Constraint based learning of mixtures of trees

François Schnitzler

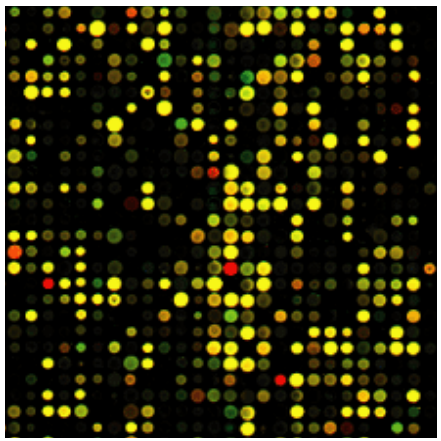
Méthodes stochastiques
University of Liège

MODGRAPH - 6th June 2009

The goal of this research is to improve the learning of high-dimensional probability densities

This has great potential in bioinformatics :

- Classification
- Determining reaction to a combination of factors
- Experiments planning



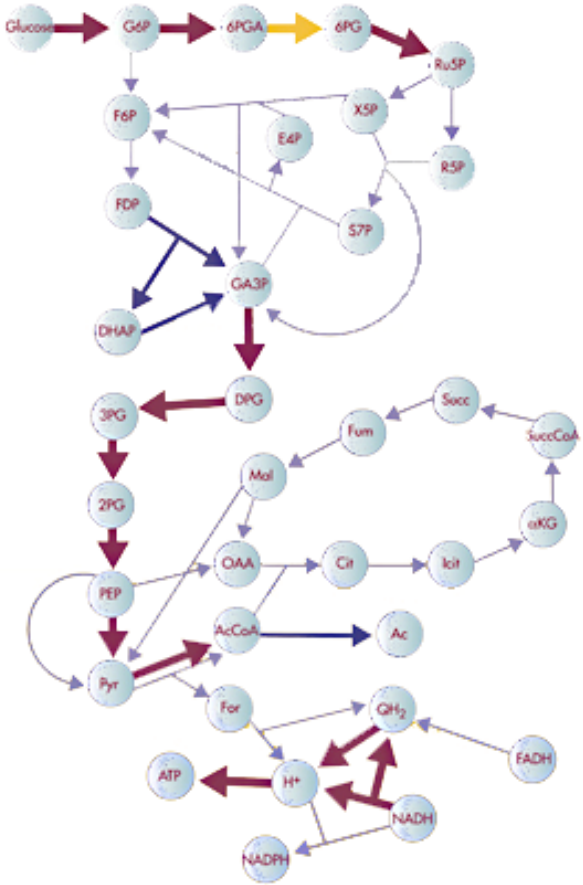
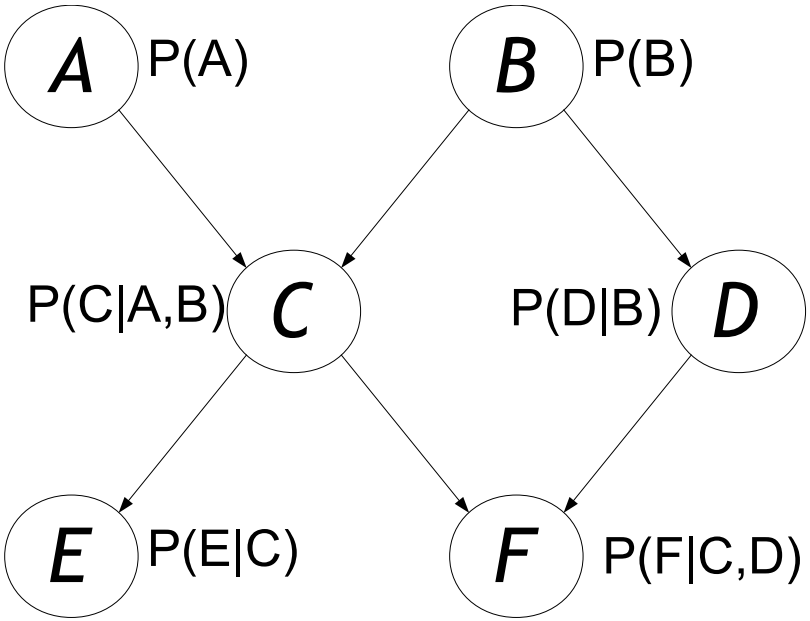


Table of contents

Bayesian networks model probability densities



$$P(A, B, ..., F) = P(A)P(B)P(C|A, B)...P(F|C, D)$$

- Directed graph without directed cycles
- A set of local functions
- Each node $\equiv$ one random variable
- Each local function $\equiv$ cond. prob. table

$\Rightarrow$ Factorization of the probability density

$\Rightarrow$ Set of conditional independences

Several methods exist for building bayesian networks

2 main techniques

- Expert specification: exploiting knowledge
- Machine learning: exploiting data

Decomposition of the learning problem

- Learning graph structure
  - Analyzing relations between variables
  - Optimizing a score on a set of candidate structure
- Local functions parameters estimation

The choice of the structure search space is a compromise

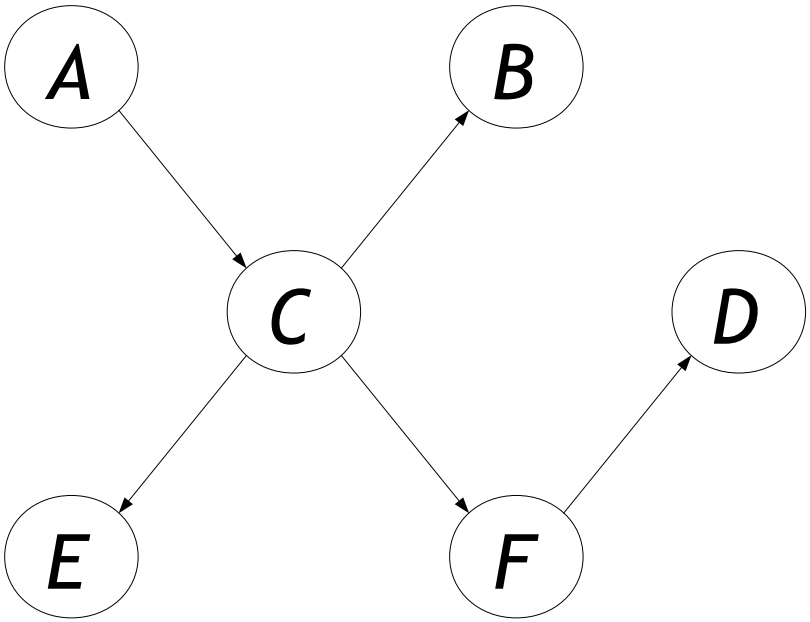
Sets of all bayesian networks

- Ability to model any density
- Superexponential number of structures
  $\Rightarrow$ Structure learning is difficult
  $\Rightarrow$ Overfitting
- Inference is difficult

Sets of simpler structures

- Reduced modeling power
- Learning and inference potentially easier

Trees are a subclass of bayesian networks



- Each node as at most one parent

$\Rightarrow$ no cycle

- Fully connected (in this talk)
- Edge directions irrelevant

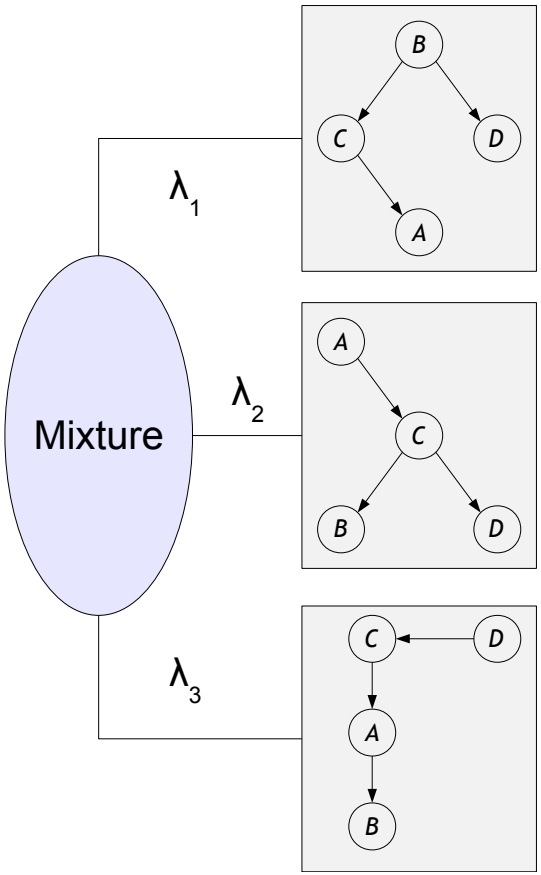
Interesting properties

- Inference is linear in the number of variables (n)
- Optimal learning (with respect to data log-likelihood) is ($\mathcal{O}(n^2 \log n)$)

Drawback

- Weak representational power

Mixture models improve modeling capacity



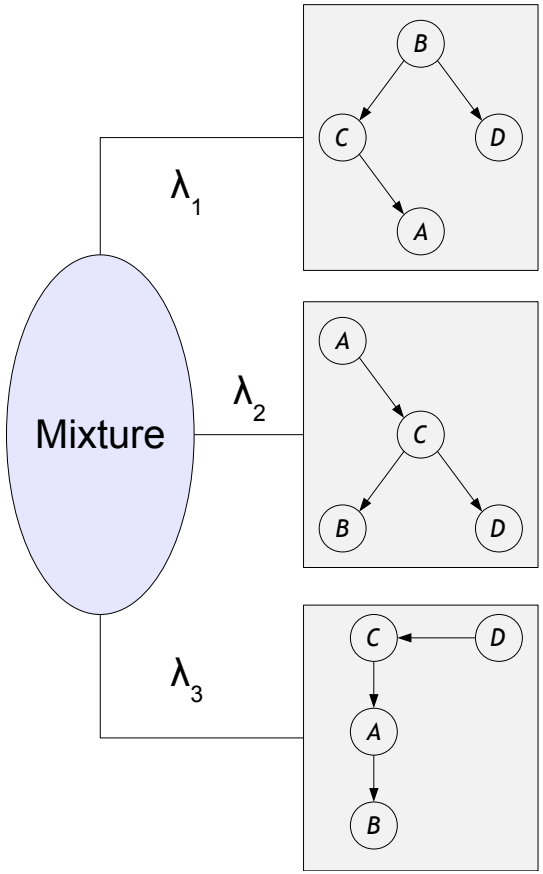
$$P(\mathbf{X}) = \sum \lambda_i \, P_i(\mathbf{X})$$

- Simple models : low computing complexity
- Several models : Larger modeling power
- Complexity control by adjusting model size

Mixtures of m trees can be built in the optimization framework :

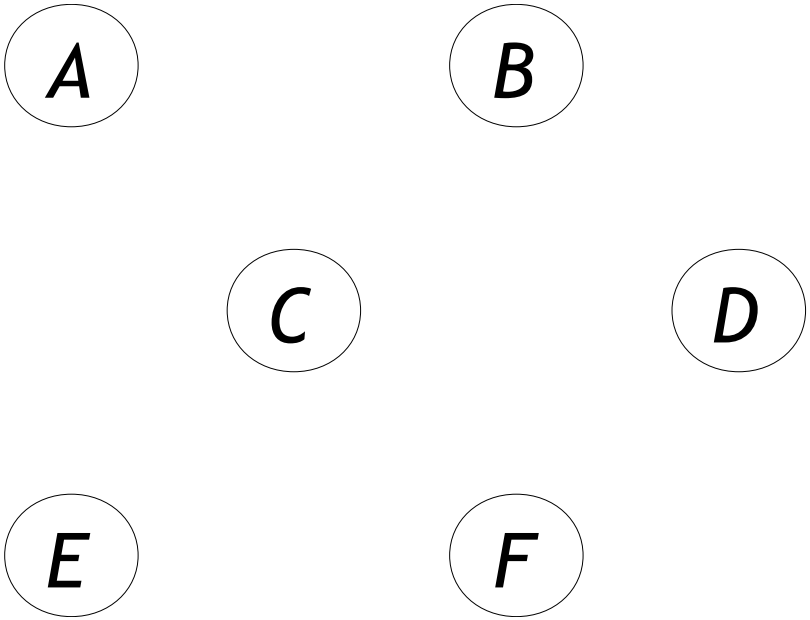
Based on the EM algorithm and a fixed number of trees

- Optimization of all trees and parameters simultaneously
- Trying to fit trees to partitions of the data



Learning with mixtures of Trees, M. Meila & M.I. Jordan, JMLR 2001.

Optimal construction of a tree

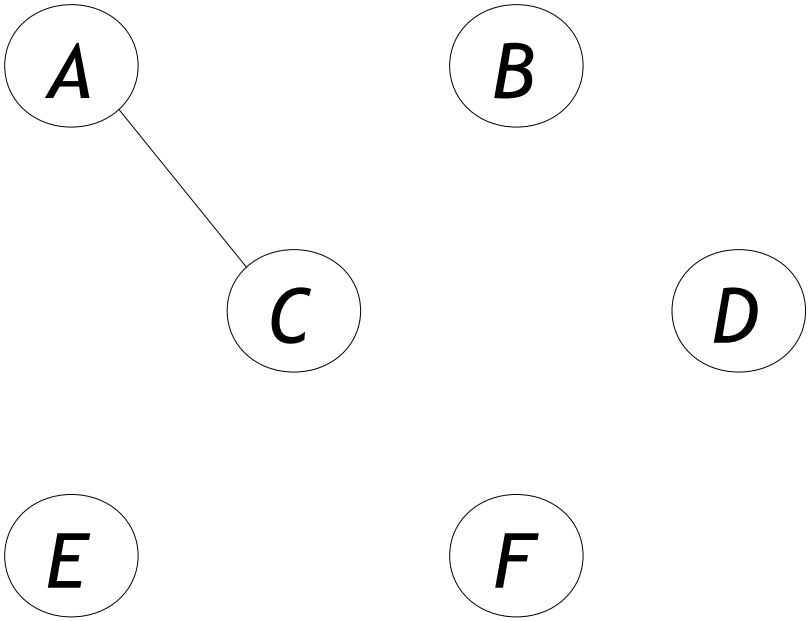


	A	B	C	D	E	F
A		21	25	14	3	15
B			23	17	5	8
C				11	13	18
D					7	19
E						4
F						

Algorithm

1. Compute mutual information between pairwise variables
2. Build a maximum width spanning tree
3. Determine parameters

Optimal construction of a tree

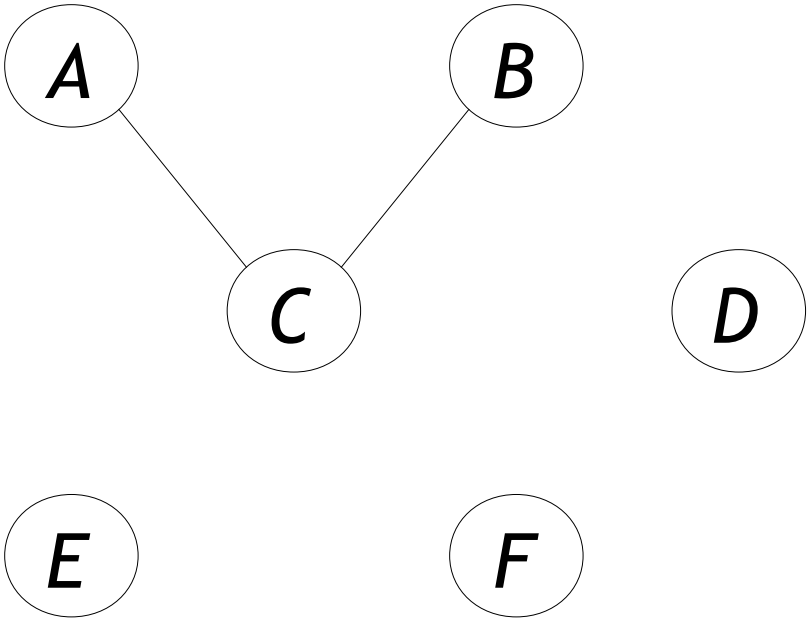


	A	B	C	D	E	F
A		21	25	14	3	15
B			23	17	5	8
C				11	13	18
D					7	19
E						4
F						

Algorithm

1. Compute mutual information between pairwise variables
2. Build a maximum width spanning tree
3. Determine parameters

Optimal construction of a tree

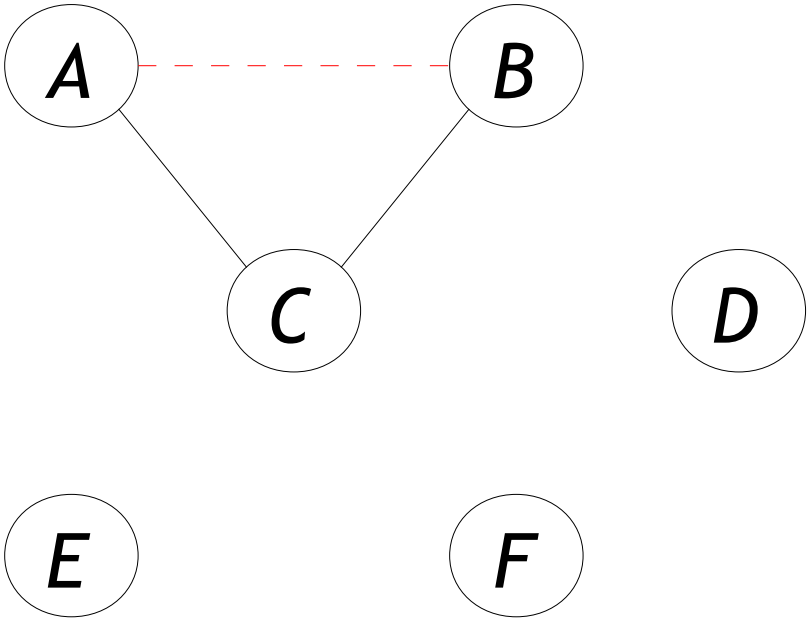


	A	B	C	D	E	F
A		21	25	14	3	15
B			23	17	5	8
C				11	13	18
D					7	19
E						4
F						

Algorithm

1. Compute mutual information between pairwise variables
2. Build a maximum width spanning tree
3. Determine parameters

Optimal construction of a tree

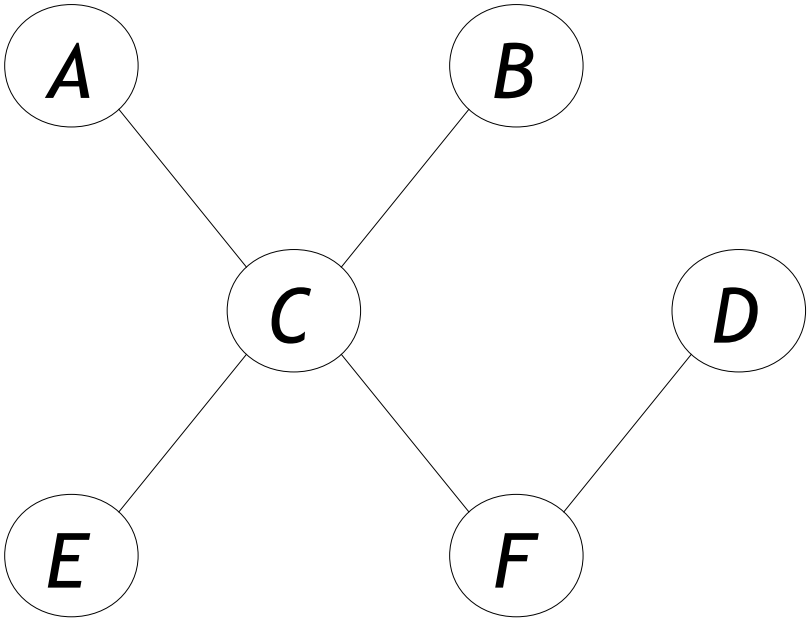


	A	B	C	D	E	F
A		21	25	14	3	15
B			23	17	5	8
C				11	13	18
D					7	19
E						4
F						

Algorithm

1. Compute mutual information between pairwise variables
2. Build a maximum width spanning tree
3. Determine parameters

Optimal construction of a tree

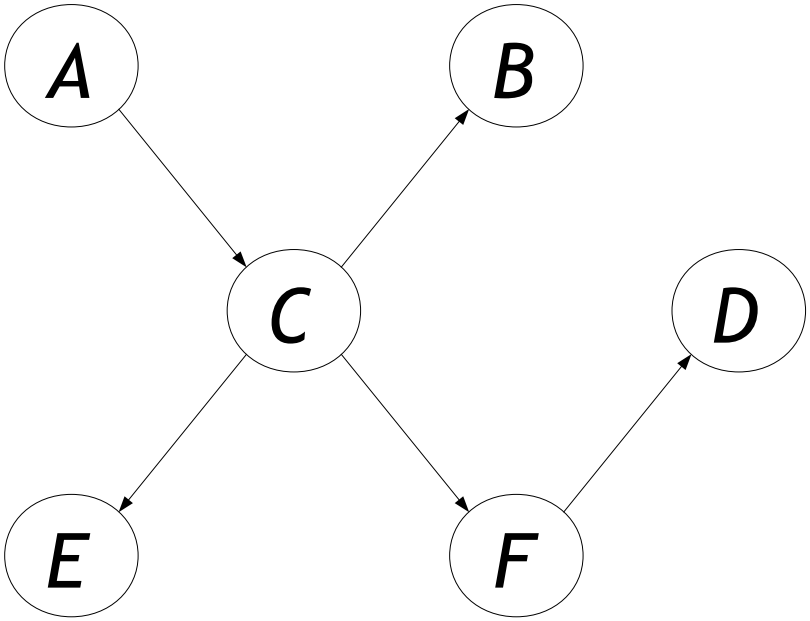


	A	B	C	D	E	F
A		21	25	14	3	15
B			23	17	5	8
C				11	13	18
D					7	19
E						4
F						

Algorithm

1. Compute mutual information between pairwise variables
2. Build a maximum width spanning tree
3. Determine parameters

Optimal construction of a tree



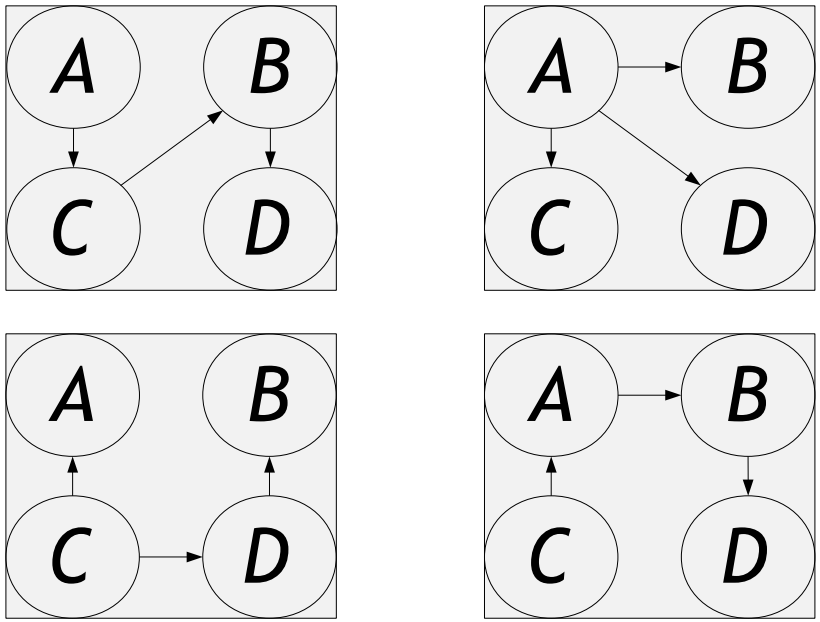
	A	B	C	D	E	F
A		21	25	14	3	15
B			23	17	5	8
C				11	13	18
D					7	19
E						4
F						

Algorithm

1. Compute mutual information between pairwise variables
2. Build a maximum width spanning tree
3. Determine parameters

A mixture of trees built using EM algorithm can provide information about the relation between variables.

Relations between pair-wise variables ← number of models where those two variables are directly connected.



→

	A	B	C	D
A		2	4	1
B			1	3
C				1
D				

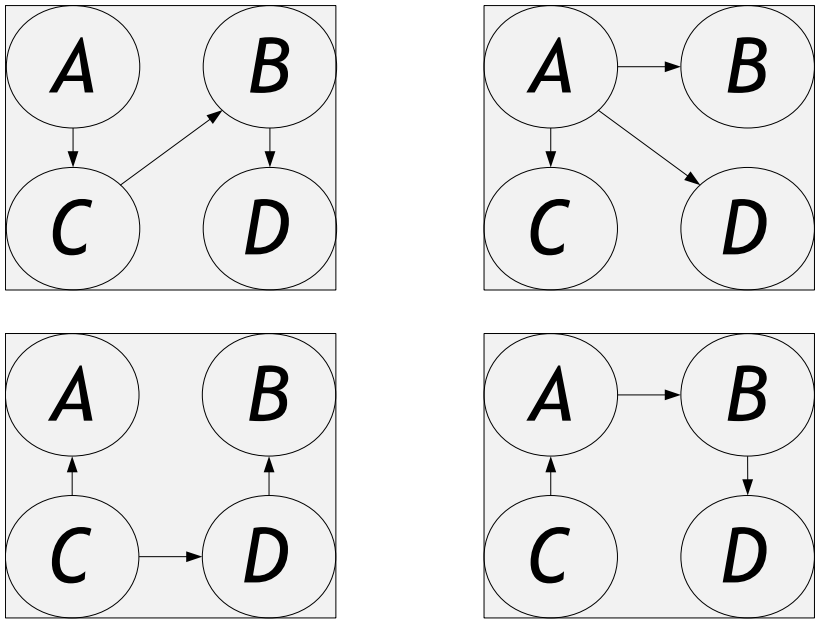
Question : Is it possible to build a mixture based on the structure of the problem?

1. Rationale
   - Bayesian networks
   - Mixture models
2. First approach
3. Results

Idea : Trying to represent the mutual information

	A	B	C	D
A		4.1	7.8	1.9
B			1.6	5.7
C				2.2
D				

$\xrightarrow{?}$



Building a mixture model that can be interpreted to recover the mutual information matrix

1. determining number of edges
2. building trees
3. Computing parameters

The ideal edge numbers are computed from the mutual information matrix

Associating relationship and edge

- Mutual information was used as a measure of relationship
- $N_{i,j} \; \alpha \; I_{i,j}$
- A tree of n variables has exactly $n-1$ edges
- $\sum_{i=1}^{n} \sum_{j=i+1}^{n} N_{i,j} = m(n-1) = \alpha \sum_{i=1}^{n} \sum_{j=i+1}^{n} I_{i,j}$

$$\Rightarrow N_{i,j} = m * (n-1) \frac{I_{i,j}}{\sum_{i=1}^{n} \sum_{j=i+1}^{n} I_{i,j}}$$

	A	B	C	D
A		4.1	7.8	1.9
B			1.6	5.7
C				2.2
D				

$\rightarrow$

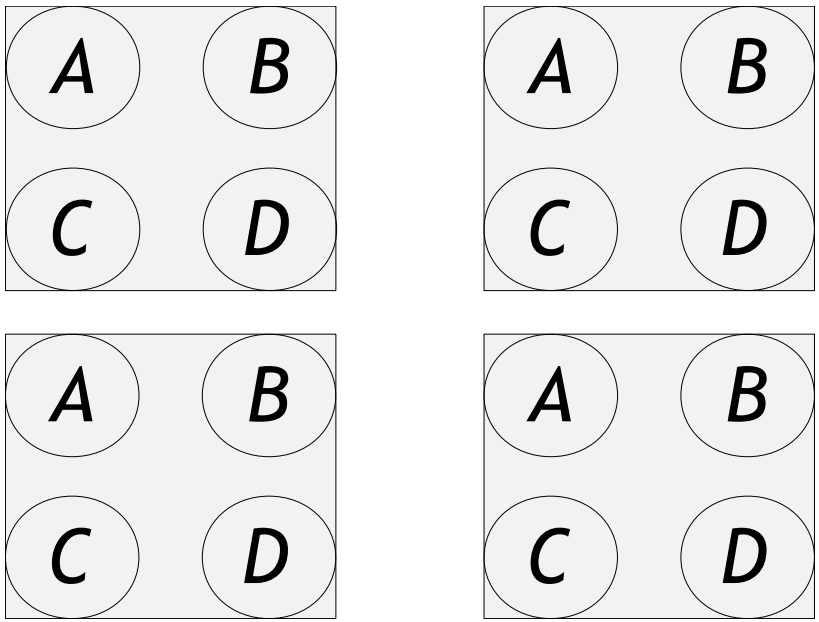
	A	B	C	D
A		2.1	4	1
B			0.8	2.9
C				1.1
D				

The edges must be distributed among the terms of the model to form tree-structures

non optimal method tested : loop on the different terms:

- MWST based on the remaining number of edges
- remove the edges used

	A	B	C	D
A		2.1	4	1
B			0.8	2.9
C				1.1
D				

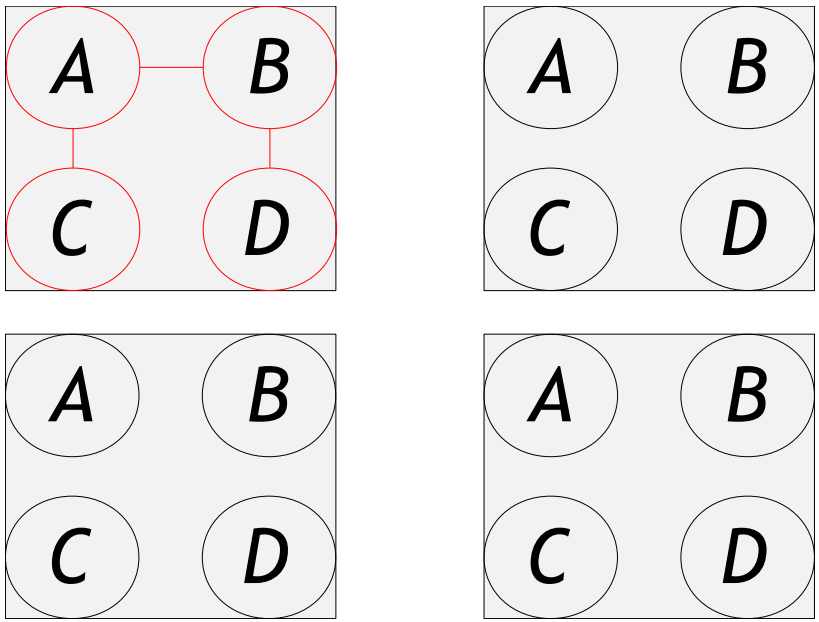


The edges must be distributed among the terms of the model to form tree-structures

non optimal method tested : loop on the different terms:

- MWST based on the remaining number of edges
- remove the edges used

	A	B	C	D
A		2.1	4	1
B			0.8	2.9
C				1.1
D				

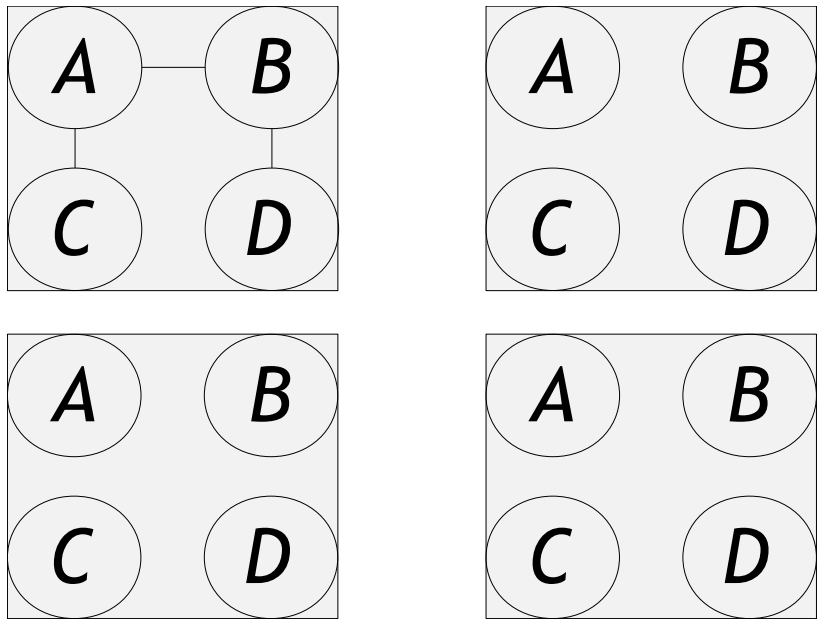


The edges must be distributed among the terms of the model to form tree-structures

non optimal method tested : loop on the different terms:

- MWST based on the remaining number of edges
- remove the edges used

	A	B	C	D
A		1.1	3	1
B			0.8	1.9
C				1.1
D				

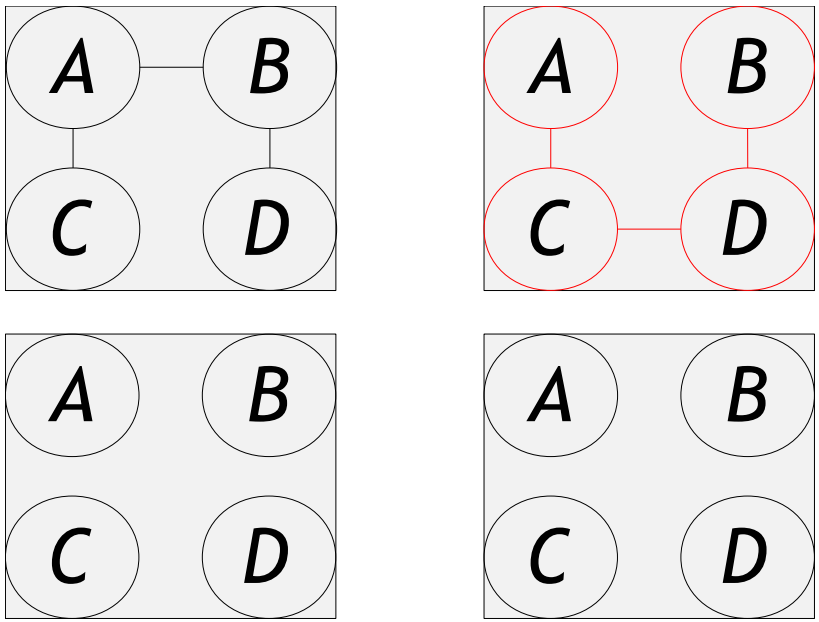


The edges must be distributed among the terms of the model to form tree-structures

non optimal method tested : loop on the different terms:

- MWST based on the remaining number of edges
- remove the edges used

	A	B	C	D
A		1.1	3	1
B			0.8	1.9
C				1.1
D				

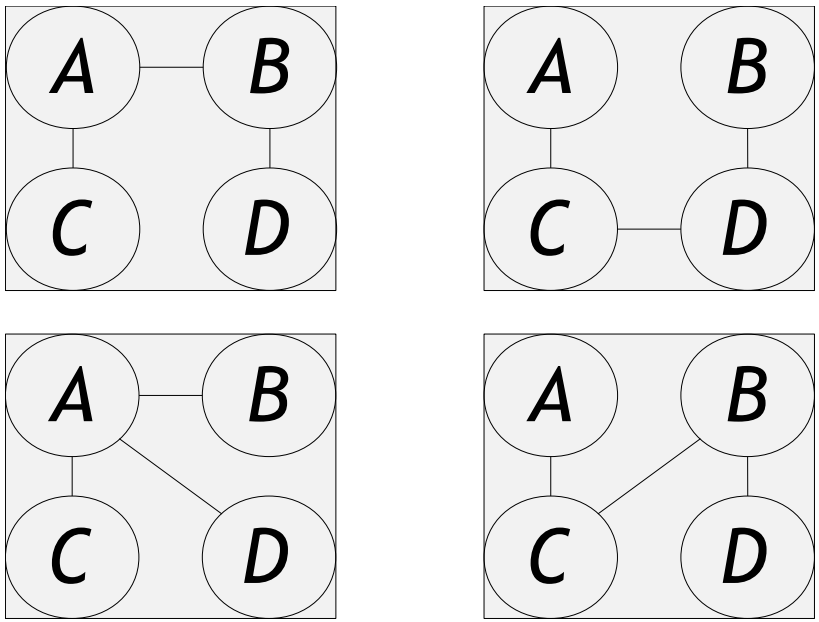


The edges must be distributed among the terms of the model to form tree-structures

non optimal method tested : loop on the different terms:

- MWST based on the remaining number of edges
- remove the edges used

	A	B	C	D
A		0.1	0	-0
B			-0.2	-0.1
C				0.1
D				



Parameters determination

Tree parameterization

- Each tree is parametrized based on the whole learning set
- Parameters are chosen according to the maximum likelihood criteria

Model parameterization

- Uniform weights for all trees

Results

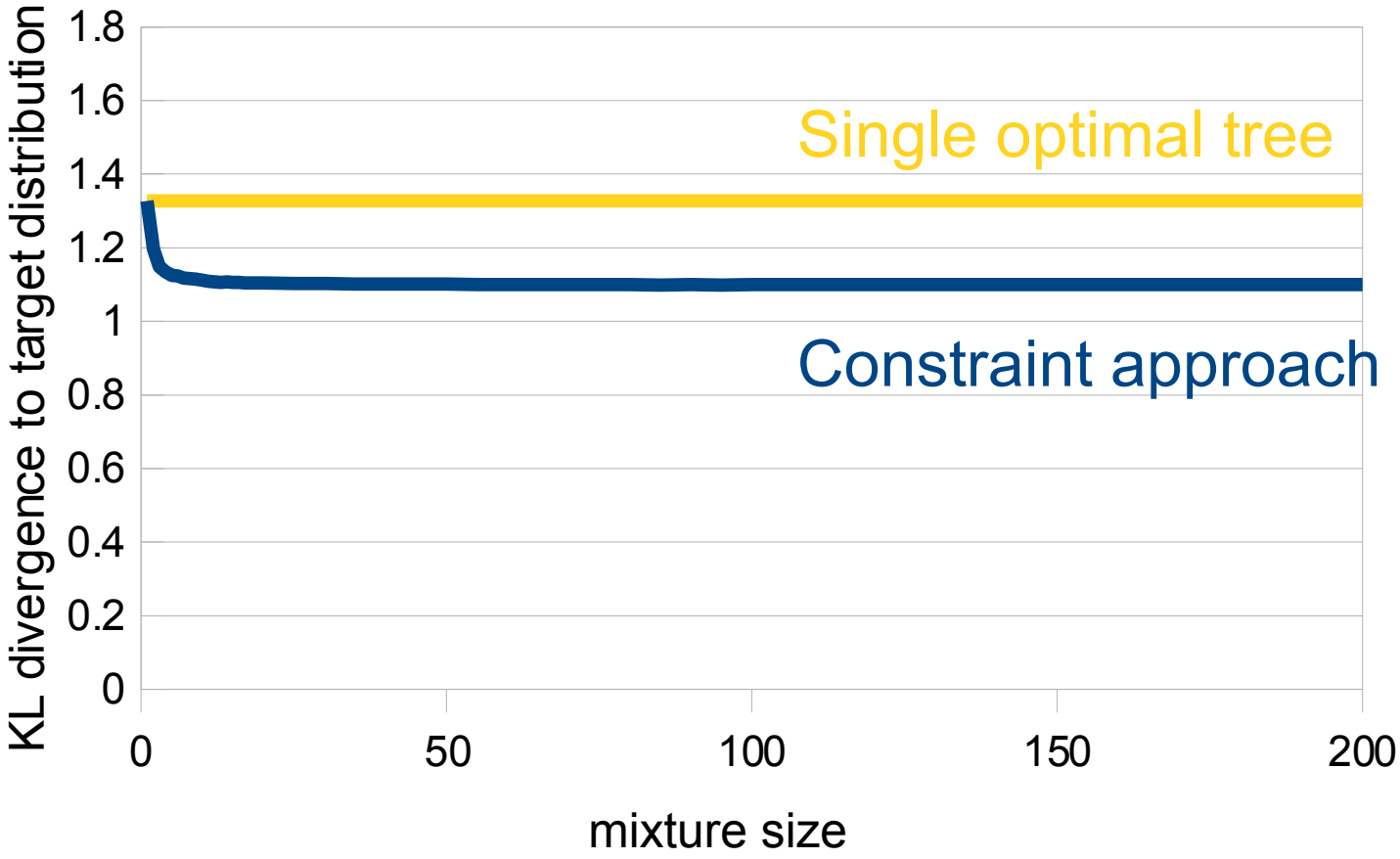
The approach was evaluated on artificial models

- Generation of 10 random bayesian networks of 16 variables
- Generation of several data sets for each model
- Apply algorithm to each data set, compare to original distribution (Kullback-Leibler divergence)
- Average on all data set

Comparison with :

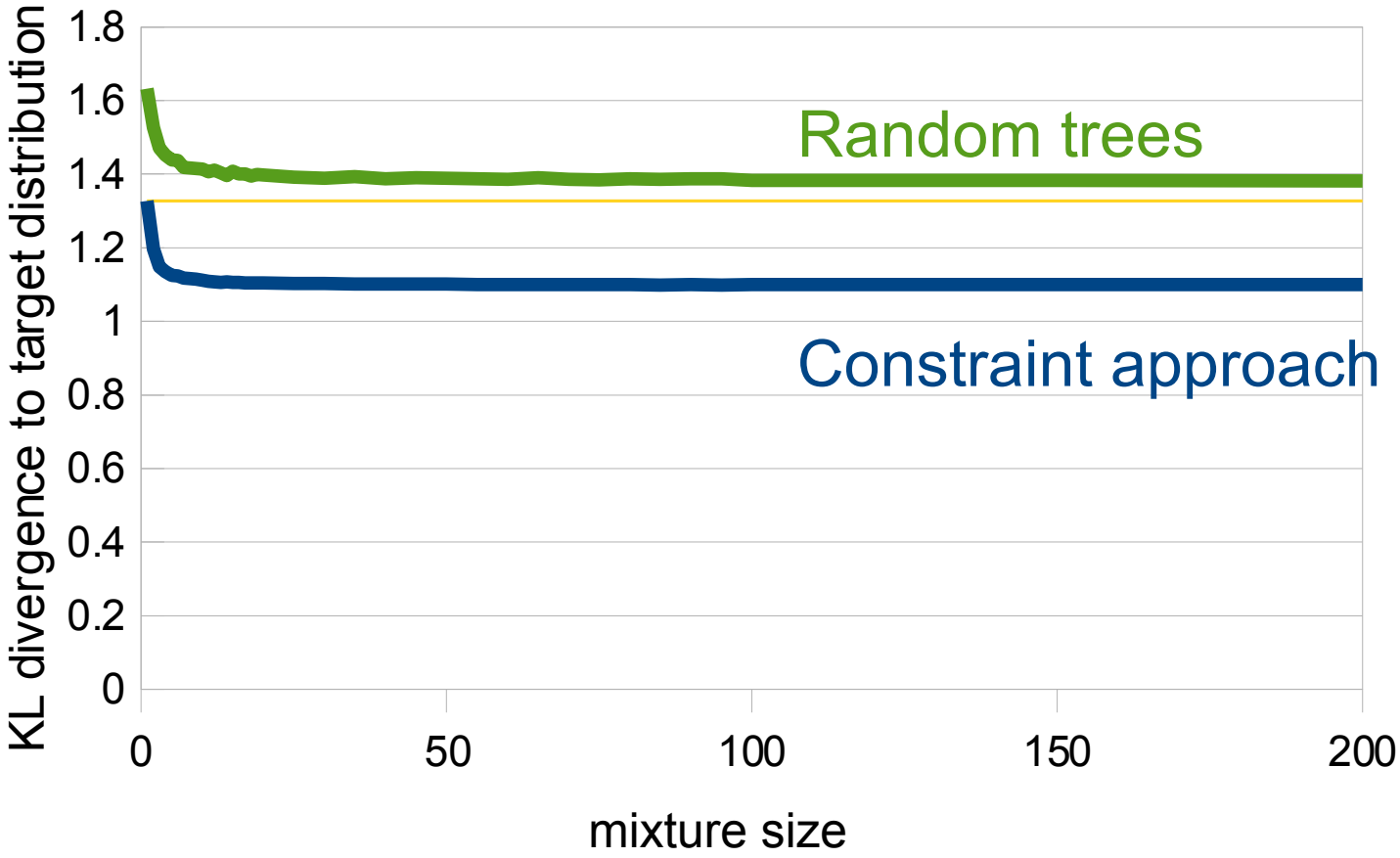
- Optimal tree
- Random tree-structures

Results on small sets (50 samples) are promising...



- Improvement of the mixtures over the single tree
- fast convergence, no overfitting

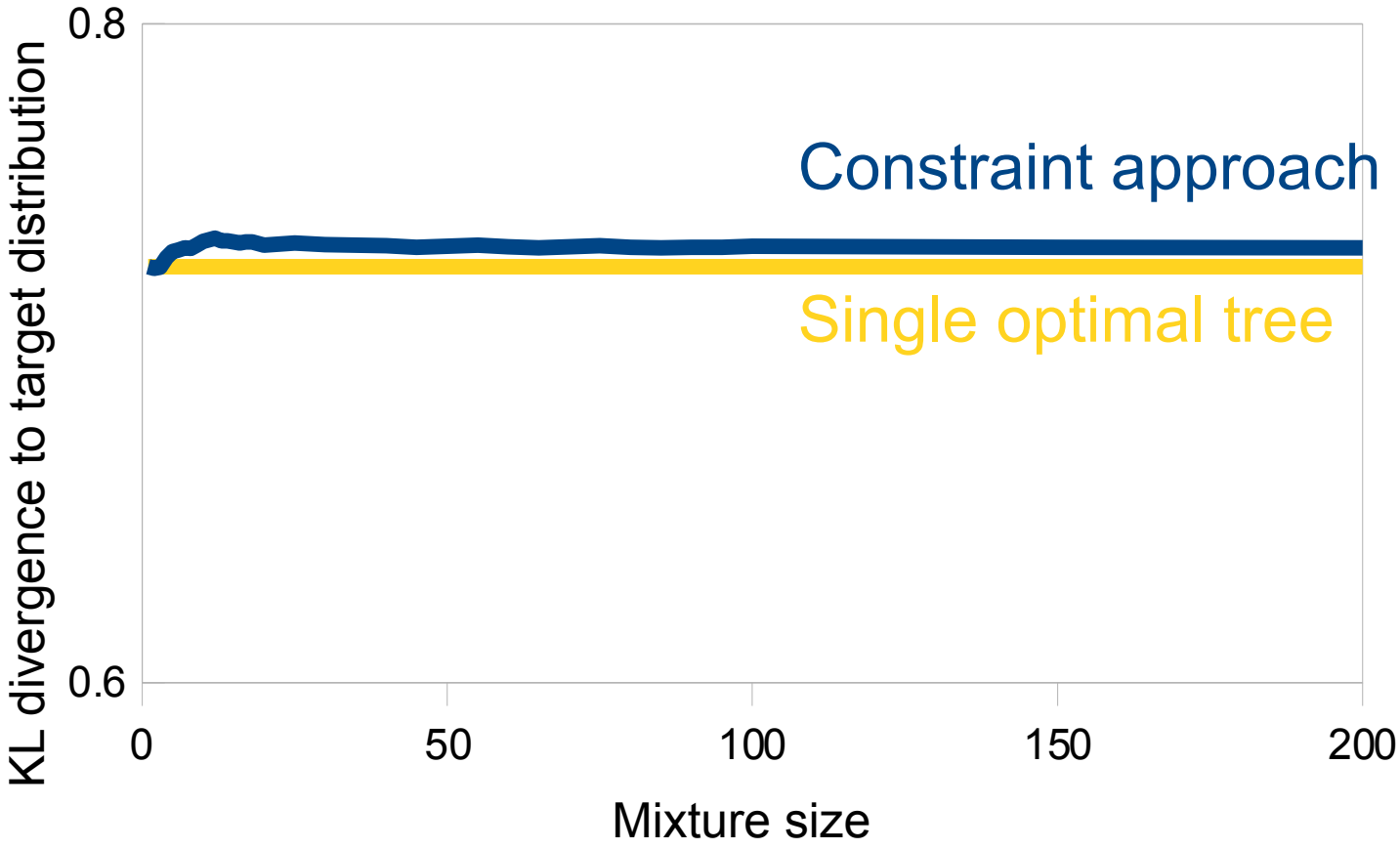
Results on small sets (50 samples) are promising...



- Better behavior than a mixture of random trees

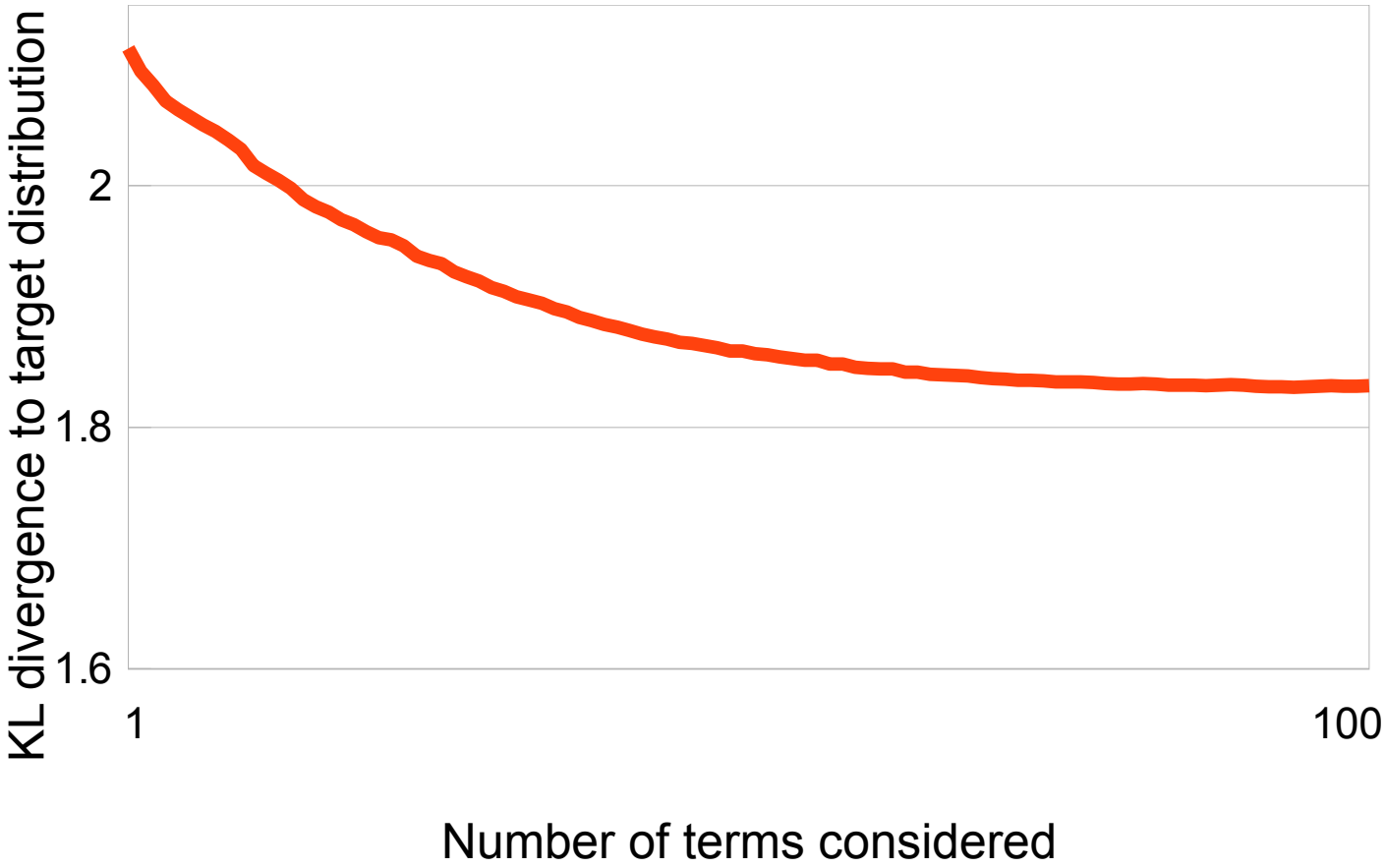
$\Rightarrow$ The approach seems interesting

results on big data sets (500 samples), not so much.



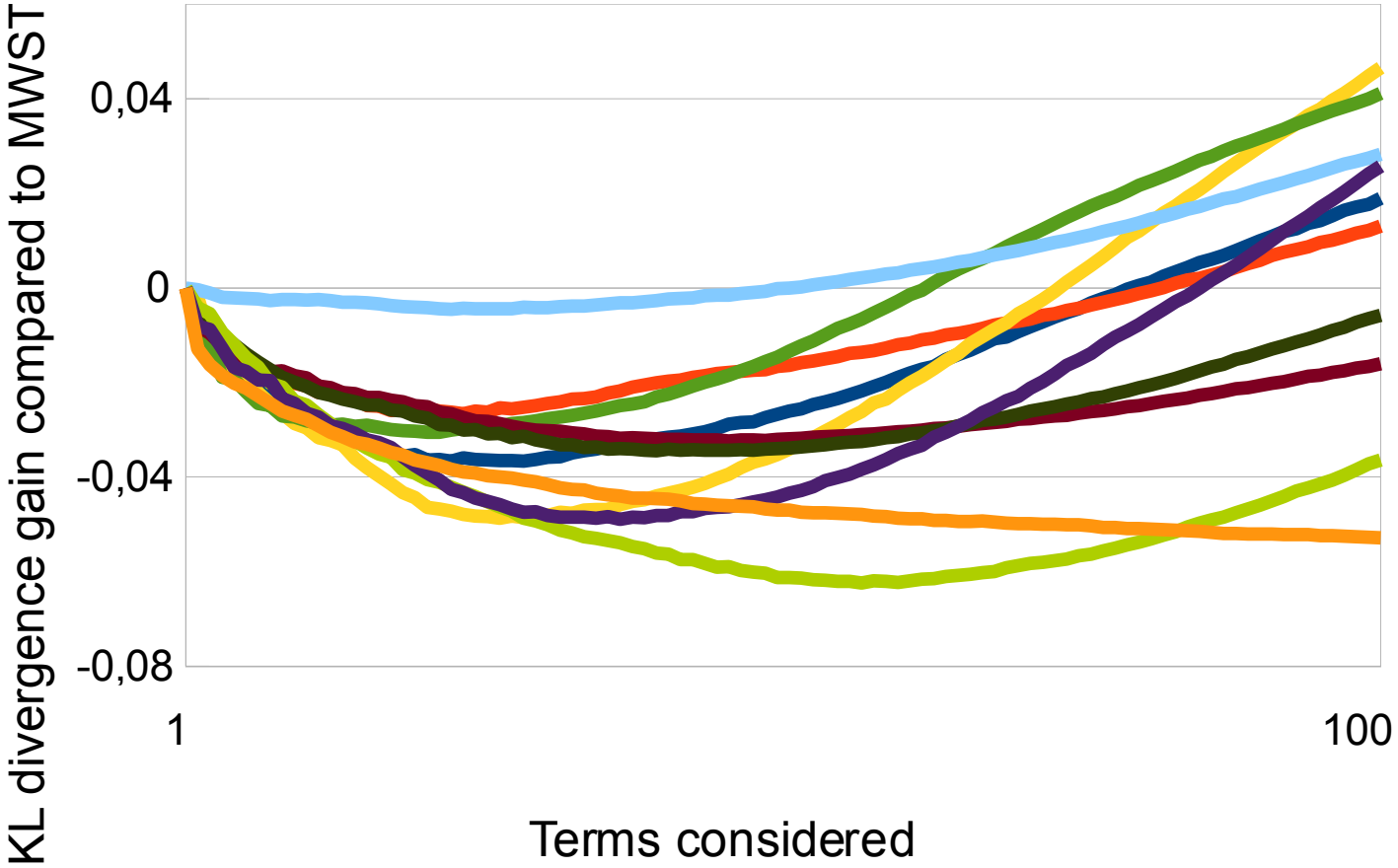
- convergence still fast
- to a worst value

Trimming a mixture shows the construction of the model for small data sets



- Improvement with the addition of each new term, although they are getting worse
- Convergence before the end

The behavior is different for larger data sets



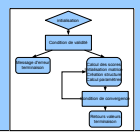
- 2 stages
  1. Increasingly improving
  2. Deteriorating and often overfitting
- Suggest using a weighting scheme

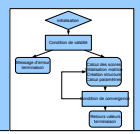
Conclusion

- Mixtures of trees scales well with the number of variables
- Our constraintapproach to building mixtures of trees shows real improvements on small data sets, marginal on big ones
- Possible ways for improvement :
  - other construction of trees
  - different penalty strategies
  - non fully developed trees
- Test on real problems needed

Simulation methodology

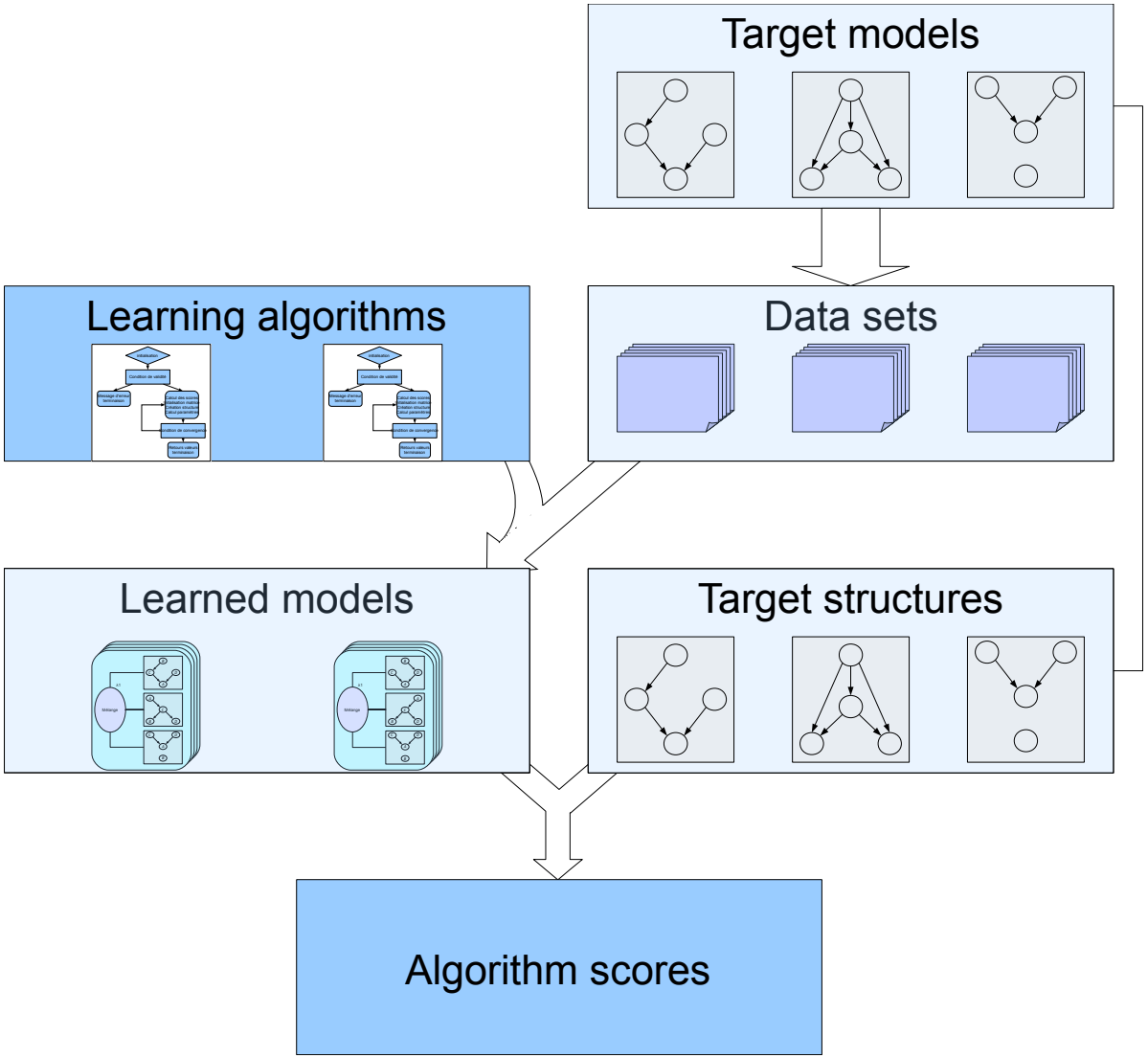
Learning algorithms



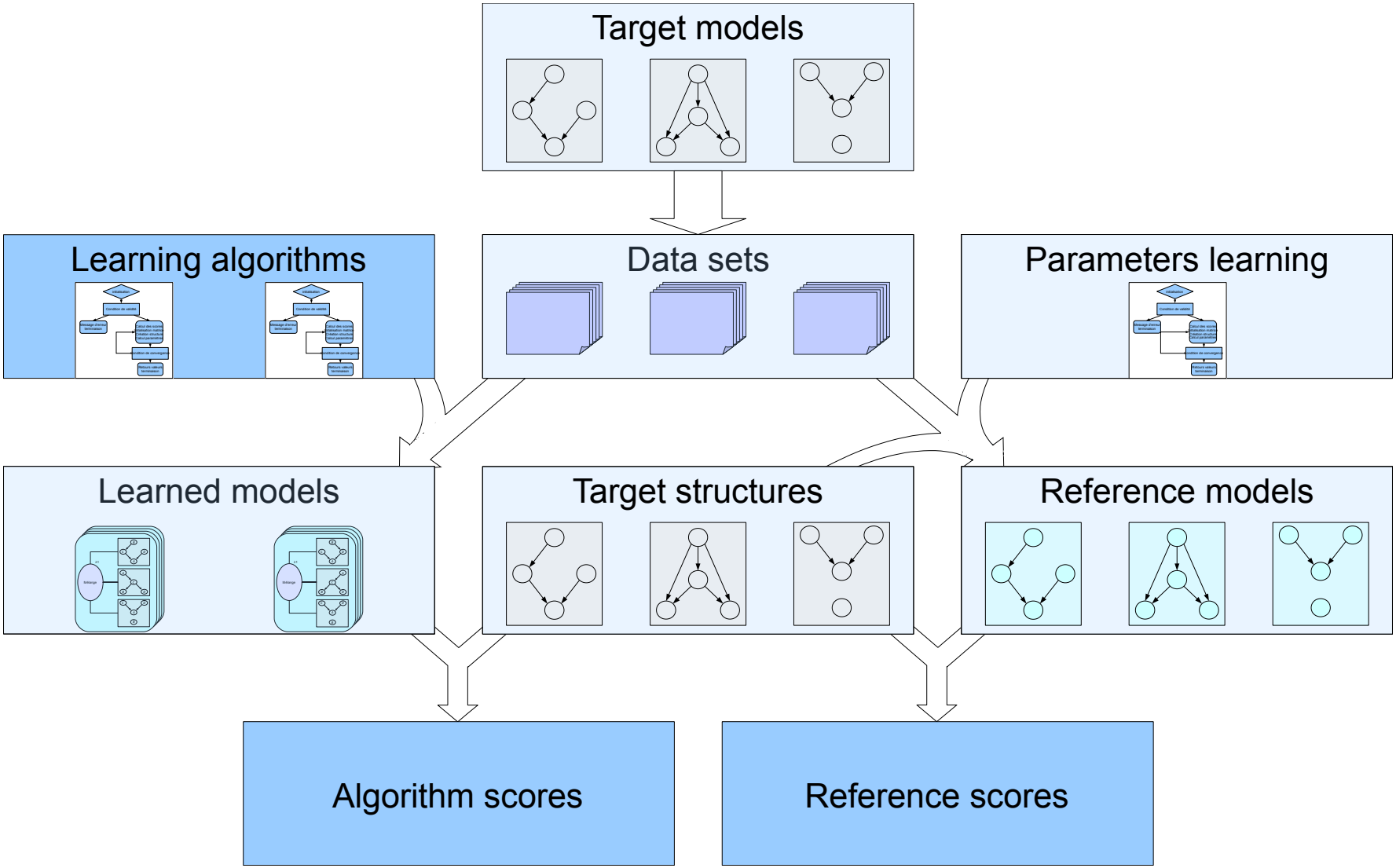


Algorithm scores

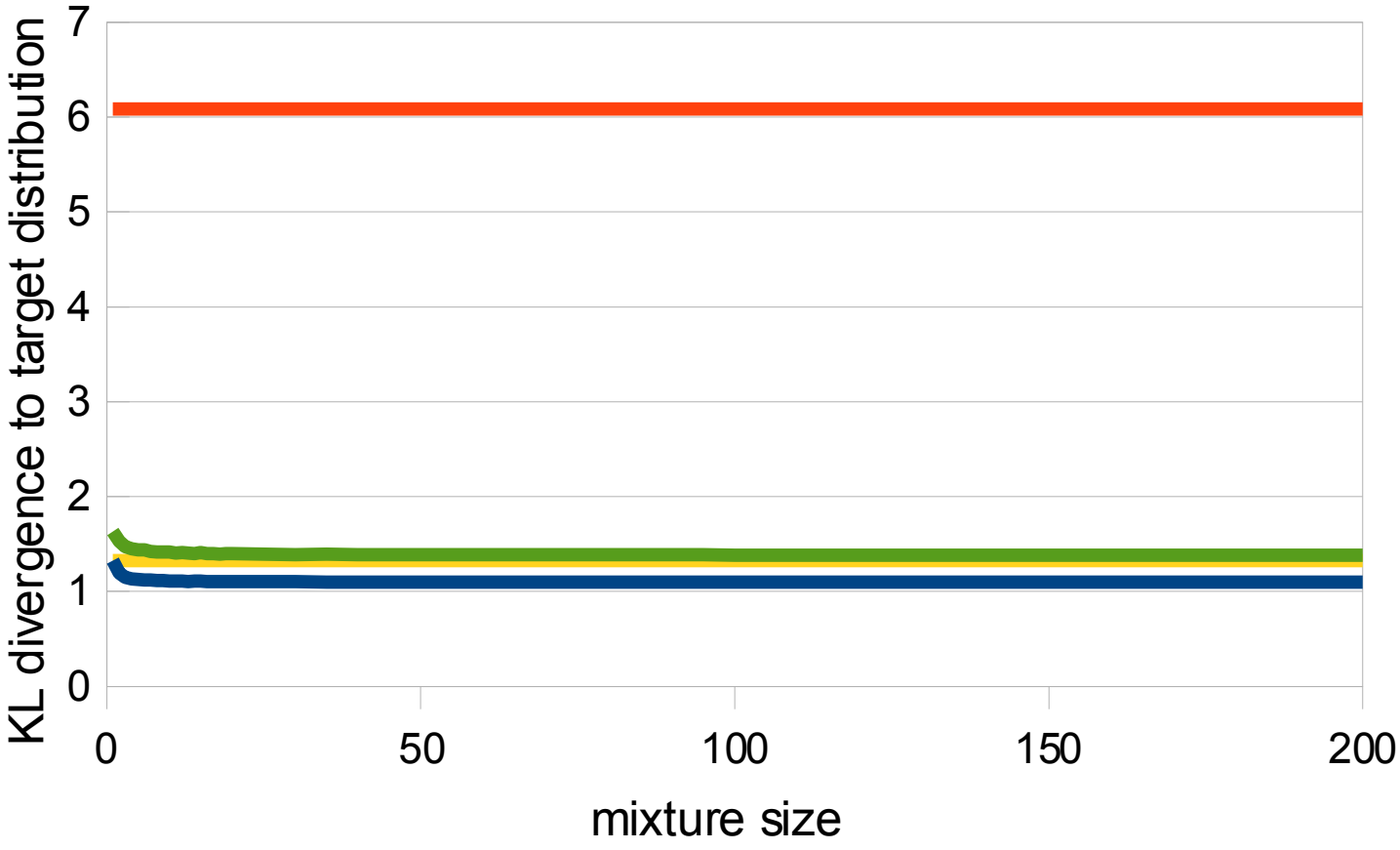
Simulation methodology



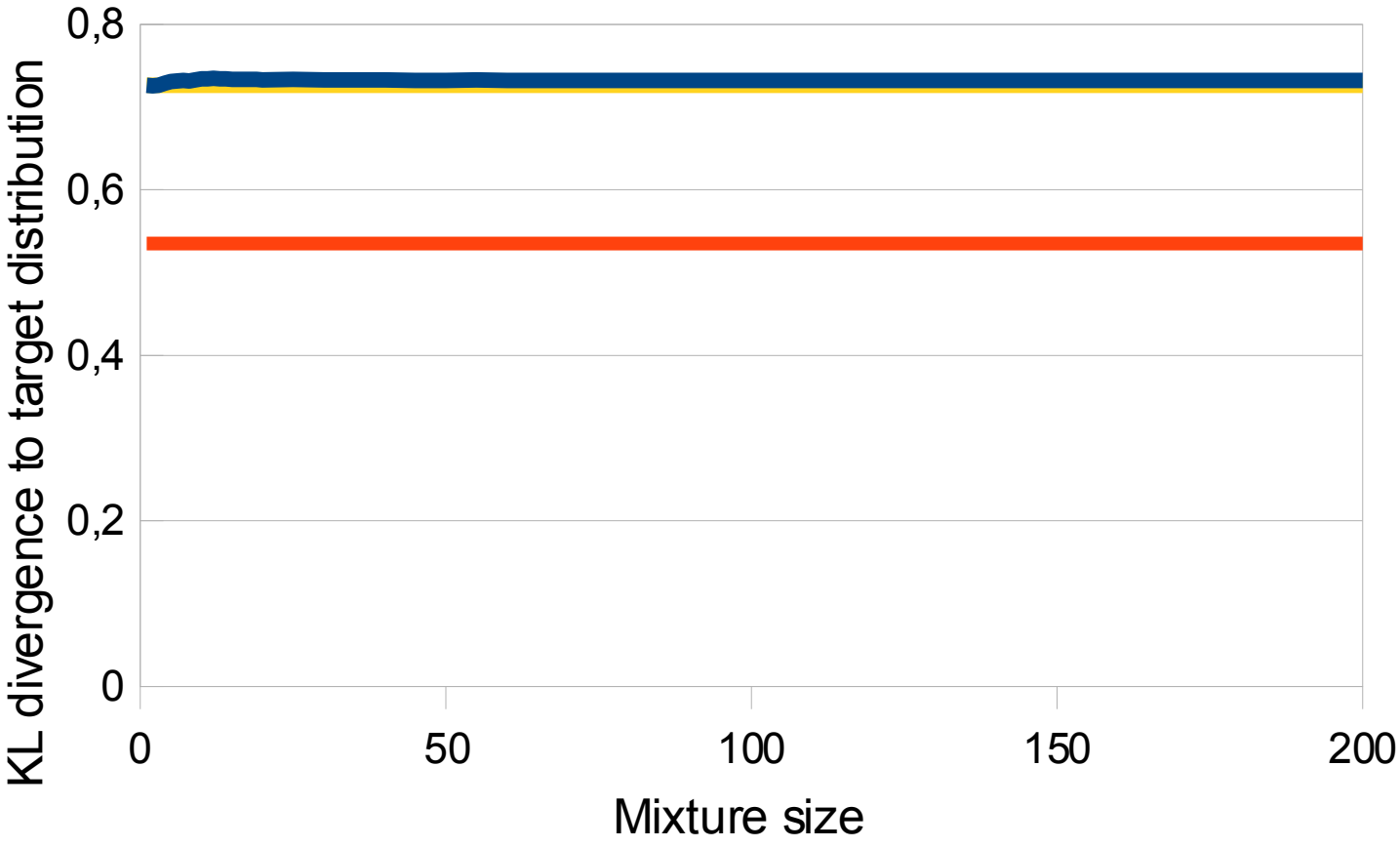
Simulation methodology



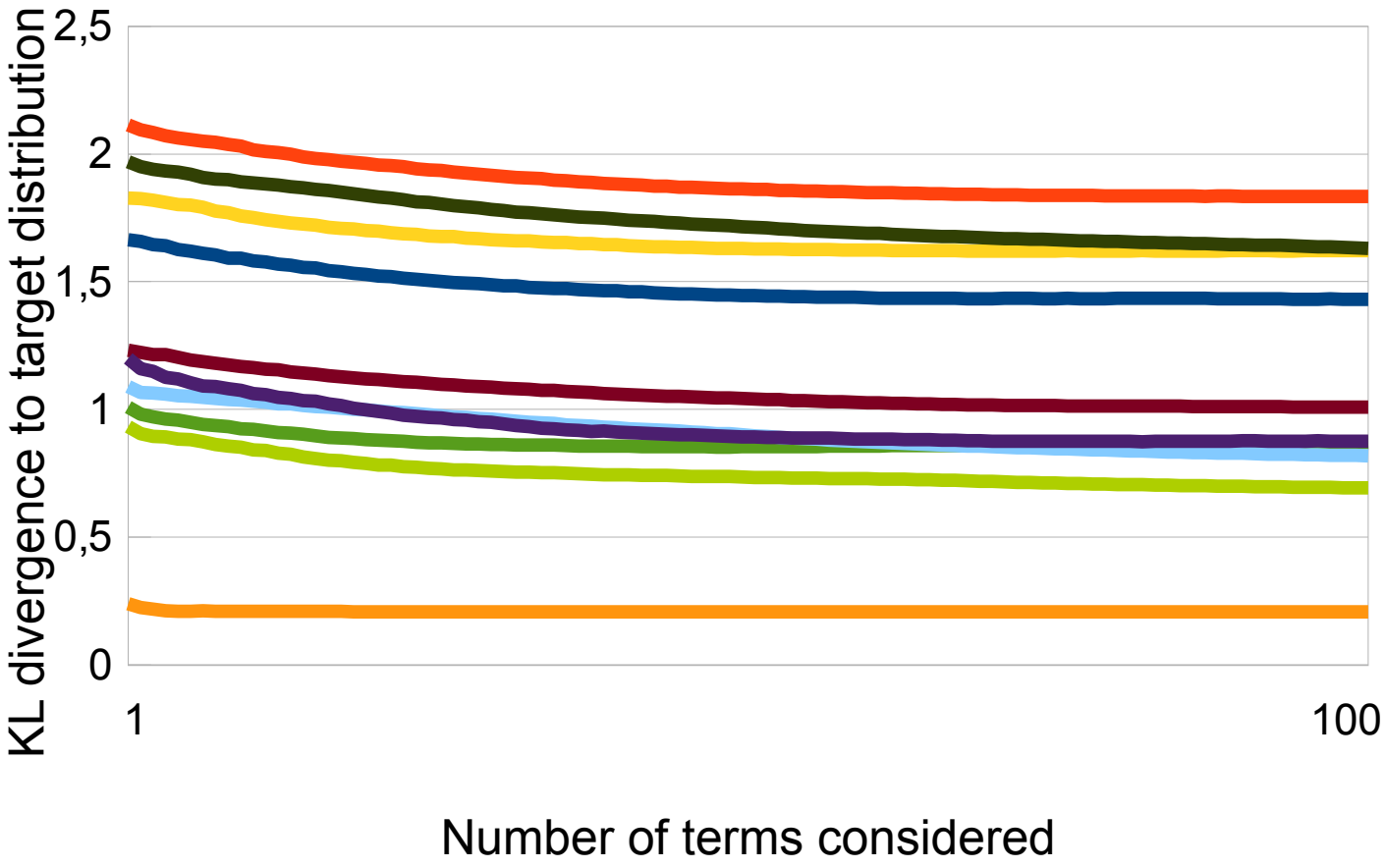
Results on small sets (50 samples)



Results on big data sets (500 samples)



Results on small sets (50 samples)



Results on big data sets (500 samples)