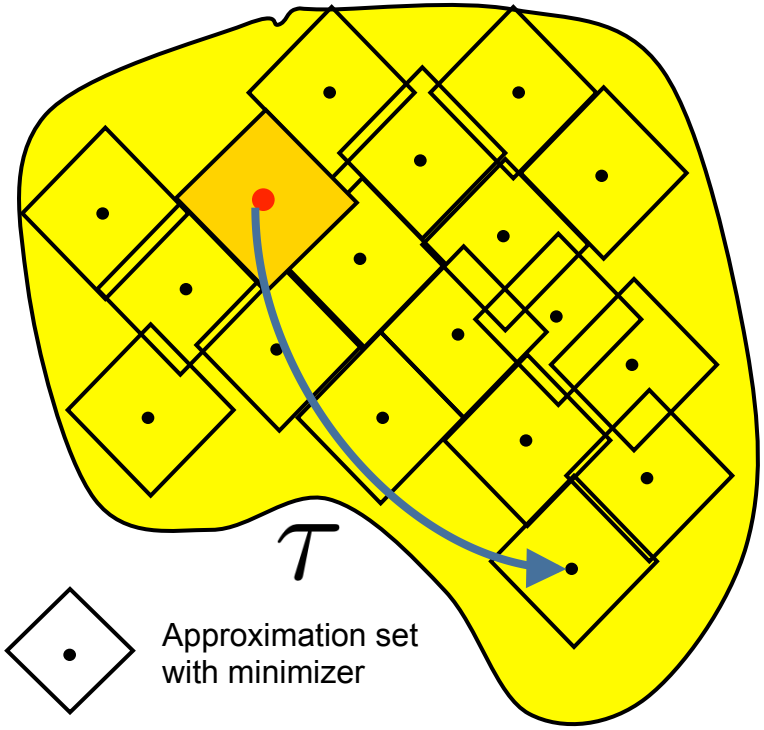
- Special case: **Approximation set**

$$w_\beta(c, \mathbf{X}) = \begin{cases} 1, & R(c, \mathbf{X}) \leq R\left(c^\perp, \mathbf{X}\right) + 1/\beta; \\ 0, & \text{otherwise.} \end{cases}$$

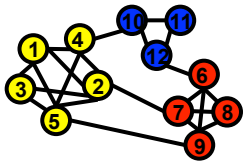
Code problems define approximation sets

- IT: **Space of strings** is partitioned by code vectors
- ML: **Hypothesis class** is partitioned by code problems





Generalization in optimization



Instance space
(data space)

Solution spaces
(Hypothesis class)

$\mathbf{X}^{(1)}$

$\mathbf{X}^{(2)}$

$2^{\binom{n}{2}}$ states
for graph 3-cut

$\mathcal{A}(\mathbf{X}^{(1)})$

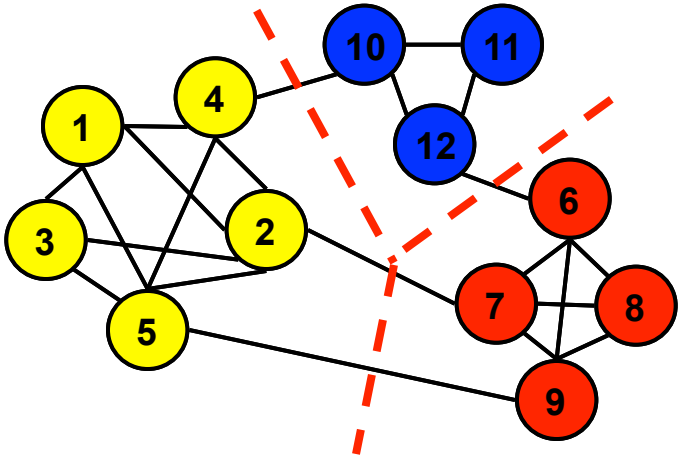
$\mathcal{A}(\tau_2 \circ \mathbf{X}^{(1)})$

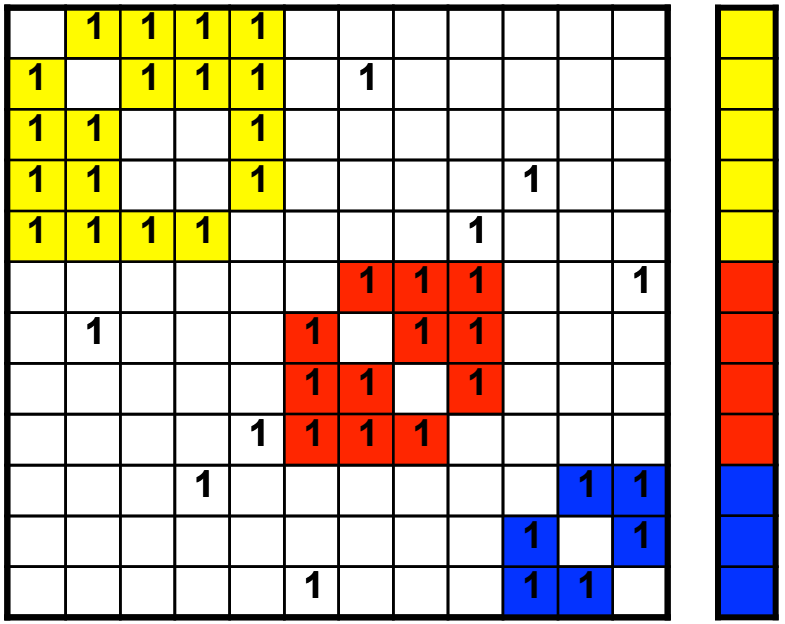
$\mathcal{A}(\tau \circ \mathbf{X}^{(1)})$

3^n solutions
for graph 3cuts

Graph Cut - Clustering in three groups

- **Graph** representation: **vertices** denote **objects**
  edges express (dis)**similarities**
- **Hypothesis class:** all **3-cuts** of a graph

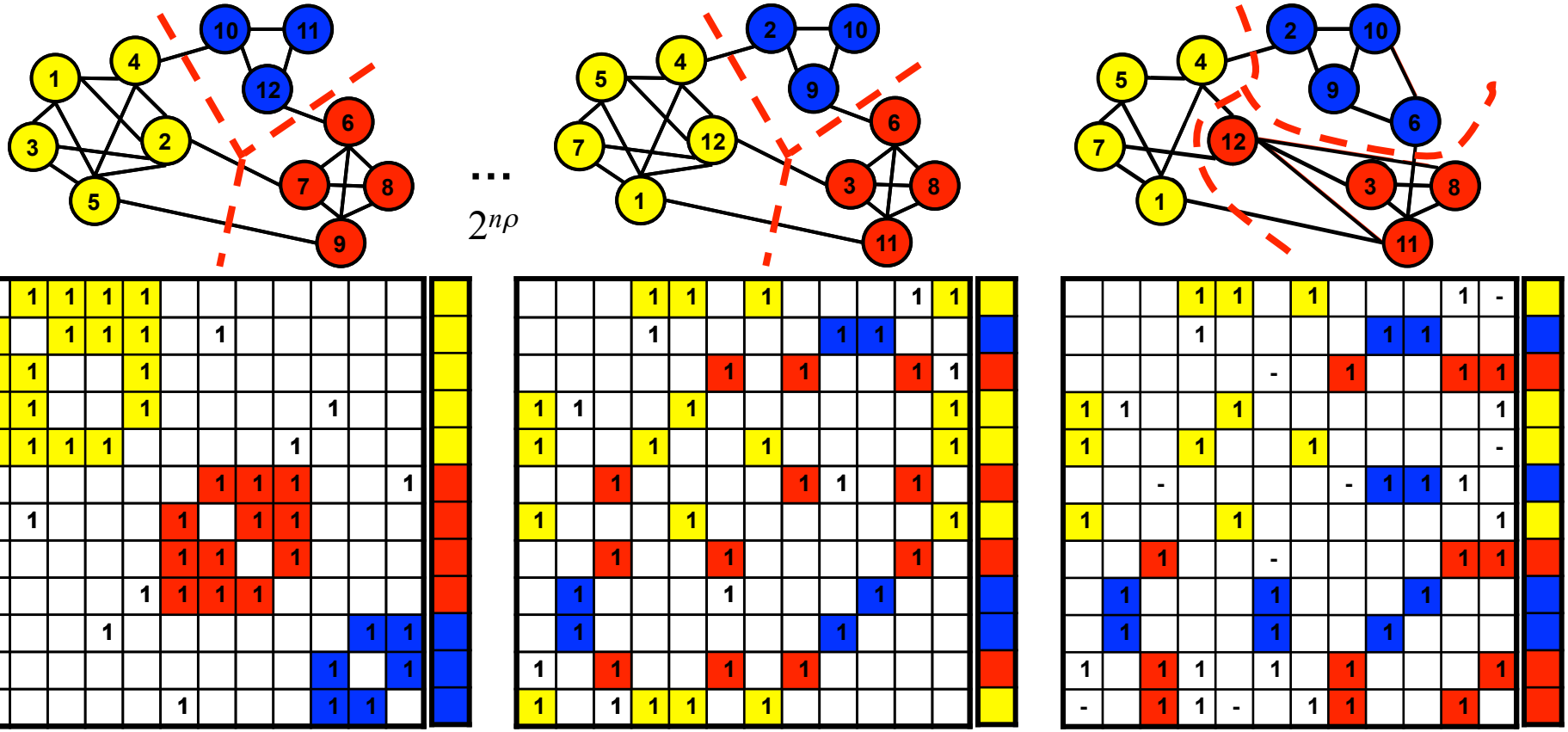




Code problem generation for Graph Cut

graph cut *code* problems

graph cut *test* problem



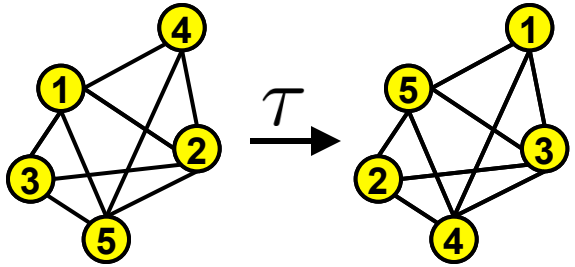
Coding by Code Problems

- **Idea**: define a **code** by transforming a given optimization problem with a **transformation set**

$$\mathcal{T} = \{\tau : R(c, \mathbf{X}) = R(\tau \circ c, \tau_{\mathbf{X}} \circ \mathbf{X})\}$$

$\Rightarrow$ Codebook $\mathbb{T} = \{\tau_i \in \mathcal{T} : 1 \leq i \leq 2^{n\rho}\}$

Combinatorial optimization:
permutation of vertices in a graph



- **Identifiable transformations τ are the <u>messages</u>!**

Coding with Graph Cut approximation sets

define a set of code problems

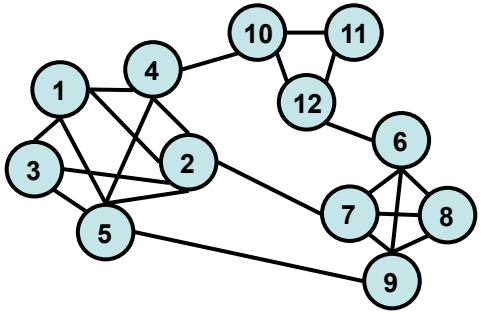
problem generator PG

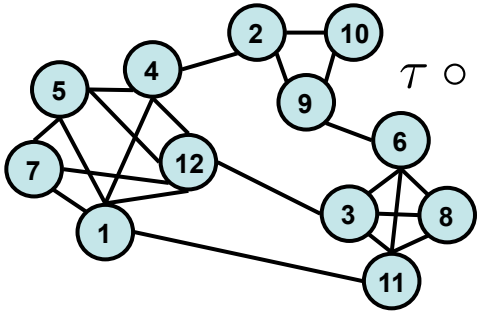
$R(\cdot, \mathbf{X}^{(1)})$ $R(\cdot, \mathbf{X}^{(1)})$

sender

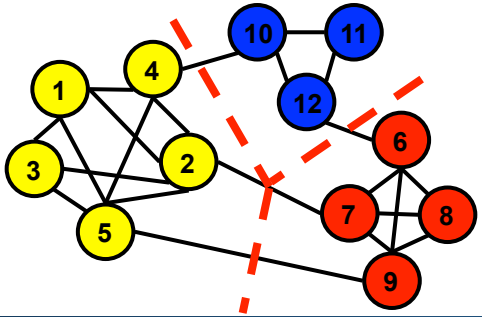
receiver

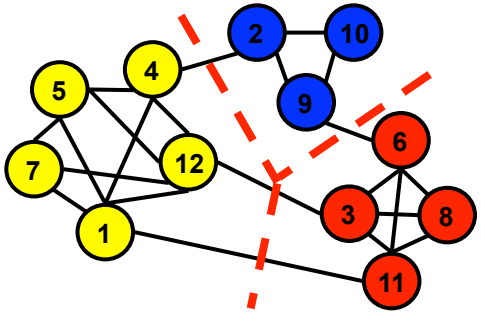
$\{\tau_1, \ldots, \tau_{2^{n\rho}}\}$





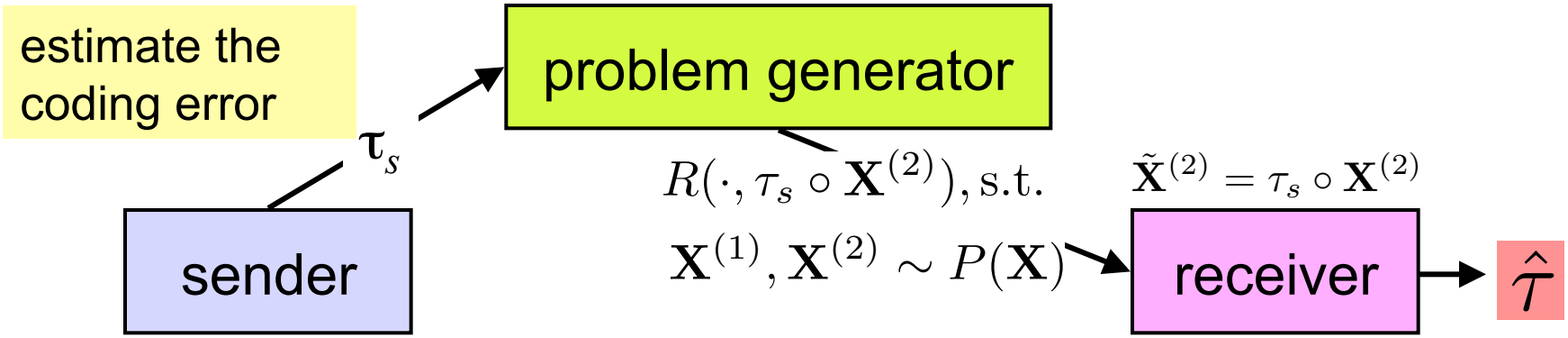
$\tau \circ \mathbf{X}^{(1)}$

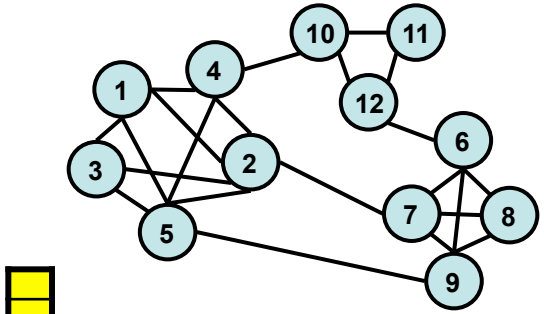


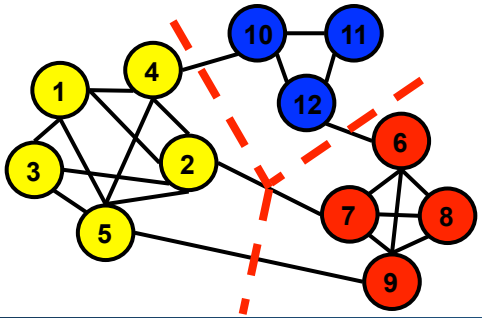


Communication by approximation sets

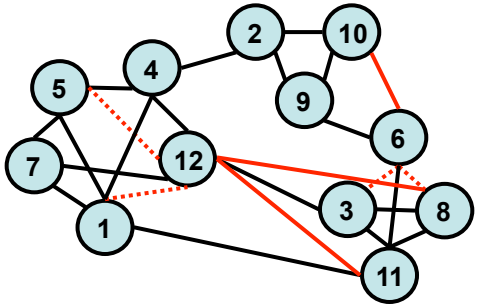
estimate the coding error

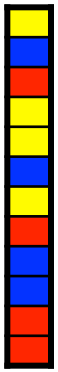


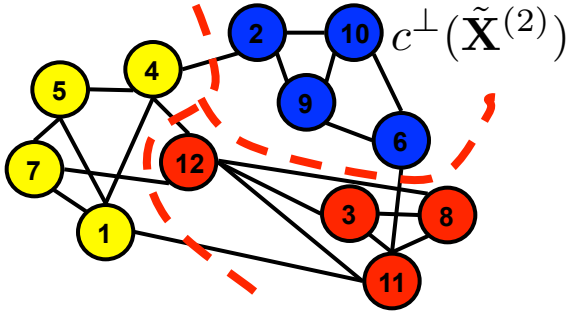




1. Sender sends a permutation index τ_s to problem generator.
2. Problem generator sends a new problem with permuted indices to receiver without revealing τ_s.
3. Receiver identifies the permutation $\hat{\tau}$ by comparing approximation sets.







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 $\lim_{n\to\infty} P(\hat{\tau} \neq \tau_s | \tau_s, \mathbf{X}^{(1)}, \mathbf{X}^{(2)}) = 0$

Error Probability

- Estimate conditional error for data $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}$ given a set of random transformations $\tau \in \mathcal{T}$

$$
\begin{aligned}
P\left(\hat{\tau} \neq \tau_s | \tau_s, \mathbf{X}^{(1,2)}\right) &= P\left(\max_{j \neq s} Z_{\beta,j}^{(1\&2)} > Z_{\beta,s}^{(1\&2)} \middle| \tau_s, \mathbf{X}^{(1,2)}\right) \\
&\underset{\text{bound}}{\leq} \sum_{j \neq s} P\left(Z_{\beta,j}^{(1\&2)}) > Z_{\beta,s}^{(1\&2)}) \middle| \tau_s, \mathbf{X}^{(1,2)}\right) \\
&\leq 2^{n\rho} P\left(Z_{\neq s}^{(1\&2)} > Z_s^{(1\&2)} \middle| \tau_s, \mathbf{X}^{(1,2)}\right) \\
&\underset{\text{ineq.}}{\overset{\text{Markov}}{\leq}} 2^{n\rho} \frac{\mathbb{E}_{\tau_{\neq s}} Z_{\neq s}^{(1\&2)}}{Z_s^{(1\&2)}} \leq 2^{n\rho} \frac{Z_\beta^{(1)} Z_\beta^{(2)}}{|\mathcal{T}| \, Z_s^{(1\&2)}}
\end{aligned}
$$

Asymptotical error-free communication

$$\lim_{n\to\infty} P(\hat{\tau} \neq \tau_s | \tau_s, \mathbf{X}^{(1)}, \mathbf{X}^{(2)}) = 0$$ **is possible if …**

- … rate ρ is bounded by mutual information $\hat{\mathcal{I}}_\beta(\tau_s, \hat{\tau})$

$$\rho < \hat{\mathcal{I}}_\beta(\tau_s, \hat{\tau}) \equiv \frac{1}{n} \log_2 \frac{|\mathcal{T}|\, Z_\beta^{(1\&2)}}{Z_\beta^{(1)}\, Z_\beta^{(2)}}$$

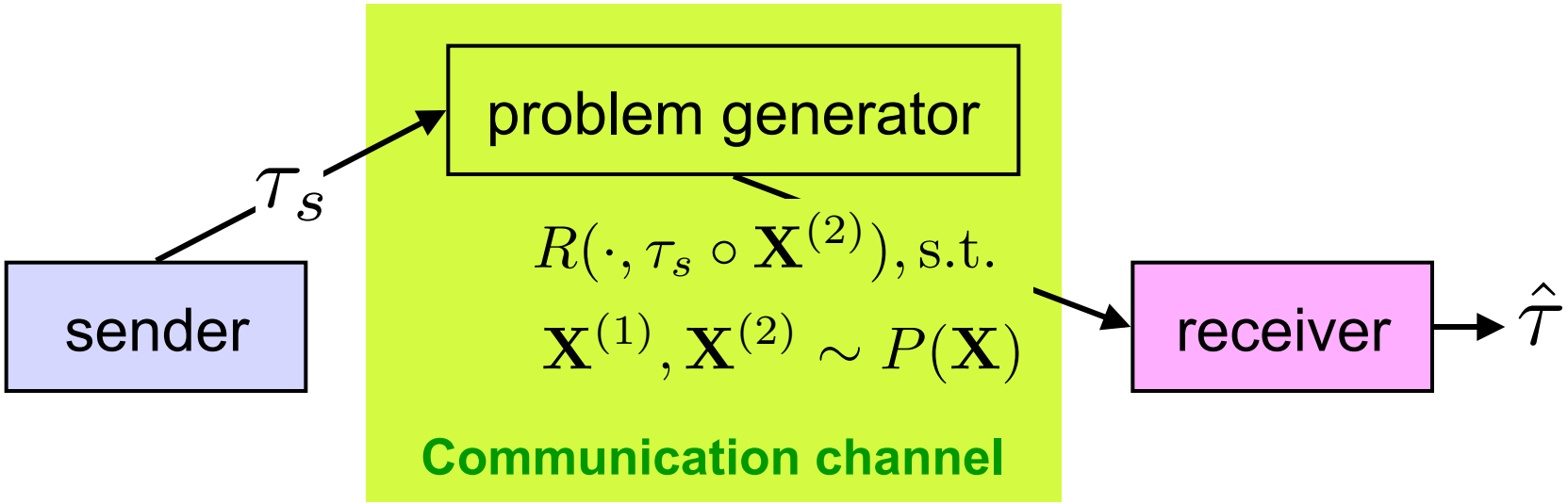
$$= \frac{1}{n} \left(\log_2 \frac{|\mathcal{T}|}{Z_\beta^{(1)}} + \log_2 \frac{|\mathcal{C}^{(2)}|}{Z_\beta^{(2)}} - \log_2 \frac{|\mathcal{C}^{(2)}|}{Z_\beta^{(1\&2)}} \right)$$

Bound calculation involves partition functions for individual and joint costs

$$Z_\beta^{(\nu)} = \sum_{c \in \mathcal{C}(\mathbf{X}^{(\nu)})} \exp(-\beta R(c, \mathbf{X}^{(\nu)})), \quad \nu = 1, 2$$

$$Z_\beta^{(1\&2)} = \sum_{c \in \mathcal{C}(\mathbf{X}^{(1)})} \exp\left(-\beta\big(R(c, \mathbf{X}^{(1)}) + R(c, \mathbf{X}^{(2)})\big)\right)$$

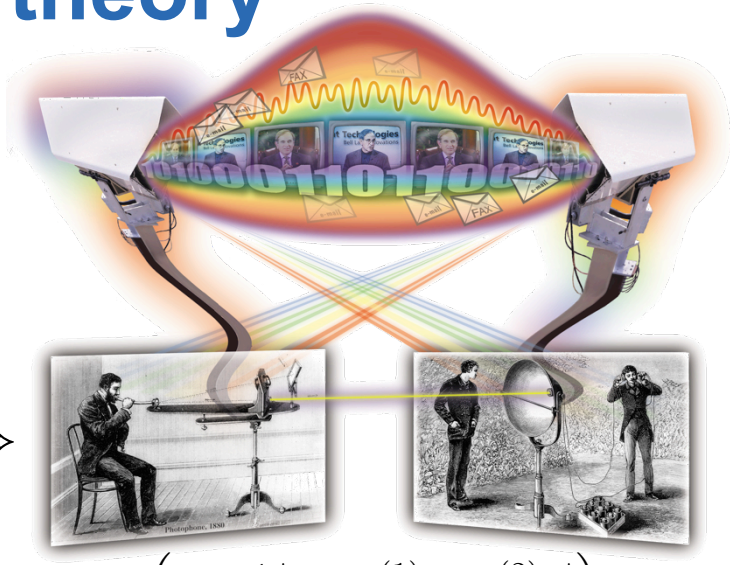
Model Selection by Maximization of Approximation Capacity



- Optimize the communication channel w.r.t. precision γ (β) and cost function $R(.,.)$
- Maximize Mutual Information

ASC for binary channel consistent with Shannon information theory

- Hypothesis class: set of binary strings $\xi^{(1)}, \xi^{(2)} \in \{-1, 1\}^n$



- Communication: $\xi^{(1)} \Rightarrow \Rightarrow \xi^{(2)}$

$$\left(\delta = \tfrac{1}{n}\left|\{i : \xi^{(1)} \neq \xi^{(2)}\}\right|\right)$$

- Costs of string s: Hamming distance

$$R(s, \xi^{(1)}) = \sum_{i=1}^{n} \mathbb{I}_{\{s_i \neq \xi_i^{(1)}\}}$$

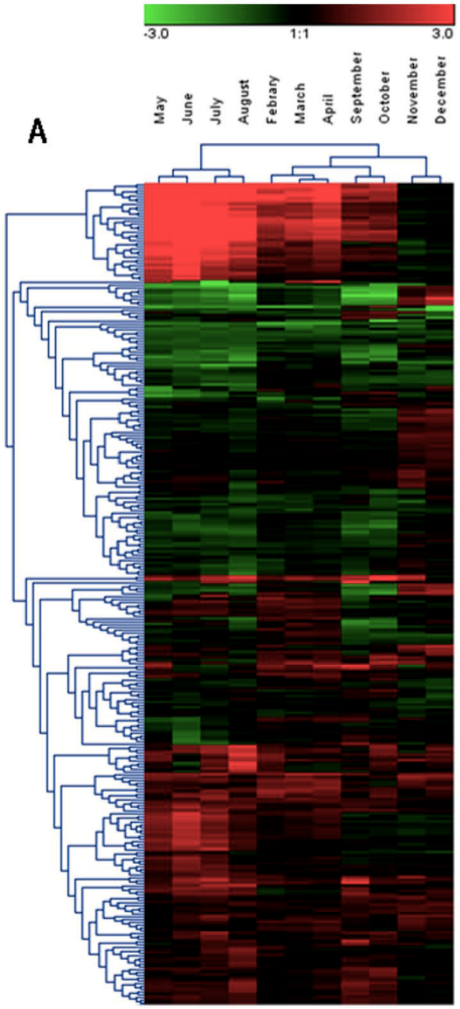
- Mutual information: $\mathcal{I}_\beta = \ln 2 + (1-\delta)\ln\cosh\beta - \ln(\cosh\beta + 1)$

Channel capacity of BSC

$$\text{for} \quad (*)\ \frac{d\mathcal{I}_\beta}{d\beta} = 0 \qquad \overset{(*)}{=} \quad \ln 2 + (1-\delta)\ln(1-\delta) + \delta \ln \delta$$

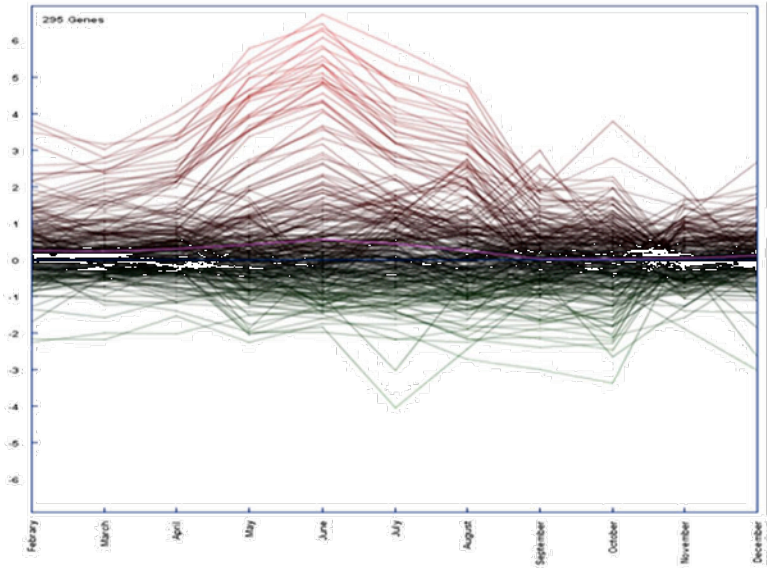
Validation of relational clustering

Joint work with
M. H. Chehreghani & **A.G. Busetto**



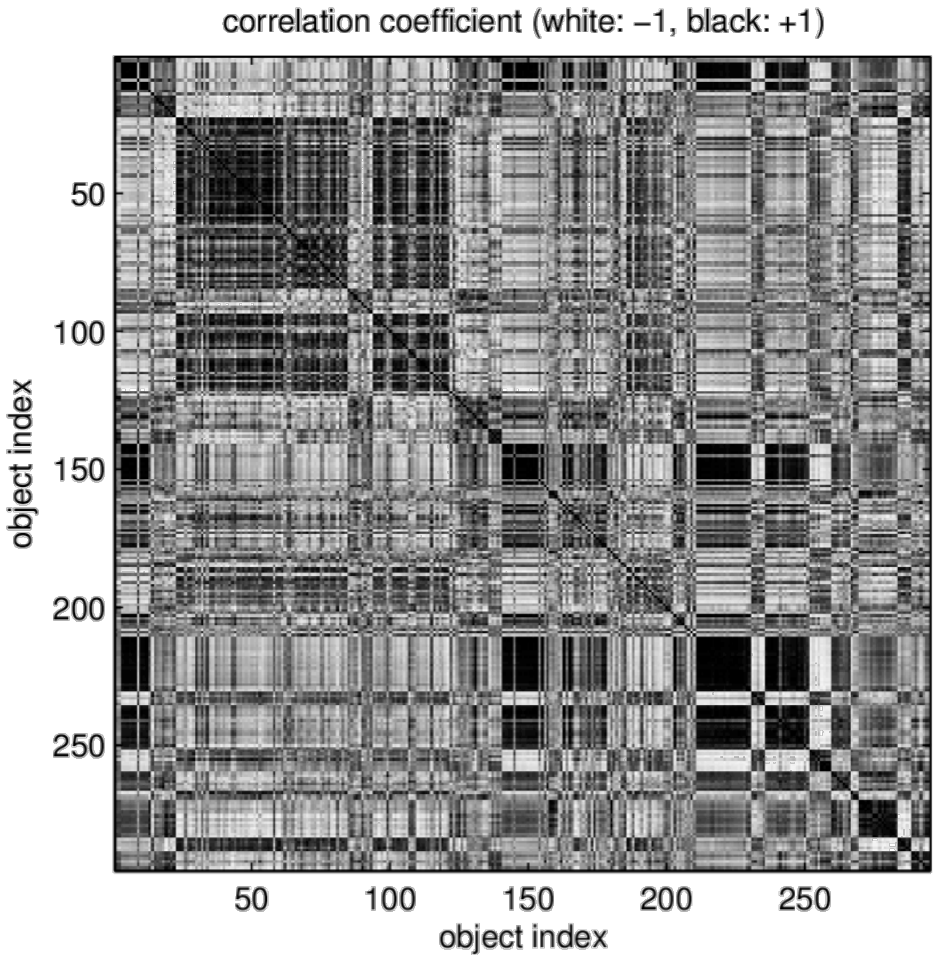
Gene Expression Rhythms

(A) in the mussel mytilus galloprovincialis (Lam.) across an Annual Cycle



Correlations between gene expressions

- **Gene expression time series** are compared by Pearson correlation coefficient
- Relational clustering by *normalized Cut* (Ncut), *correlation clustering* (CC) or *pairwise clustering* (PC)



Validating Relational clustering

- Cost function for pairwise clustering

$$R^{pc}(c, \mathbf{X}) = -\sum_{k=1}^{K} \mathcal{G}_k \sum_{(i,j)\in\mathcal{E}_{kk}} \frac{X_{ij}}{|\mathcal{E}_{kk}|}$$

- Normalized cut

$$R^{nc}(c; W) = \sum_{v\leq k} \left(\frac{cut(\mathcal{G}_v, \mathcal{V} \setminus \mathcal{G}_v)}{assoc(\mathcal{G}_v, \mathcal{V})} \right)$$

- Correlation clustering

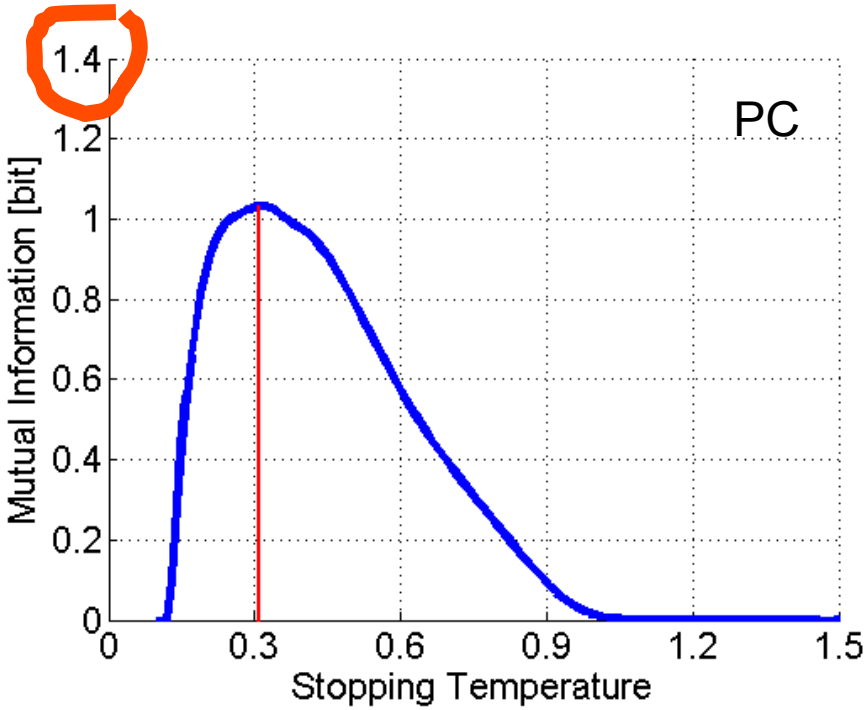
$$R^{cc}(c, \mathbf{X}) = \frac{1}{2} \sum_{k,l=1}^{K} \sum_{(i,j)\in\mathcal{E}_{kl}} \left((-1)^{\delta_{kl}} X_{ij} + |X_{ij}| \right)$$

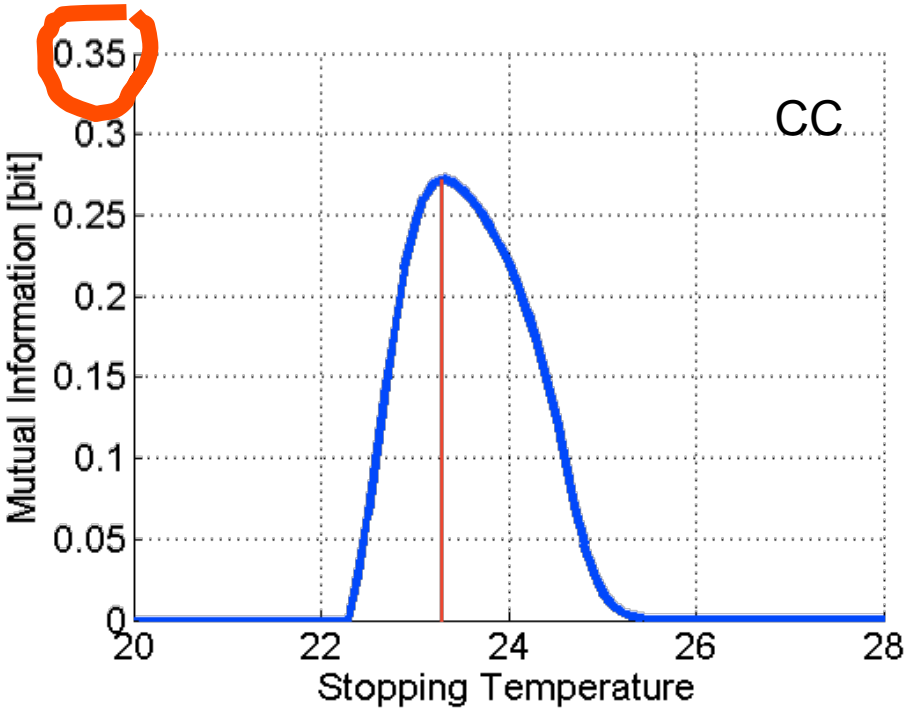
Defs.: $\mathcal{G}_k = \{i : c(i) = k\}, \quad \mathcal{E}_{kl} = \{(i,j) : c(i) = k \ \wedge \ c(j) = l\}$

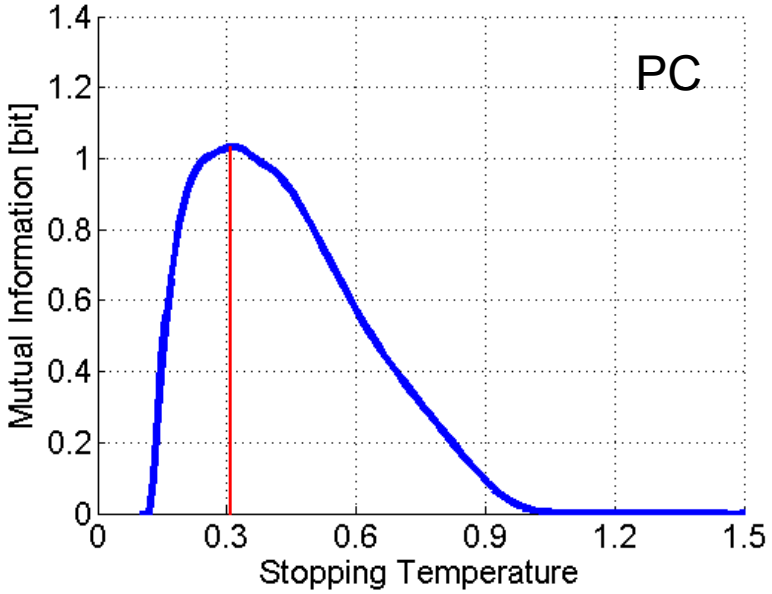
Information of Spectral Clusterings

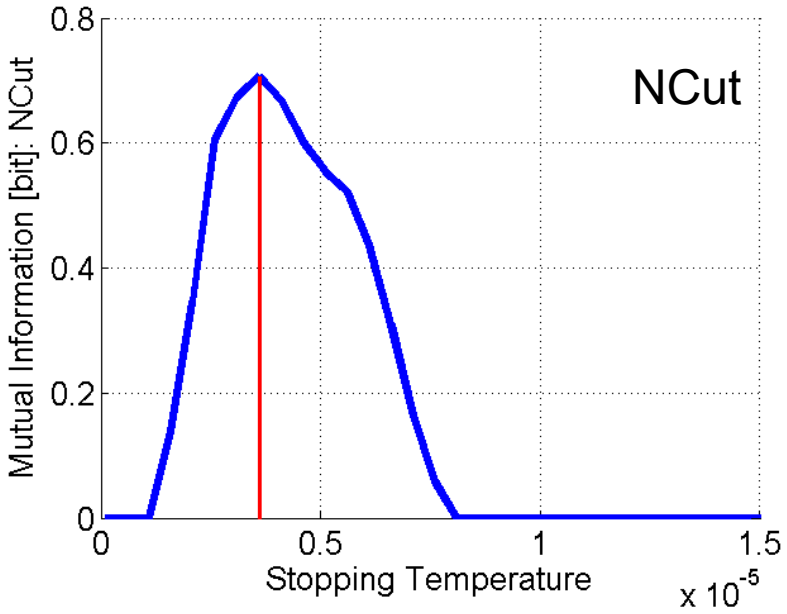
$$\hat{\mathcal{I}}_\beta(\tau_s, \hat{\tau}) \equiv \frac{1}{n} \log_2 \frac{|\mathcal{T}| \, Z_\beta^{(1\&2)}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})}{Z_\beta(\mathbf{X}^{(1)}) \, Z_\beta(\mathbf{X}^{(2)})}$$

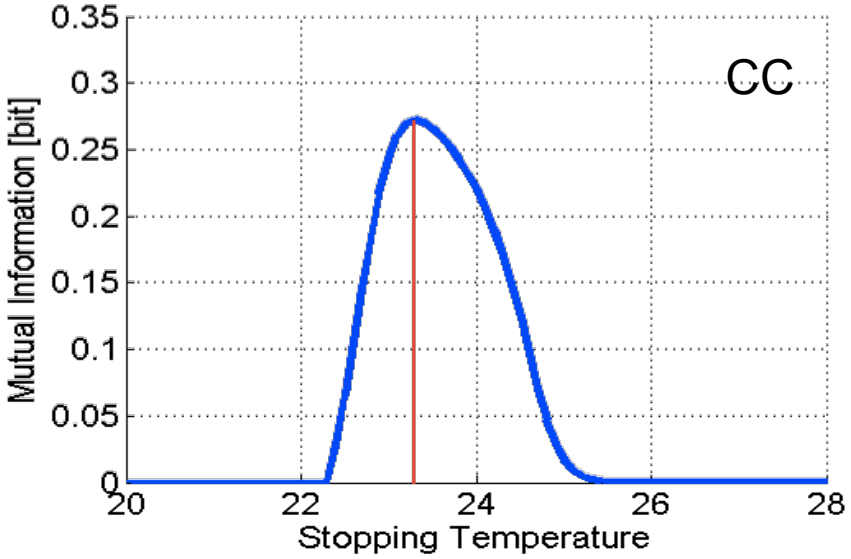
- Pairwise Clustering extracts 30% more information than NCut and 3.5 times more than Correlation Clustering

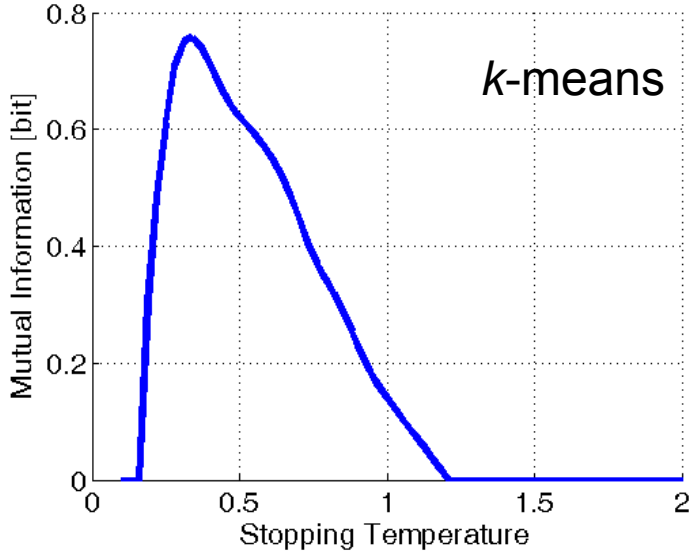




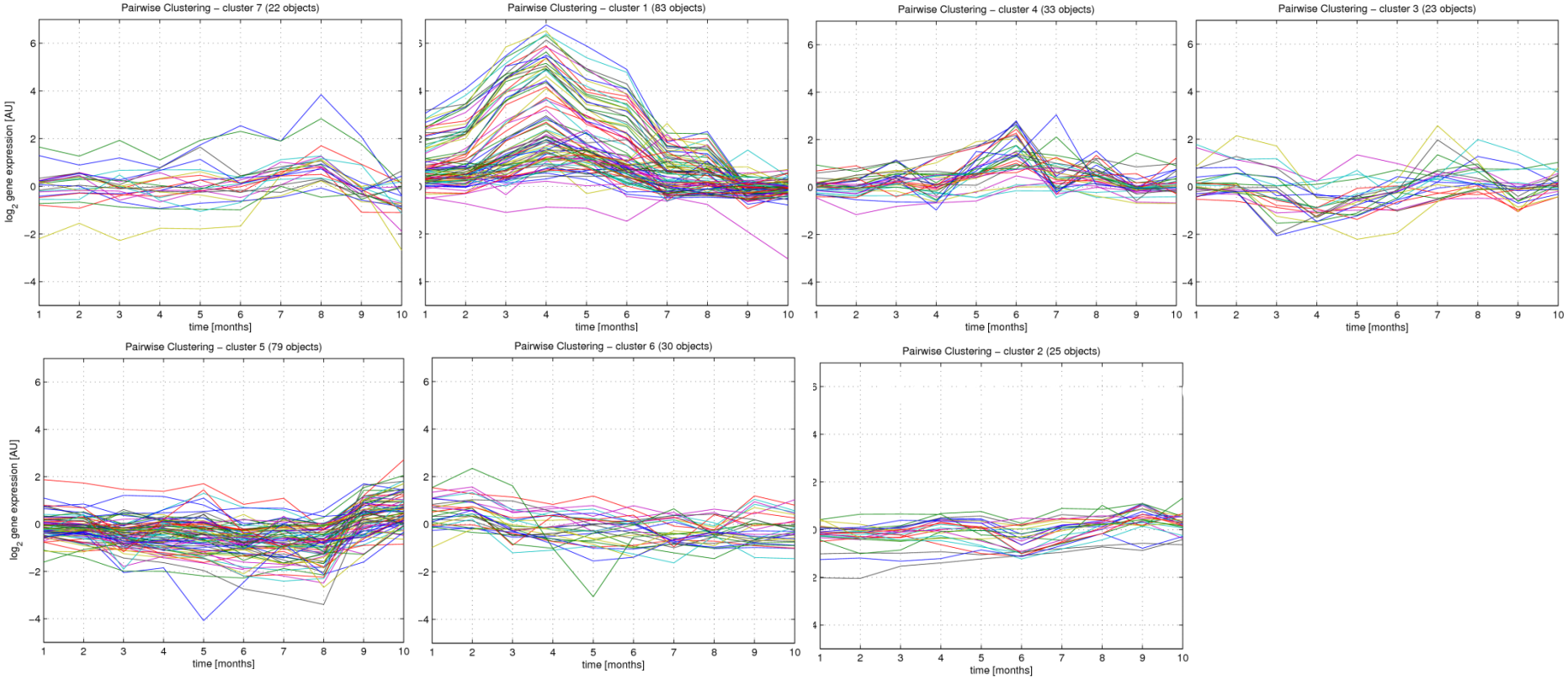

PC
Mutual Information [bit]
Stopping Temperature
0
0.2
0.4
0.6
0.8
1
1.2
1.4
0.3
0.6
0.9
1.2
1.5


NCut
Mutual Information [bit]: NCut
Stopping Temperature
x 10^-5
0
0.2
0.4
0.6
0.8
0.5
1
1.5

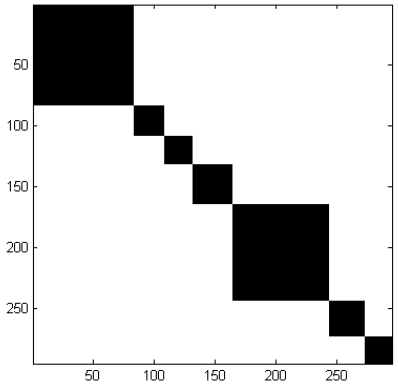

CC
Mutual Information [bit]
Stopping Temperature
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
20
22
24
26
28


k-means
Mutual Information [bit]
Stopping Temperature
0
0.2
0.4
0.6
0.8
0.5
1
1.5
2

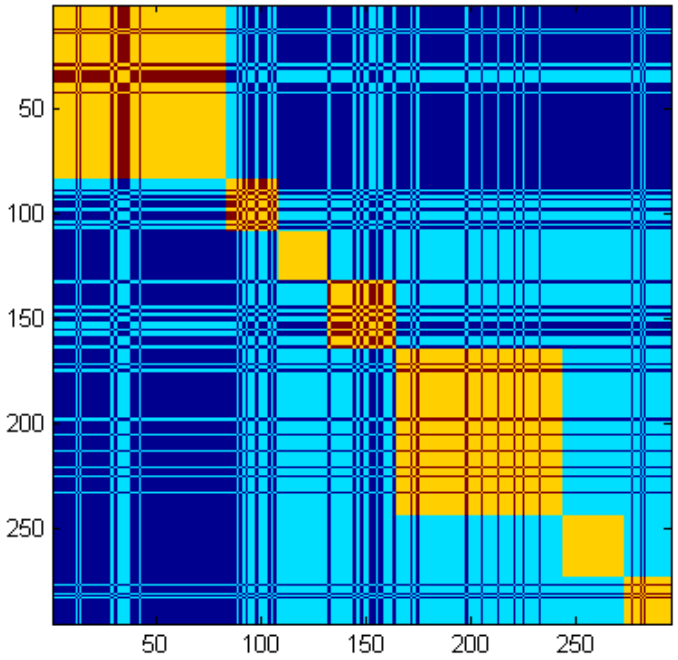
Visualization of PC clusters



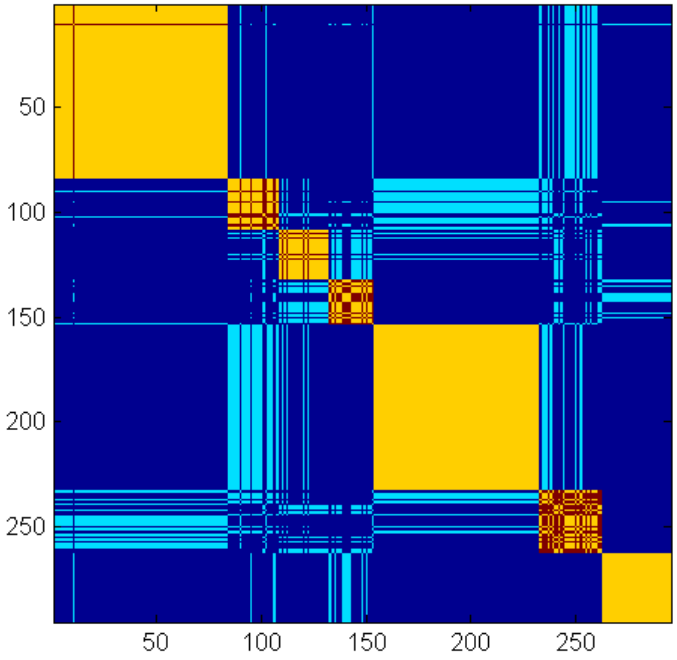
Consistency of Clusterings



- objects (i,j) are clustered together by PC & CC,
- objects (i,j) are clustered differently by PC & CC,
- objects (i,j) in same cluster by PC and different by CC,
- objects (i,j) in different clusters by PC and same by CC.



PC vs. CC

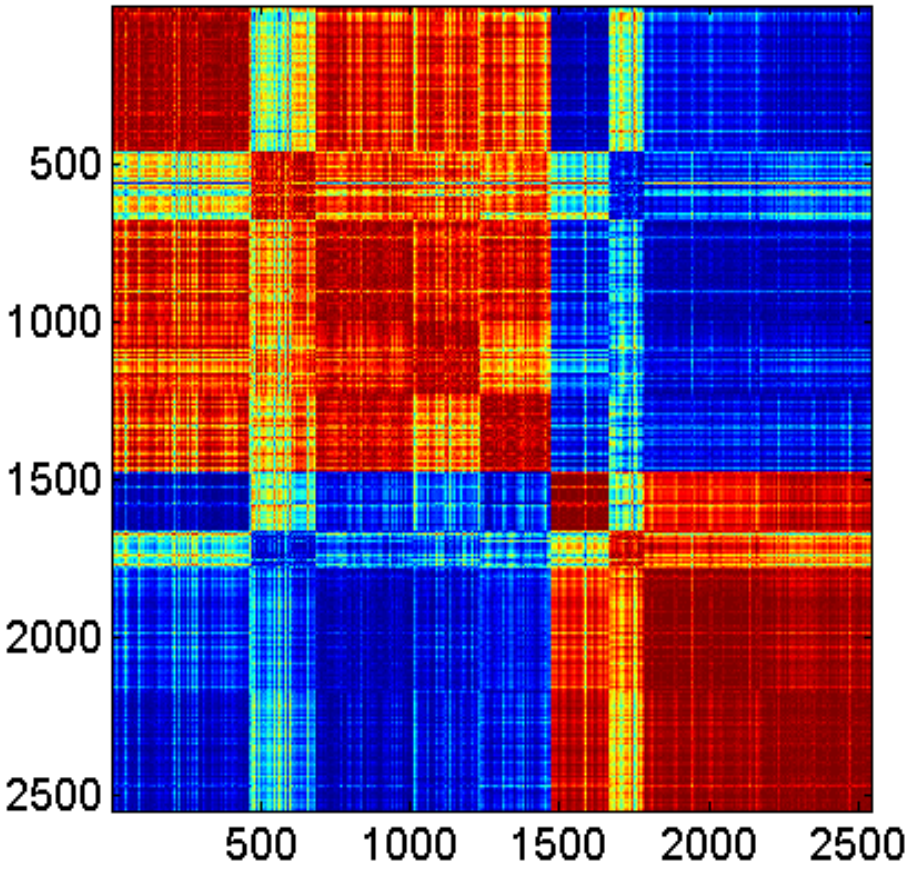


PC vs. NCut

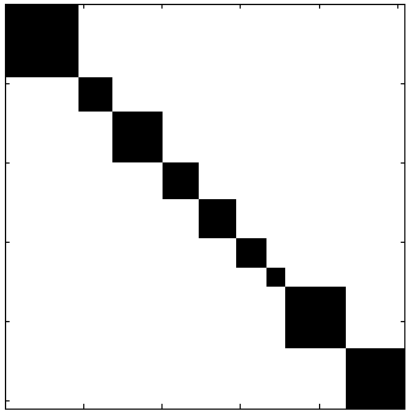
Clustering Yeast Genes for Metabolism

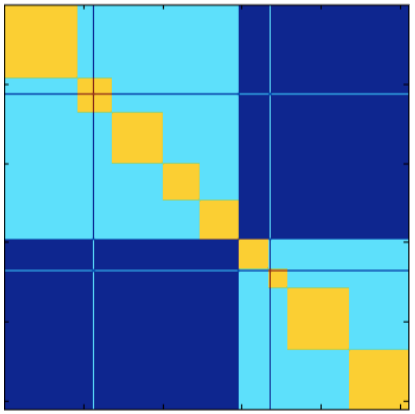
- gene expression values of approx. 2500 yeast genes are clustered to distinguish **dynamic behaviors** and **mechanistic interactions controlling cell growth**;
- Genes are measured at 18 different time points under three biological conditions (treatment with *proline, glutamine* and *rapamycin*).

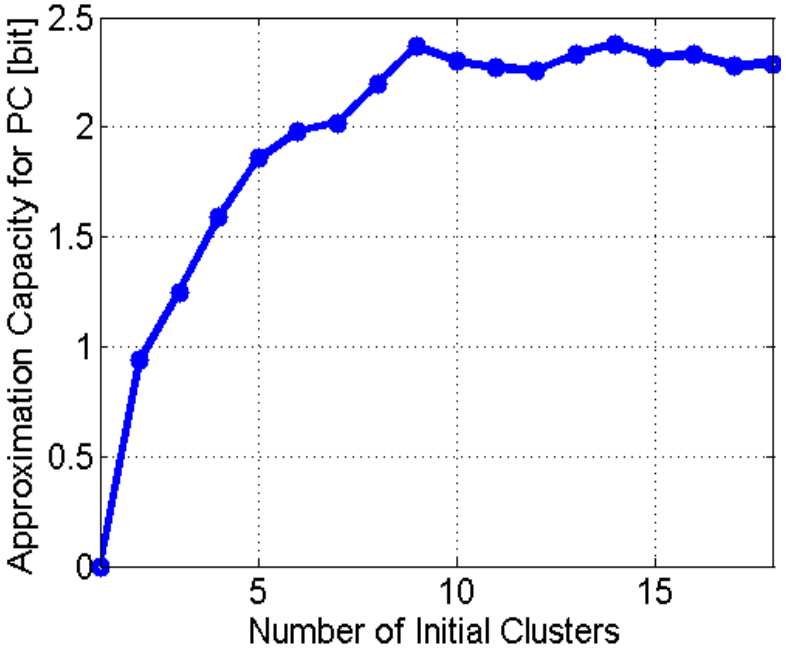
Correlation matrix of gene expressions

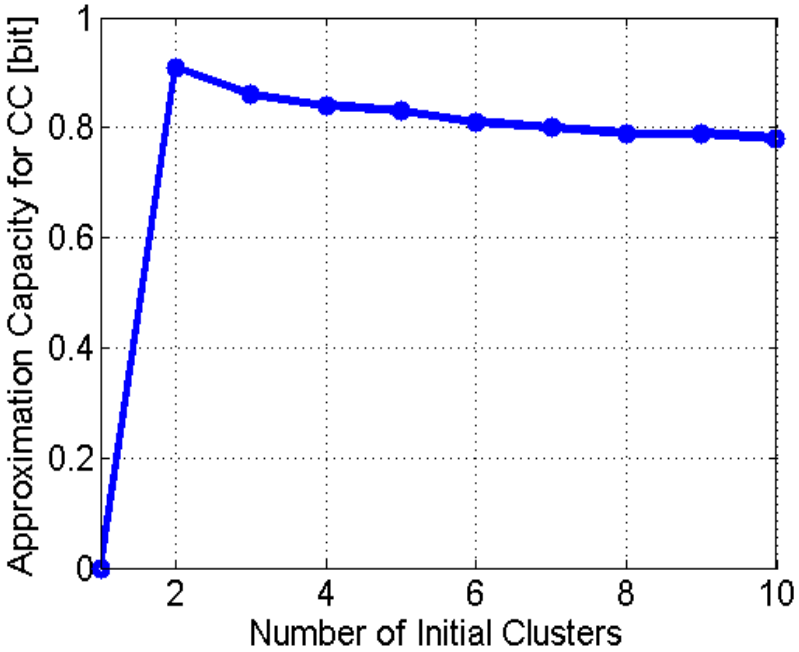


Approximation Capacities for CC and PC









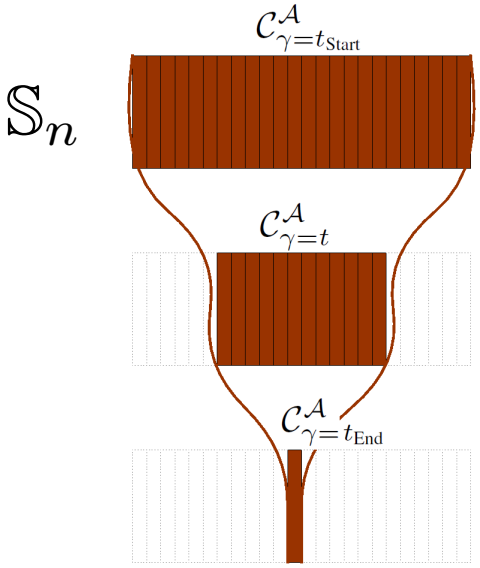
Joint work with
L.M. Busse & M.H. Chehreghani

Sorting in noisy conditions

- *Sorting items into an order*

  is **restricting the set of rankings**

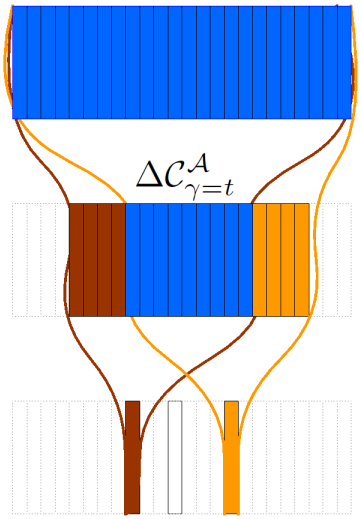
  based on pairwise comparisons



- New method to

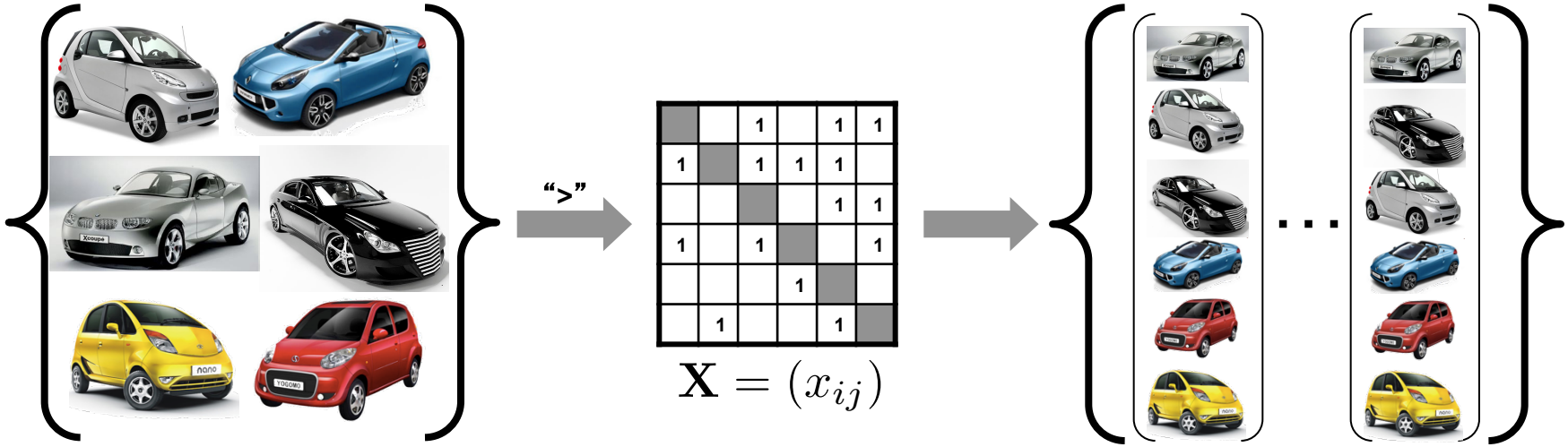
  optimally sort data in noisy conditions,

  i.e. pairwise comparisons can be affected by noise!



Cardinality of the
approximation set is a function
of the noise level in the data

Approximate Sorting

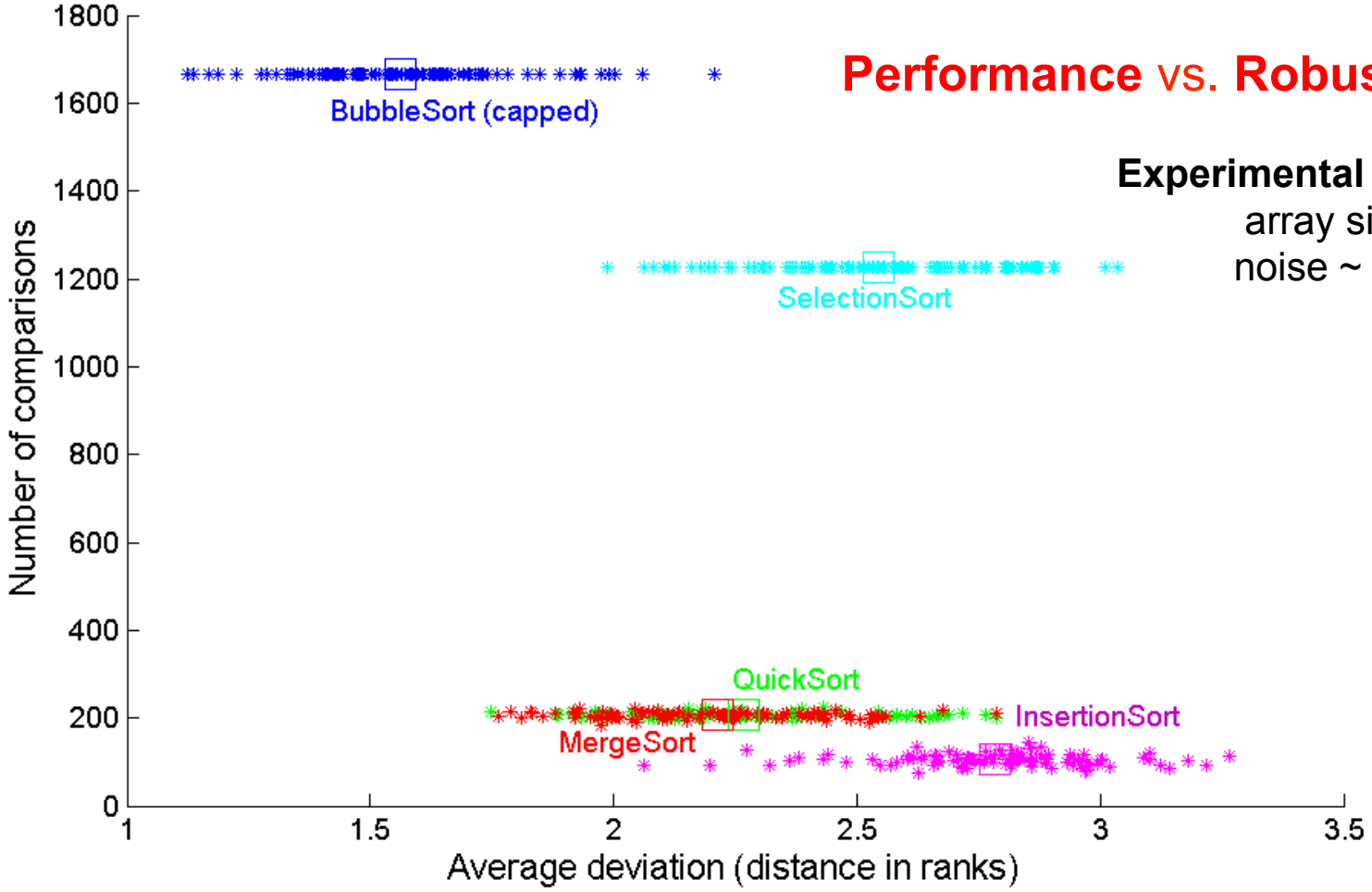


- Among all permutations π, find the one that **orders the items best**!

$$\mathcal{R}^{\text{Sorting}}(\pi|\mathbf{X}) = \sum_{i,j} x_{ij} \mathbb{I}_{\{\pi_i > \pi_j\}}$$

*(This cost function counts the number of disagreements between the **provided pairwise comparisons** and the **pairwise comparisons induced by a proposal ranking** from the hypothesis class.)*

Pareto-Optimality of Sorting Algorithms



Performance vs. **Robustness**

Experimental setting:
array size n=50
noise ~ Ber(0.1)

Approximate Sorting

- *Optimally balance the tradeoff between*

  extracting as much reliable partial order information

  as possible from the data,

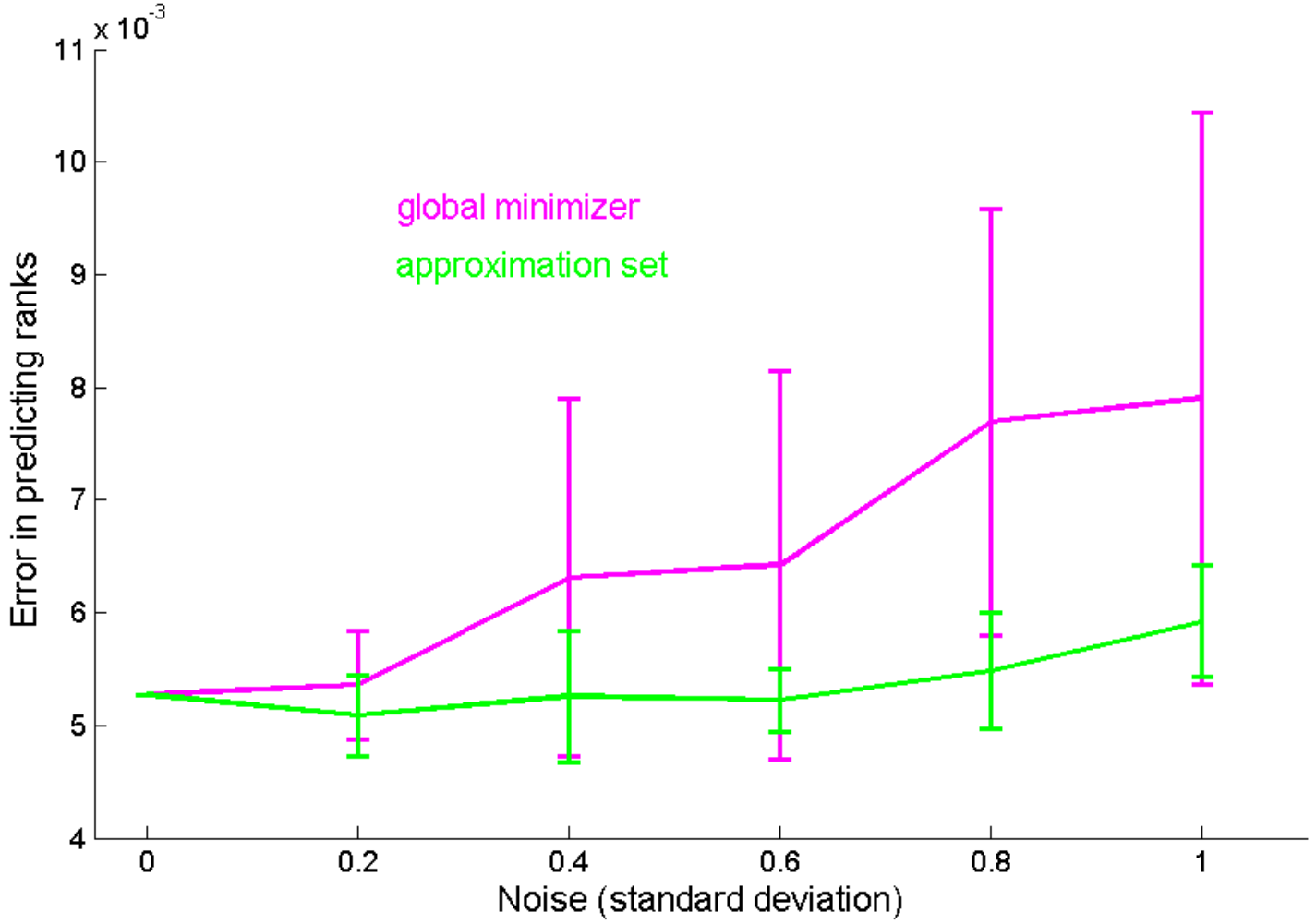
  while being **maximally robust against the noise type**

  actually present in the data.

$$\mathcal{I}_\beta = \frac{1}{n}\log|\mathcal{T}| + \frac{1}{n}\sum_{i=1}^{n}\log\sum_{k=1}^{n}\exp\left(-\beta(\mathcal{E}_{ik}^{(1)}+\mathcal{E}_{ik}^{(1)})\right)$$
$$-\frac{1}{n}\sum_{i=1}^{n}\log\left(\sum_{k=1}^{n}\exp\left(-\beta\mathcal{E}_{ik}^{(1)}\right)\sum_{k'=1}^{n}\exp\left(-\beta\mathcal{E}_{ik'}^{(2)}\right)\right)$$

Prediction Performance

e.g. predicting soccer team ranks based on records of wins and losses



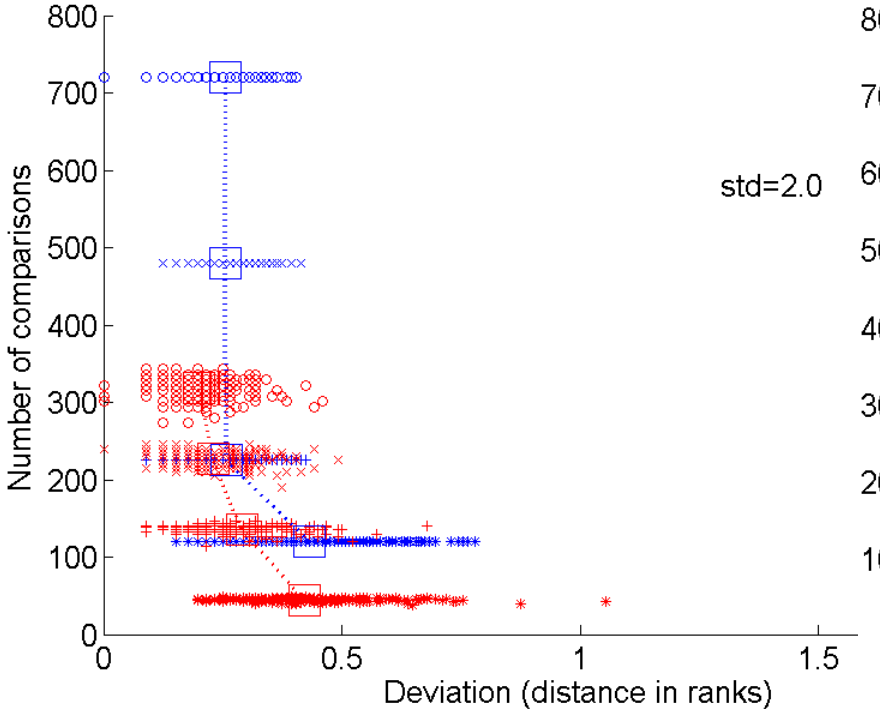
Pareto-Optimality of Sorting Algorithms

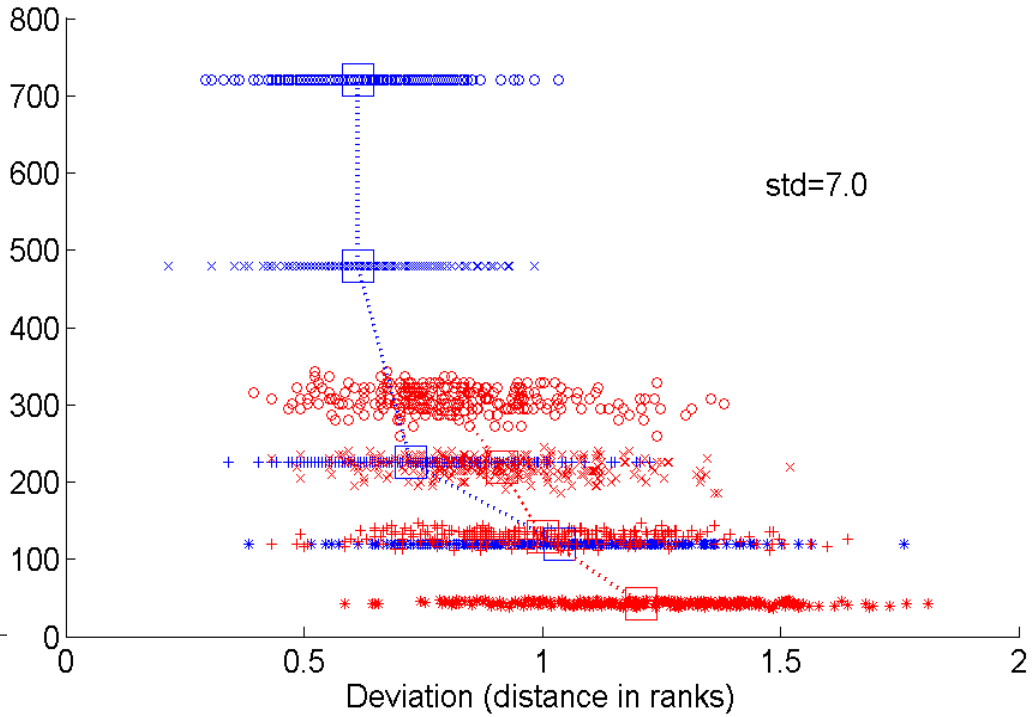
MergeSort

- o 7x majority vote
- x 5x
- + 3x
- * 1x

BubbleSort

- o 3n bubble steps
- x 2n
- + n
- * n/2

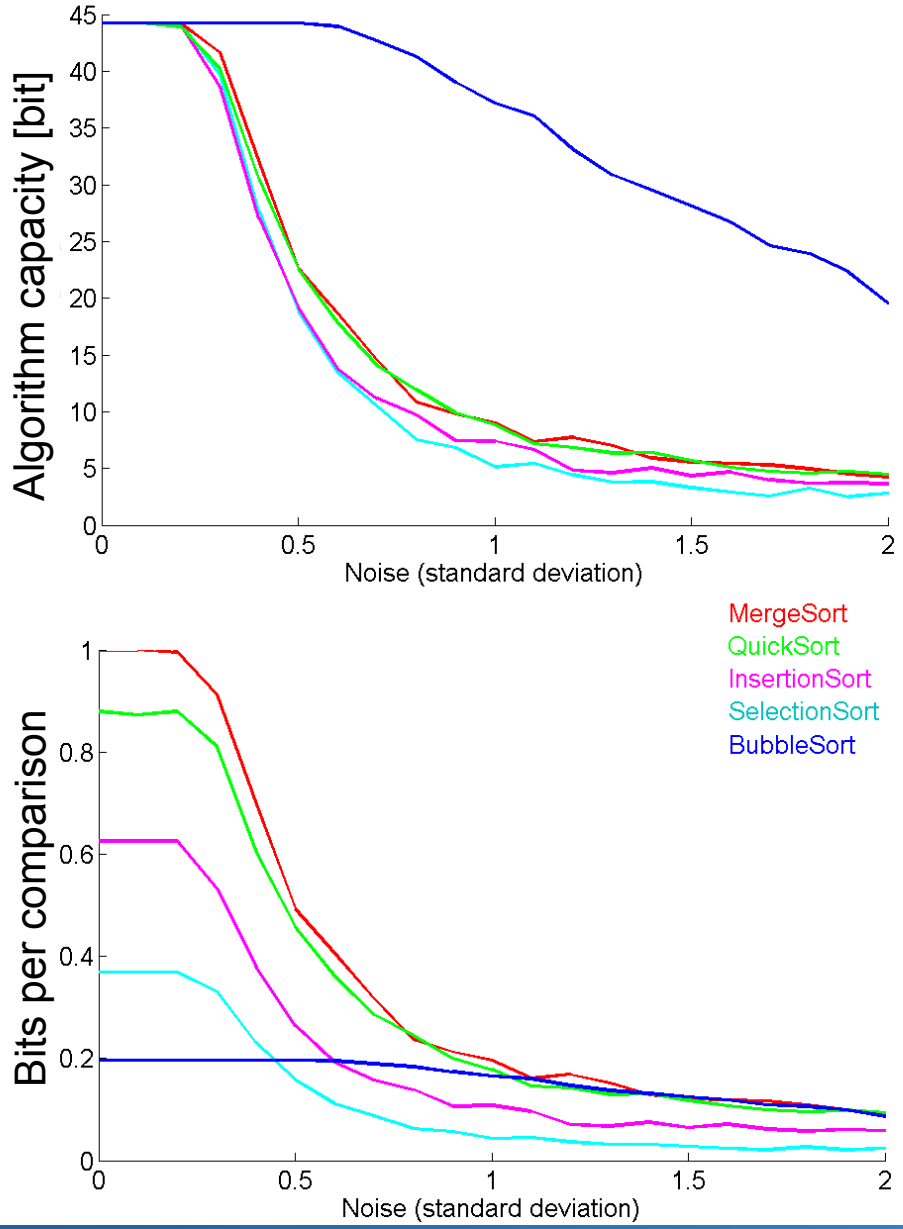




Information Capacity of Sorting Algorithms

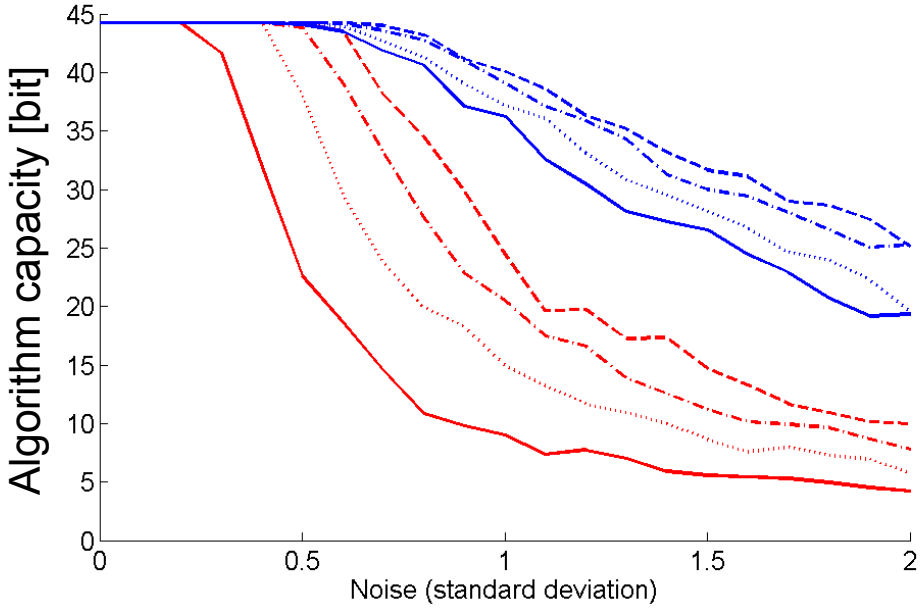
Preliminary study: n=16

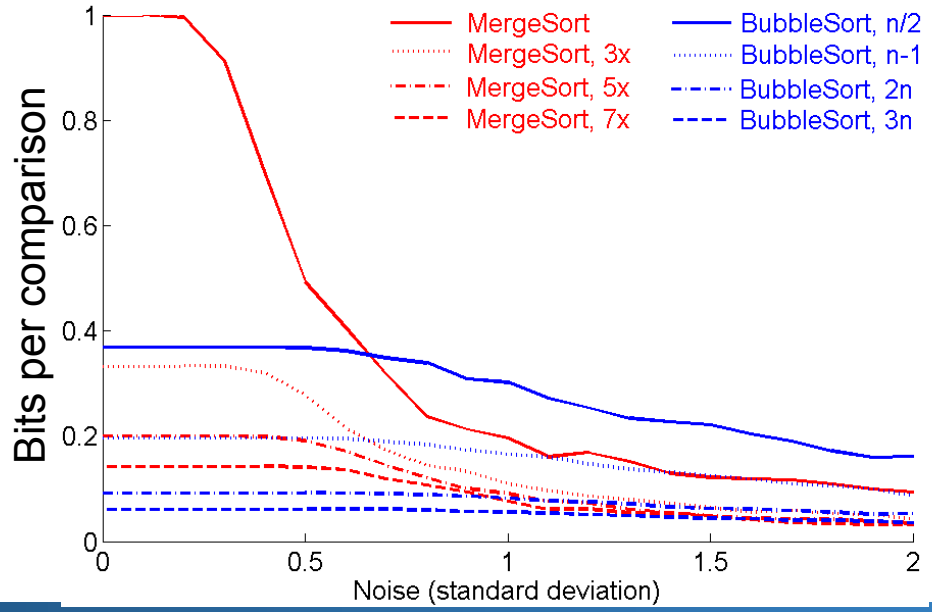
- Without runtime constraints **BubbleSort achieves maximal capacity**.
- **BubbleSort** is most robust at high noise levels, i.e., it extracts most information per comparison!



Information Content of BubbleSort ($O(n^2)$) to MergeSort ($O(n \log n)$)

- BubbleSort with resource budget outperforms MergeSort with majority voting.
- Algorithm design should be adapted to redundancy/ efficiency tradeoff!





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.

Future Work

- **Generalization**: replace approximation sets based on cost functions by smoothed outputs of **algorithms** ("smoothed generalization")
- **Model reduction** in dynamical systems: quantize sets of ODEs or PDEs (systems biology)
- Relate **statistical complexity**, i.e. the approximation capacity, to algorithmic or **computational complexity**.

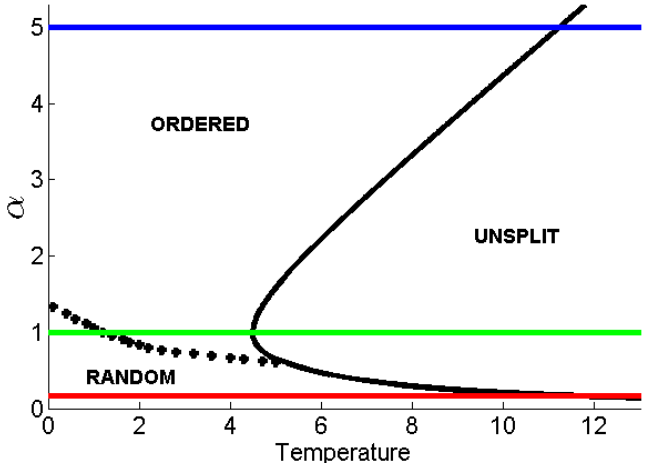
Credit to the ETH Machine Learning Group

Funding: SIMBAD (EU), FOR 916 (DFG & SNF)

Joint work with
M.H. Chehreghani
& M.Frank

High Dimensional Density Estimation

Barkai, Sompolinsky, Phys Rev E 50:1766, 1994.

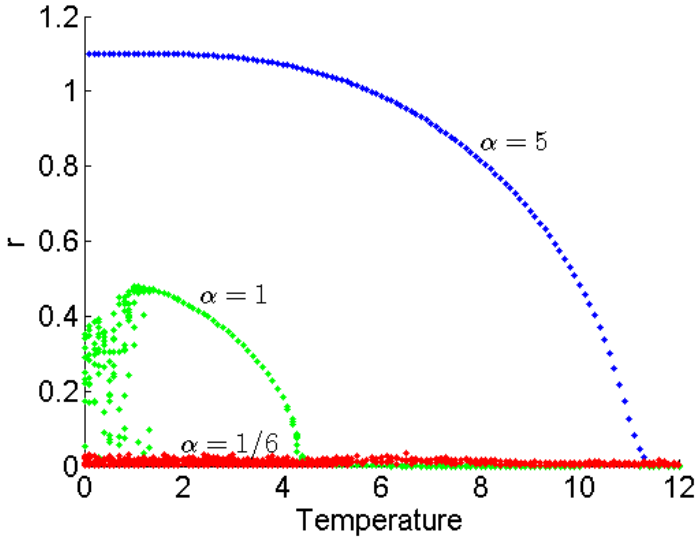


Phase Diagram:

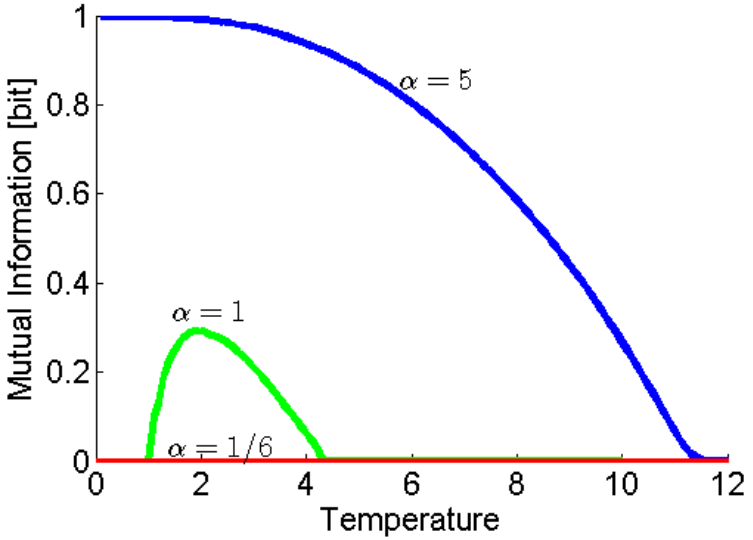
mixture of 2 Gaussians

n=500; d=100, 500, 3000; α := n/d

Overlap: $r = u_0^{-1}\langle \Delta\mu, \Delta\mu_0 \rangle$

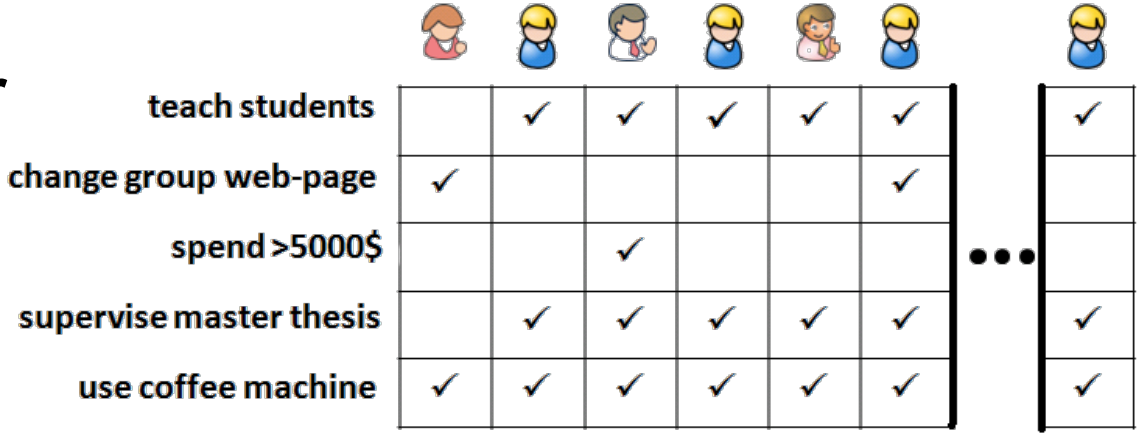


Approximation Capacity

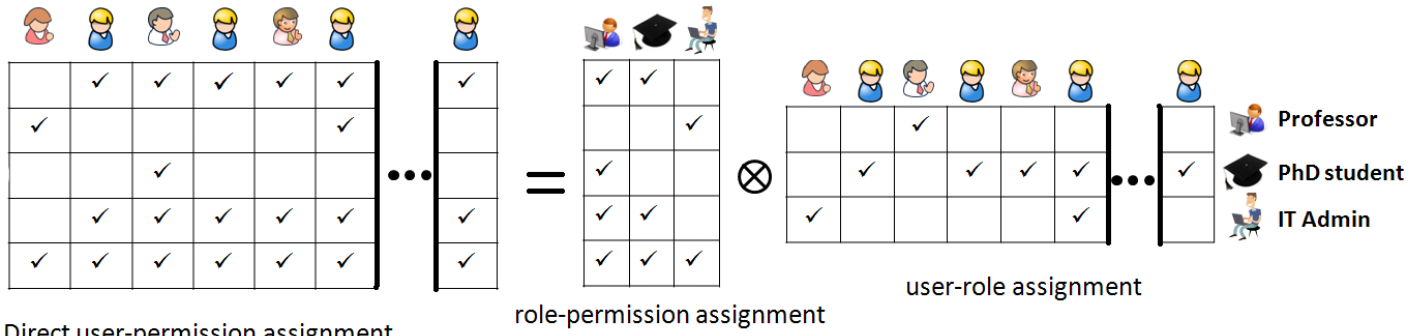


Role-Based Access Control

- **Given: Binary user permission matrix**



- **Discretional Access-Control**: Direct Assignments of users to permissions
- **Role-Based Access Control (RBAC)**: Permissions are granted via roles



Direct user-permission assignment

role-permission assignment

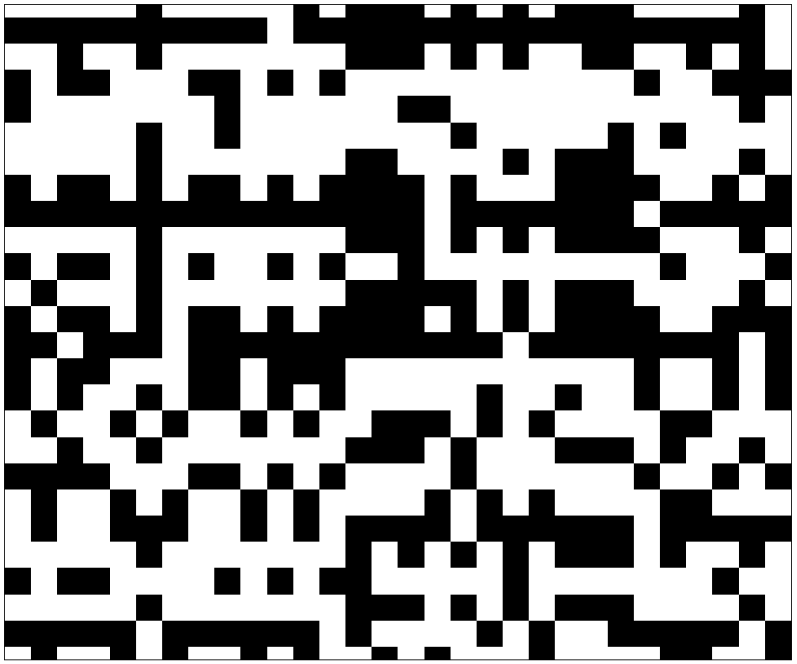
user-role assignment

Role-Mining for RBAC

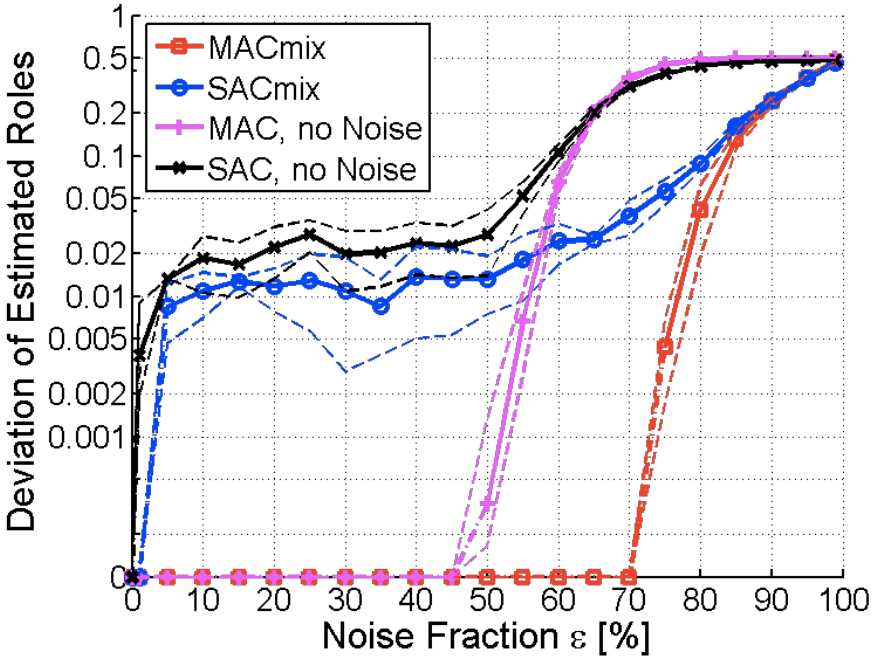
- **Role-Mining**: Given a user-permission assignment matrix **X**, find a set of roles **U** and assignments **Z** such that

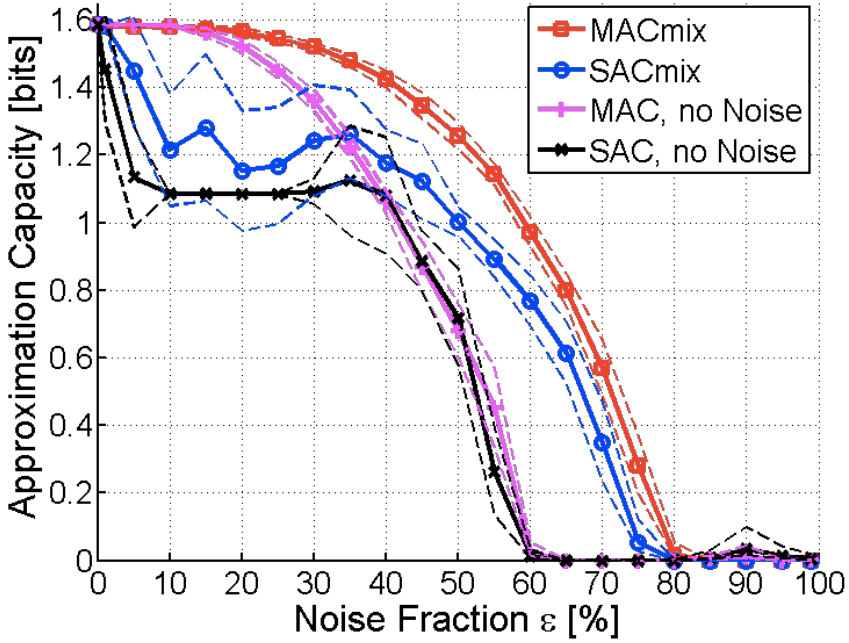
  $$\mathbf{X} \approx \mathbf{U} \otimes \mathbf{Z}$$

- **Multi Assignment Clustering**: generative approach including noise model, inference with DA



Synthetic Data: Parameter Accuracy vs. Approximation Capacity

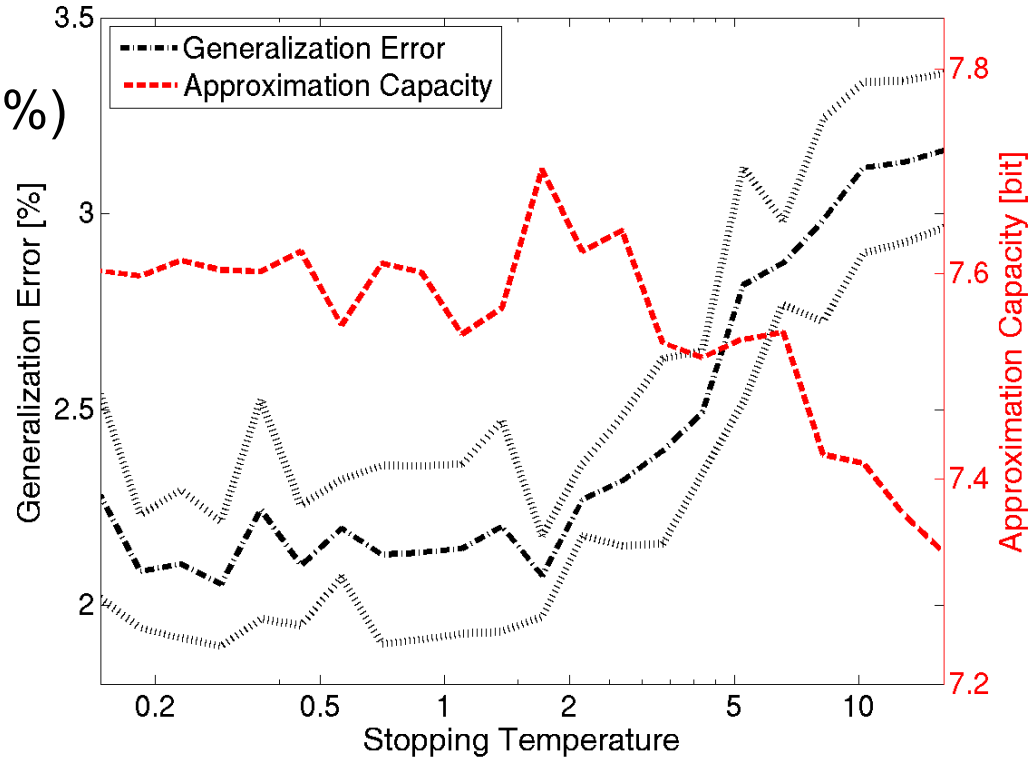




MAC: More accurate estimators for centroids, it yields higher approximation capacity than SAC.

Real-World Data: Prediction Error vs. Approximation Capacity

- **Generalization**: Can roles predict permissions of **new** users?
  1. Use few permissions (20%) to determine role set
  2. Predict hidden/missing permissions (80%).
- Centroids with maximal capacity yield minimal generalization error



Optimization approach to machine learning

- Given: **data** $\mathbf{X} \in \mathcal{X}$ **in data (input) space** $\mathcal{X}$
- **Goal**: **Learn structure from data**, i.e., interpret data relative to a hypothesis class
- **Hypothesis class** $\mathcal{C}$ with hypotheses (solutions)

$$c \;:\; \mathcal{X} \;\rightarrow\; \mathbb{K} \qquad (\text{e.g., } \mathbb{B}^n \text{ or } \{1,\ldots,k\}^n)$$

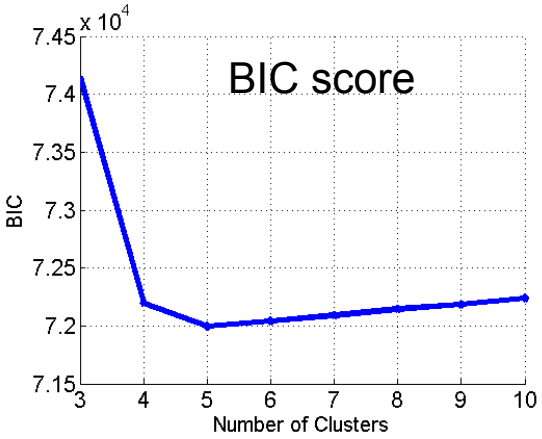
$$\mathbf{X} \;\mapsto\; c(\mathbf{X})$$

- **Cost function** to define a partial order on $\mathcal{C}$

$$R \;:\; \mathcal{C} \times \mathcal{X} \;\rightarrow\; \mathbb{R}_{\geq 0}$$

$$(c, \mathbf{X}) \;\mapsto\; R(c, \mathbf{X})$$

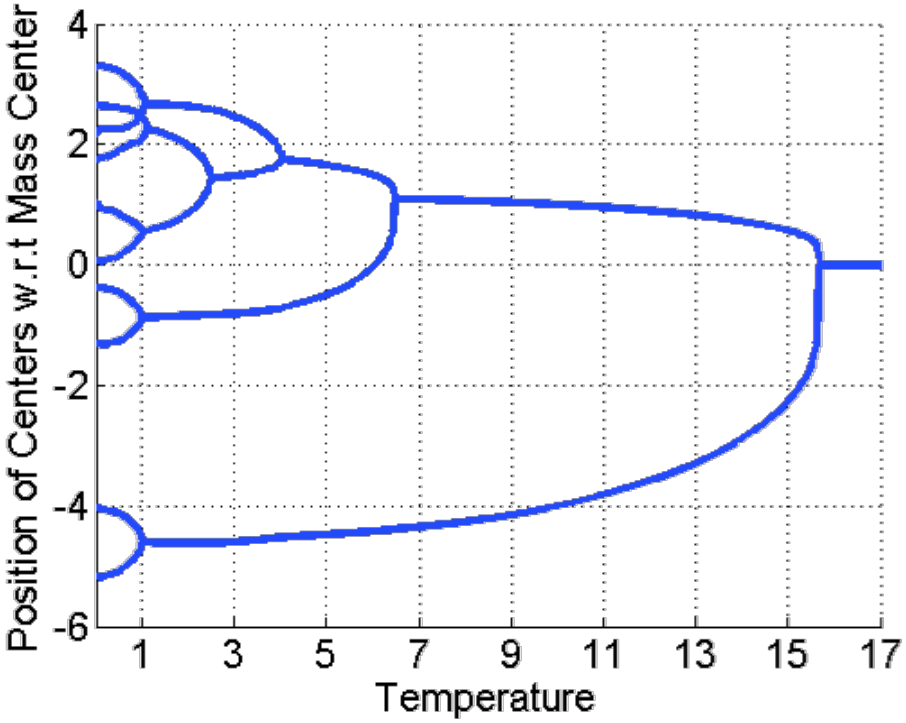
2d Mixture Model Estimation



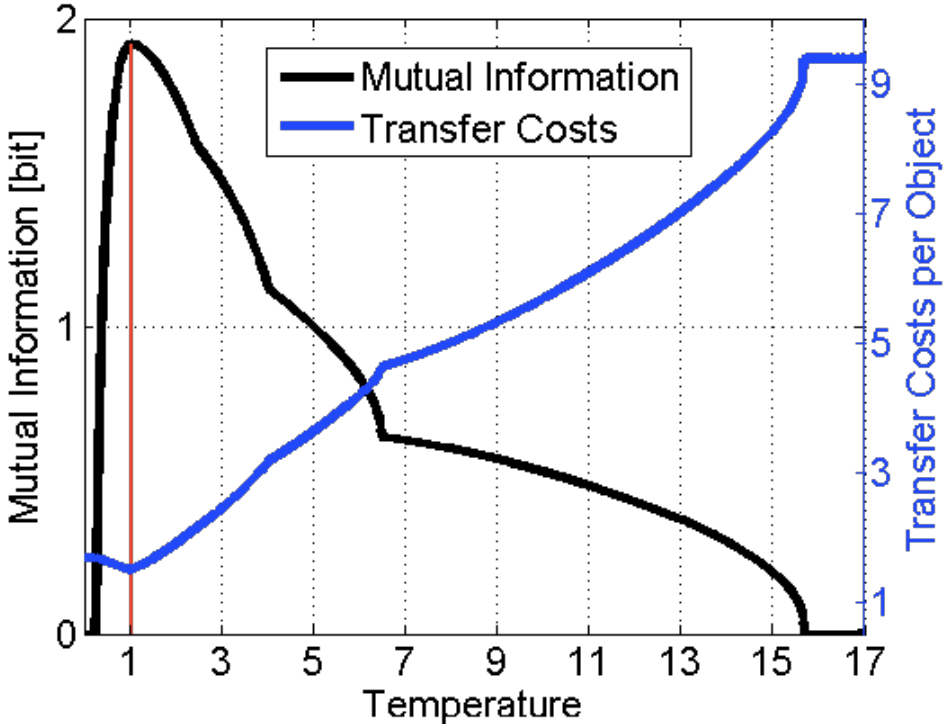
Experimental Setting:

5 Gaussians, n=10000, d=2, k^{max}=10

Cluster splitting

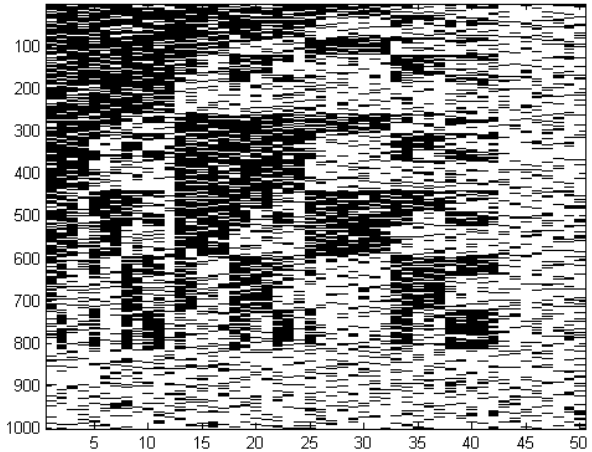


Approximation Capacity



Denoising Binary Matrices by rank-k approximation

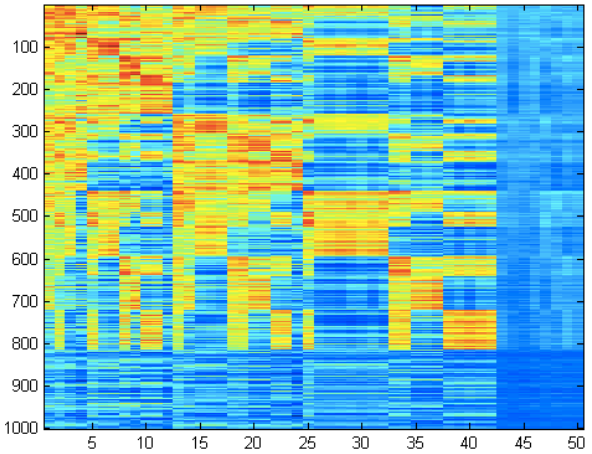
Boolean matrix with 40% random entries



$X = USV$

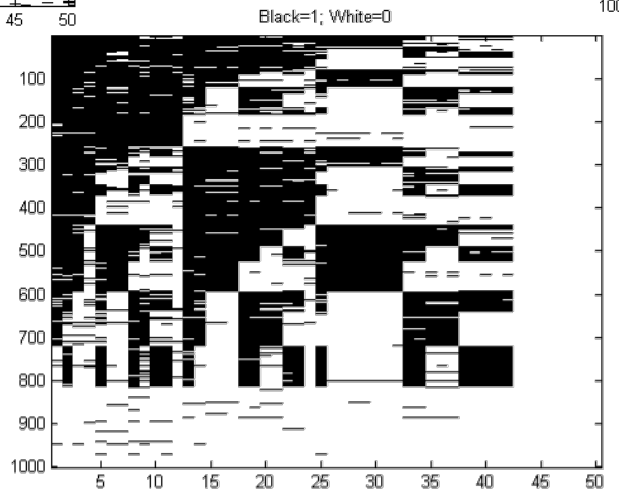
continuous rank-k approximation

$X_5 = U_5 S_5 V_5$



Rounding as approximation

$g(X_k) = \text{round}(X_k)$

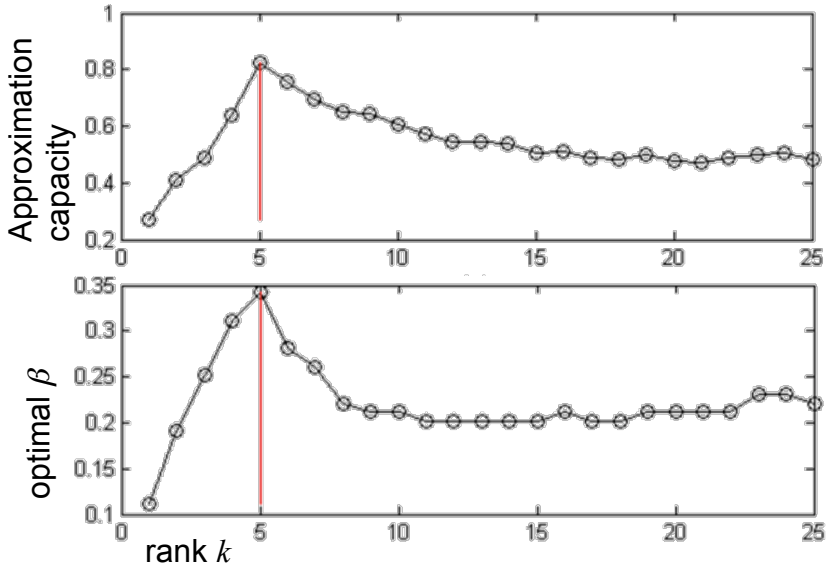


Maximum of approximation capacity selects optimal rank *k*

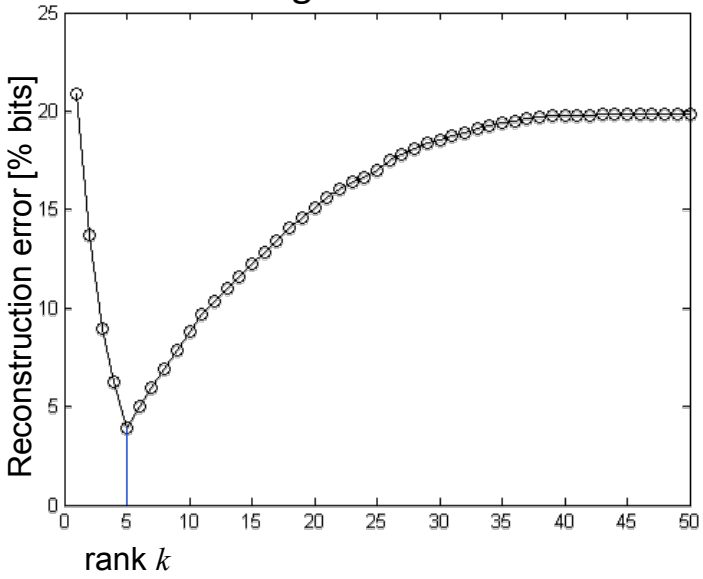
- Integrate over variations of the signal matrix U.

$$\hat{\mathcal{I}}_\beta(\tau_s, \hat{\tau}) \quad = \quad \frac{1}{n} \log \frac{|\{\tau_s\}| Z_\beta^{1+2}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})}{Z_\beta(\mathbf{X}^{(1)}) Z_\beta(\mathbf{X}^{(2)})}$$

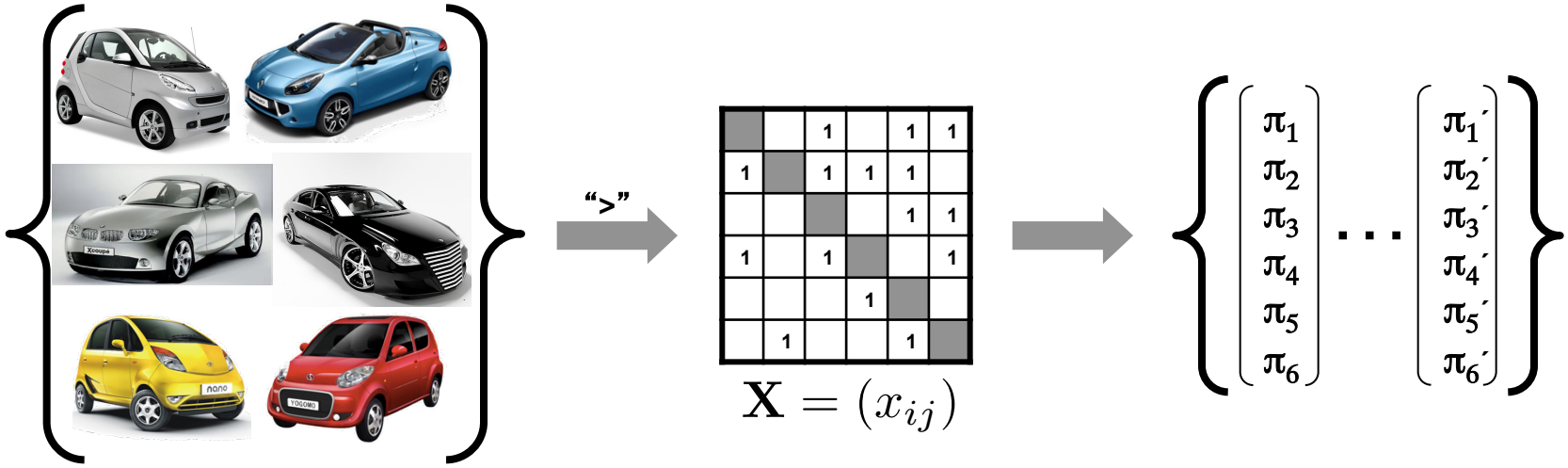
Approximation capacity vs. k



Difference to ground-truth vs. k



Approximate Sorting

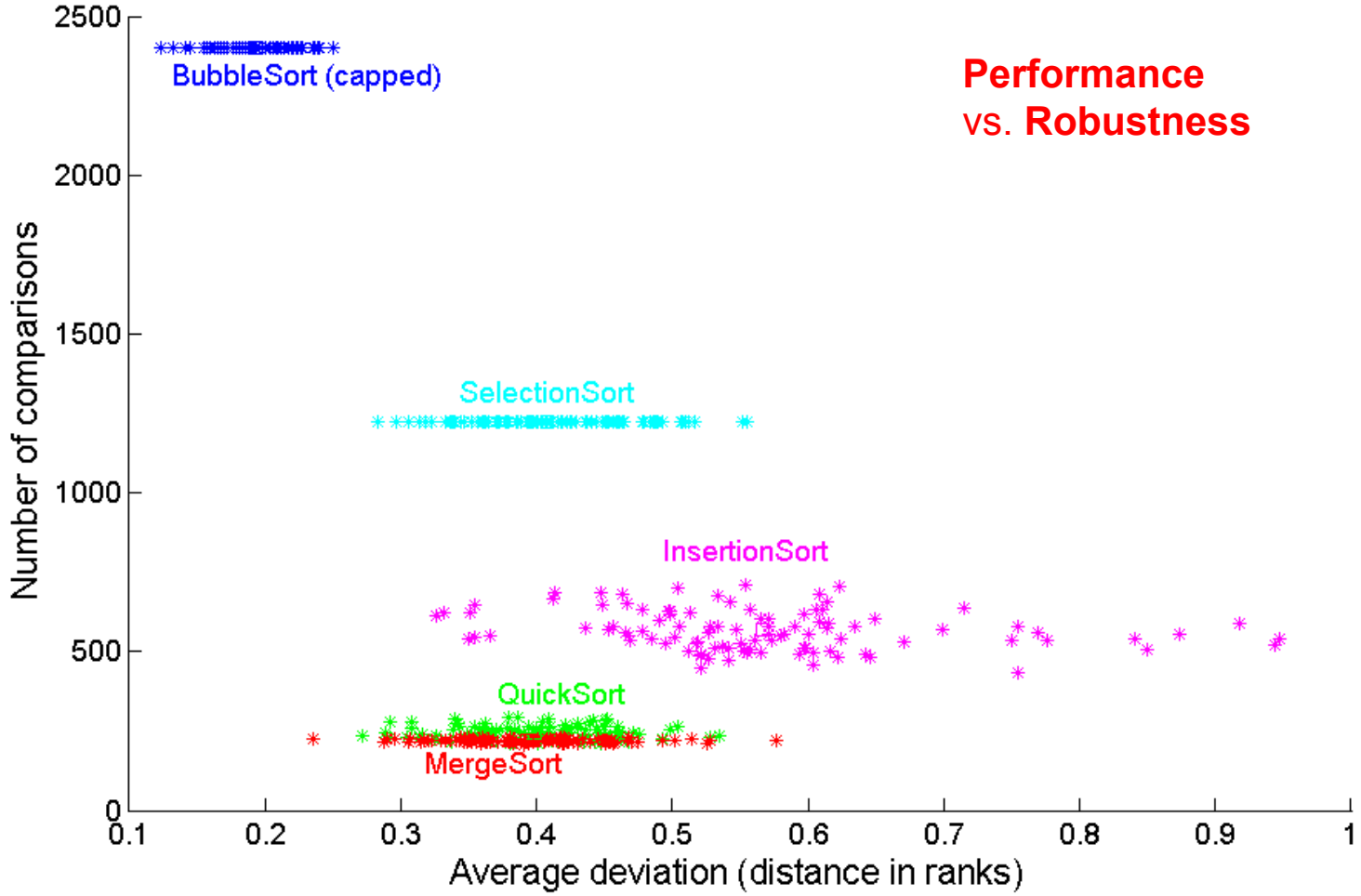


- Among all permutations π, find the one that **orders the items best**!

$$\mathcal{R}^{\text{Sorting}}(\pi|\mathbf{X}) = \sum_{i,j} x_{ij} \mathbb{I}_{\{\pi_i > \pi_j\}}$$

*(This cost function counts the number of disagreements between the **provided pairwise comparisons** and the **pairwise comparisons induced by a proposal ranking** from the hypothesis class.)*

Pareto-Optimality of Sorting Algorithms



Performance
vs. **Robustness**

Experimental setting: array size 50, noise 10%