Variational and Sequential Inference over Combinatorial Spaces

Alexandre Bouchard-Côté*

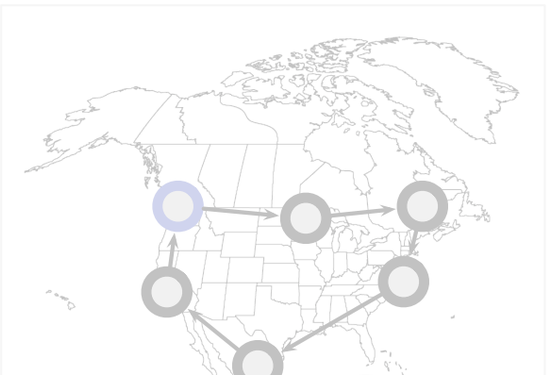
Collaborators:

Liangliang Wang*, Arnaud Doucet*, Michael I. Jordan†, Sriram Sankararaman**

* Department of Statistics, University of British Columbia

† Computer Science Division, University of California Berkeley

** Department of Genetics, Harvard University



Decoding the title of this talk

Variational and Sequential **Inference** over **Combinatorial Spaces**

Probabilistic inference

$$\sum_{x \in C} f(x)$$

Main motivations:
Computing a posterior distribution, counting

The set C is discrete, has exponential size, and is 'complicated'

Next few slides: Examples of such problems (and definition later)

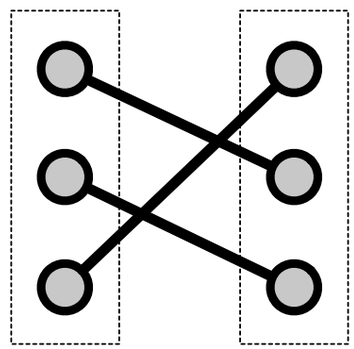
Familiar examples of inference problems over combinatorial spaces

Perfect bipartite matchings

C: set of bijection between A and B

x: one element in this set (right)

$f(x)$: multiply the the weight of each edge



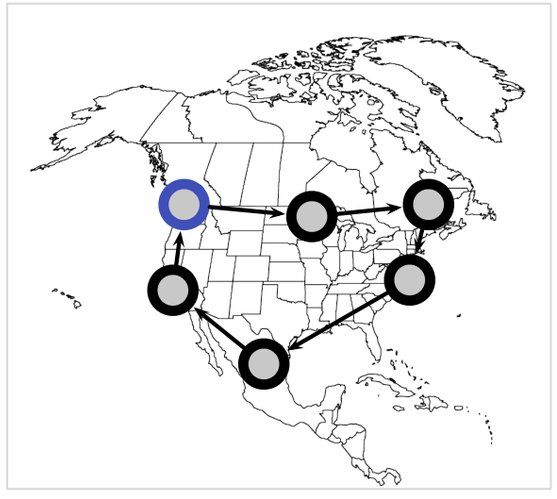
$A \longleftrightarrow B$

A twist on the Traveling Salesman Problem

C: Hamiltonian circuits on a graph

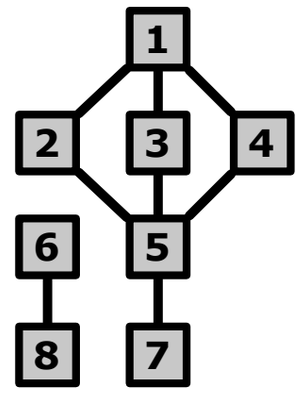
x: one element in this set (right)

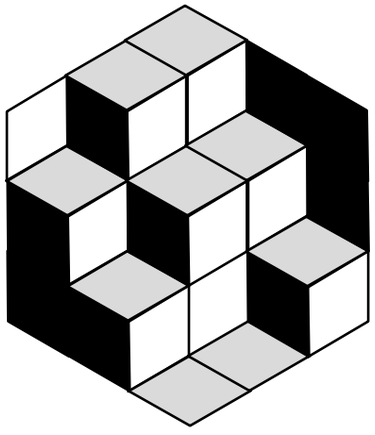
$f(x)$: multiply the the weight of each edge



Perhaps less familiar examples of inference problems over combinatorial spaces

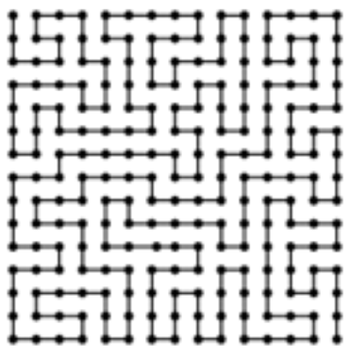
Counting *linearizations* of a partial orders P
Equivalently: number of ways to topologically sort a DAG (applications in ranking)





Counting *plane partitions*
Piling up cubes such that stacks are in increasing heights (applications in stat. mech.)

Counting *Self Avoiding Walks* (SAW)
Applications in modeling knot-theoretic properties of proteins



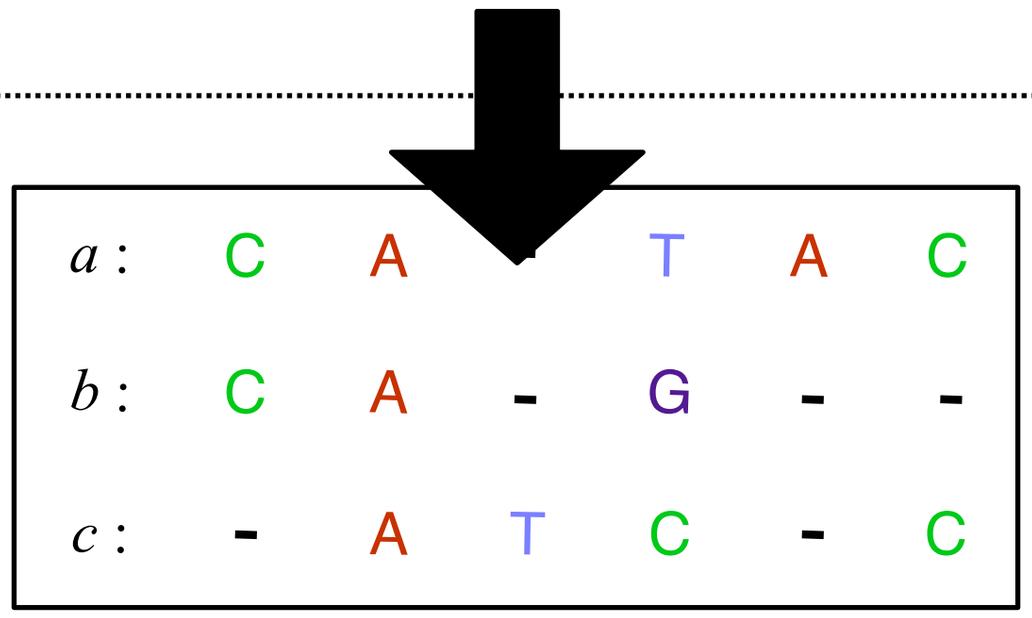
Two concrete problems in computational biology that motivated this work

Multiple Sequence Alignment (MSA)

UACUGAUCCUUAGCAUAUGCUUGCCAAAUUAAGCCAUGCAUCUAACGACGGCCGGUACAUGAAGAAUGGCUCAU

CCUGGUUGAUCCUGCCAGUAGCAUAUGCUUGUCUCAAAGAUUAAGCCAUGCAUGUCUAAG

CCUGCCGGAGGCCAUUGCUAUUGGGAUUCGAUUUAGCCAUGCUAGUCGCACGAGUUUAGACUCGUGGCGAAUAGCU



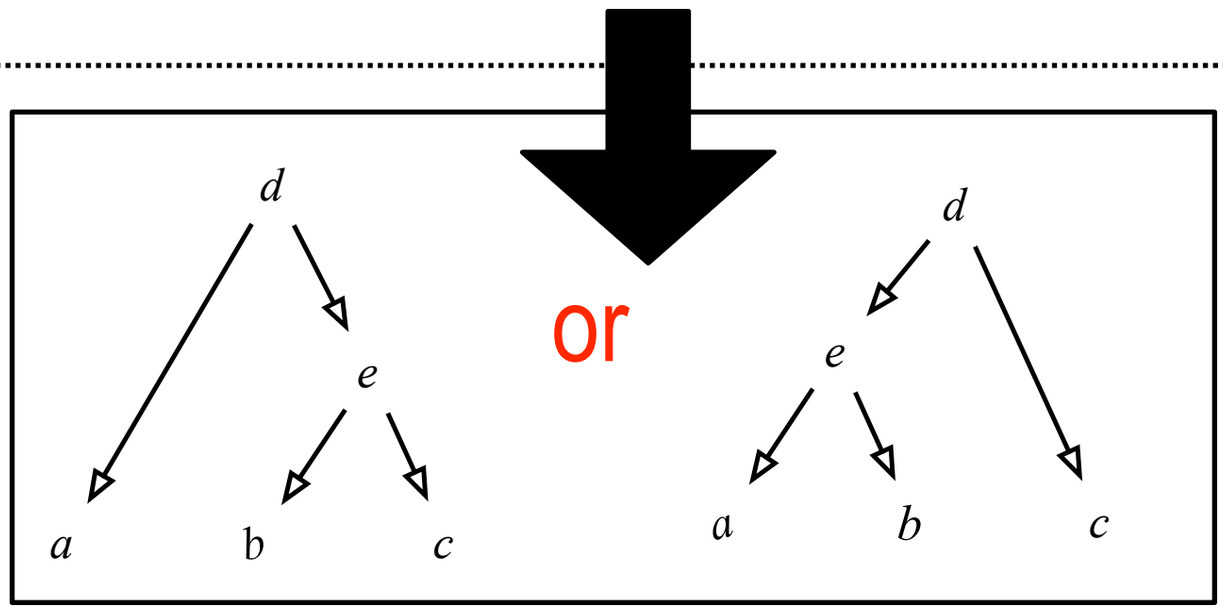
Two concrete problems in computational biology that motivated this work

Phylogenetic tree inference

UACUGAUCCUUAGCAUAUGCUUGCCAAAUUAAGCCAUGCAUCUAACGACGGCCGGUACAUGAAGAAUGGCUCAU

CCUGGUUGAUCCUGCCAGUAG.CAUAUGCUUGUCUCAAAGAUUAAGCCAUGCAUGUCUAAG

CCUGCCGGAGGCCAUUGCUAUUGGGAUUCGAUUUAGCCAUGCUAGUCGCACGAGUUUAGACUCGUGGCGAAUAGCU



Decoding the title of this talk

Variational and **Sequential** Inference over Combinatorial Spaces

Previous work: for inference in combinatorial spaces, mostly MCMC (Markov Chain Monte Carlo)

Variational methods: cast probabilistic inference problems as a constrained, convex optimization problem

Sequential methods: incrementally approximate complicated distributions by importance sampling and resampling

Both approaches have been very successful for inference in graphical models, but not in combinatorial space...

Outline: two frameworks for probabilistic inference over combinatorial spaces

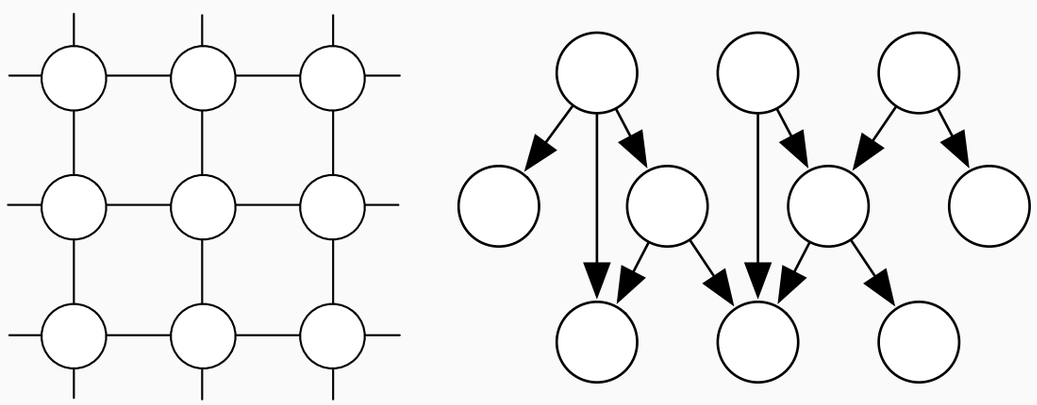
- **Variational inference by Measure Factorization**
  - User's guide
  - Theoretical foundations
  - Experiments on multiple sequence alignment
- Poset Sequential Monte Carlo
  - User's guide
  - Theoretical foundations
  - Experiments on phylogenetic tree inference

Variational inference over combinatorial spaces: a user's guide

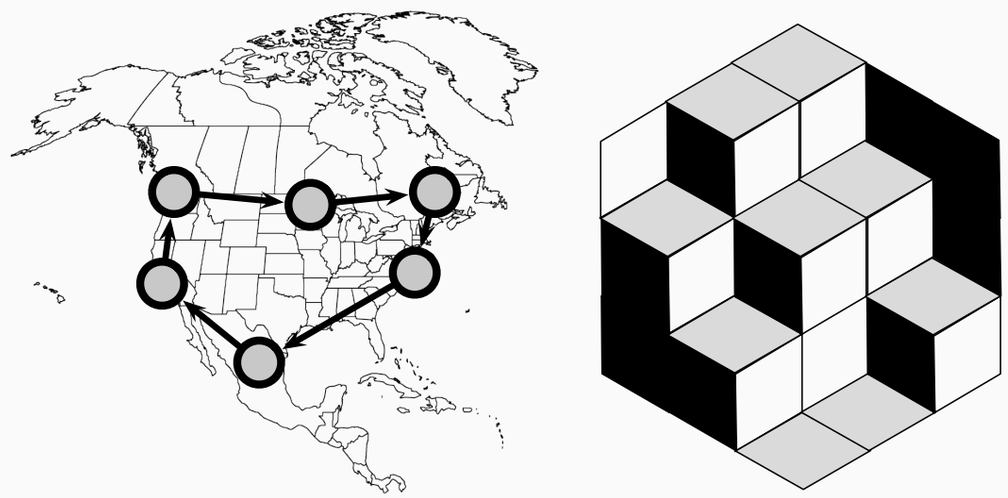
Setup: hard inference problem over an exponential family

$$Z_\theta = \sum_x \underbrace{\exp\langle T(x), \theta\rangle}_{f_\theta(x)} \quad \mathbf{1}[x \in S]$$

Large treewidth



Combinatorial space S



Variational inference over combinatorial spaces: a user's guide

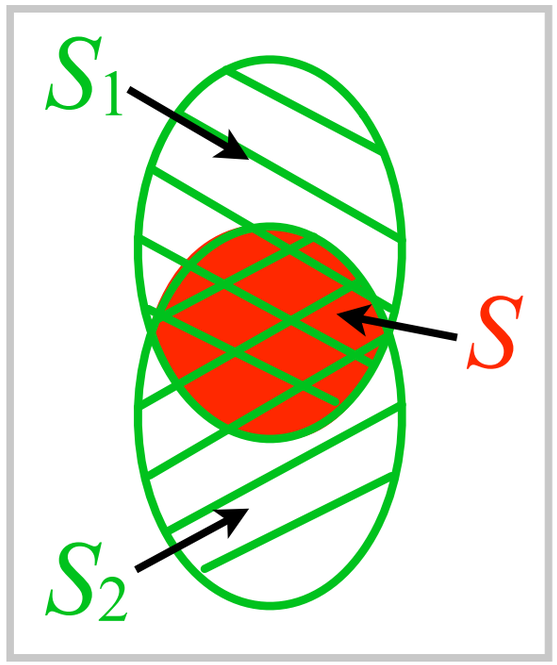
Goal:

$$Z = \sum_{x \in S} f(x) \quad \text{(intractable)}$$

Assumption: *('Measure factorization')*

$$S = \bigcap S_i \quad \text{such that:}$$

$$Z_i = \sum_{x \in S_i} f(x) \quad \text{is tractable}$$



Consequence: if you can find such decomposition, our framework gives a way to turn variational algorithms defined for graphical models into algorithms that can be used for combinatorial spaces

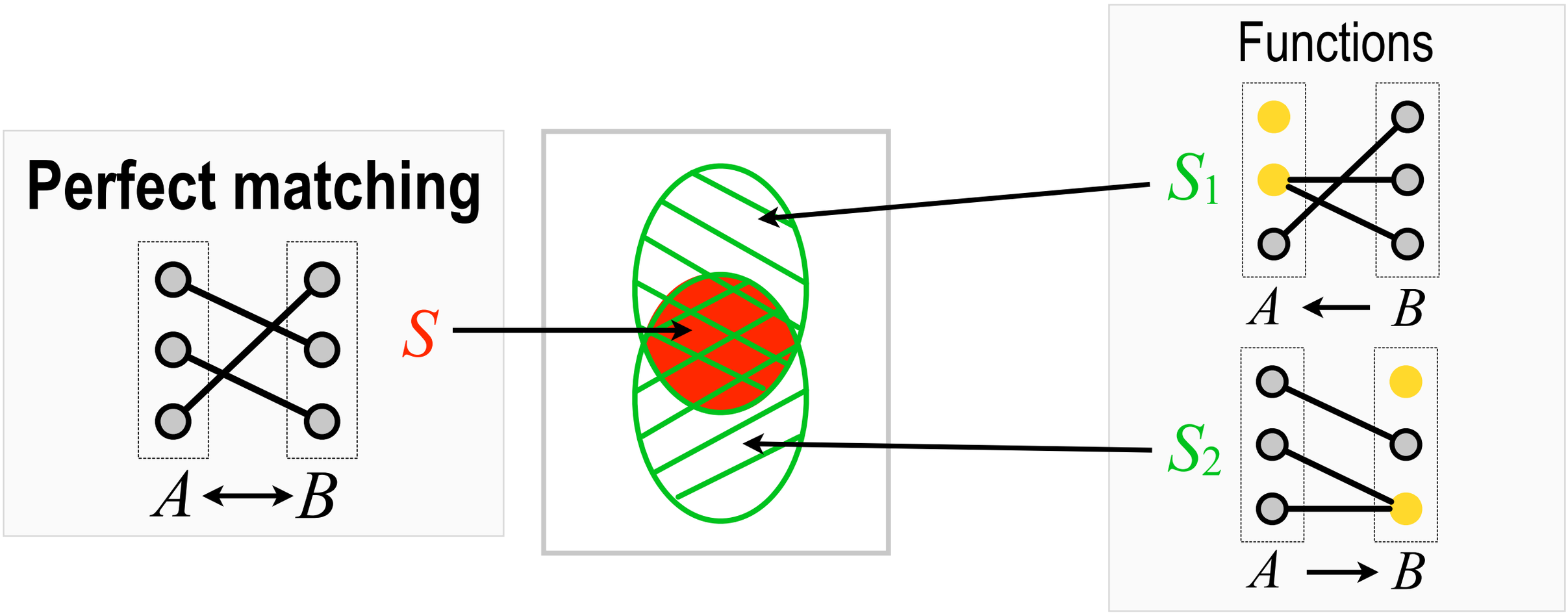
Advantages

Termination: most variational algorithms are guaranteed to converge in a finite number of steps (very small in practice)

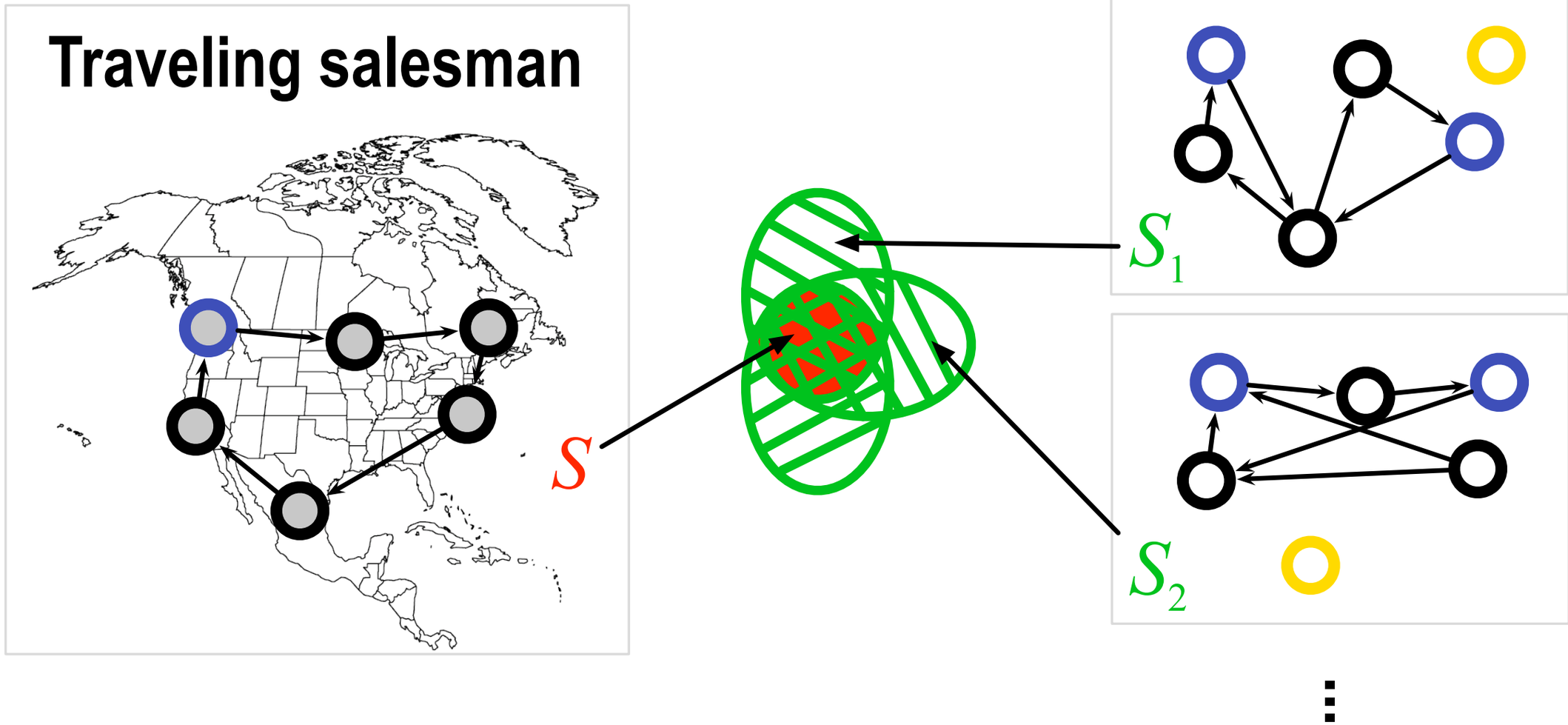
Bounds: using a specific variational algorithm, we can guarantee that the estimated value for Z is an upper bound

Empirically: good accuracy on a range of problems

Examples of tractable measure factorizations

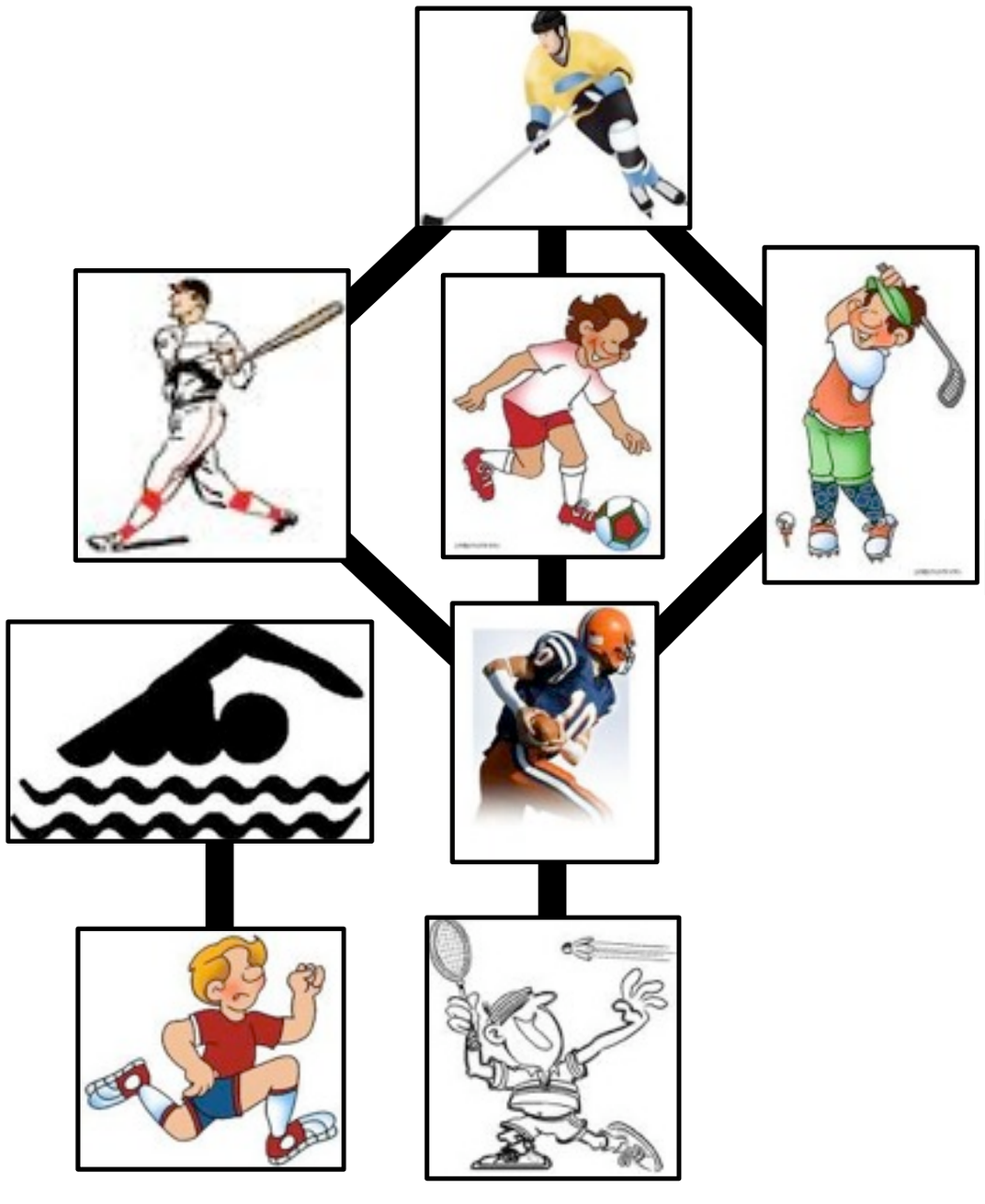


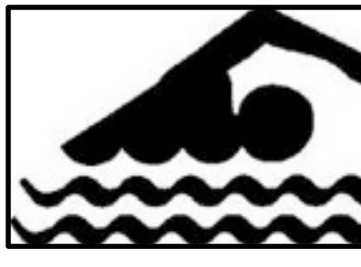
Examples of tractable measure factorizations



Examples of tractable measure factorizations

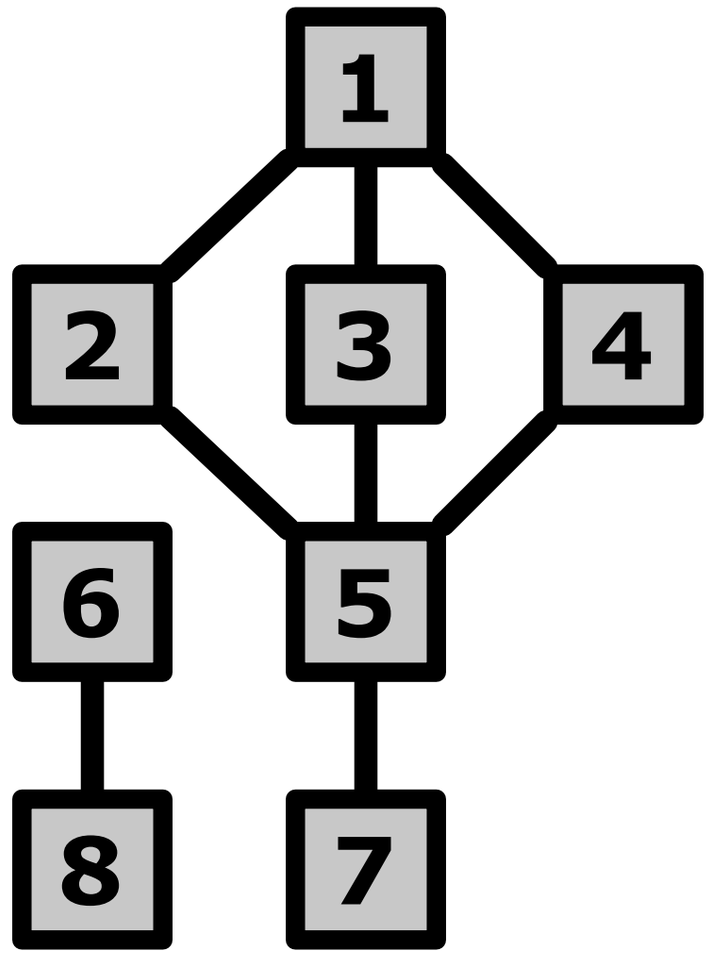
More advanced example: linearizations of partial orders





Examples of tractable measure factorizations

More advanced example: linearizations of partial orders

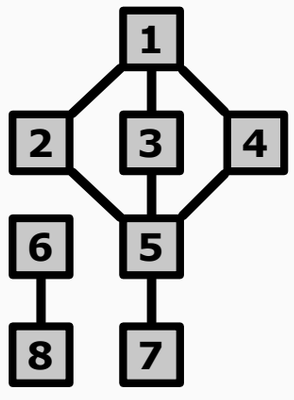


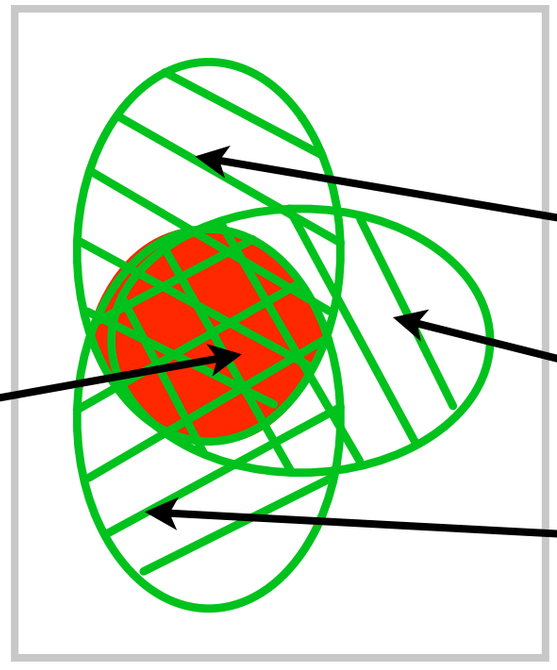
‘Hasse diagram’

Examples of tractable measure factorizations

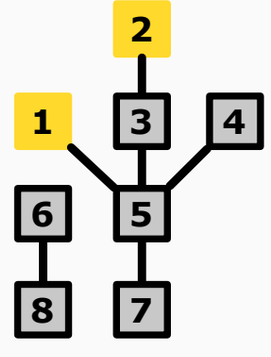
More advanced example: linearizations of partial orders

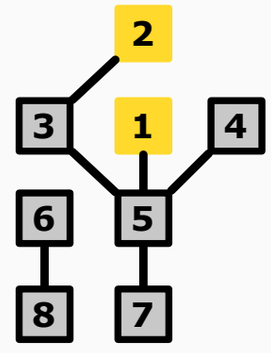
Linearization of Posets

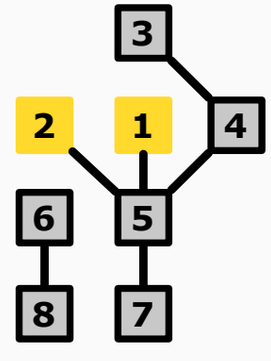




Forest cover

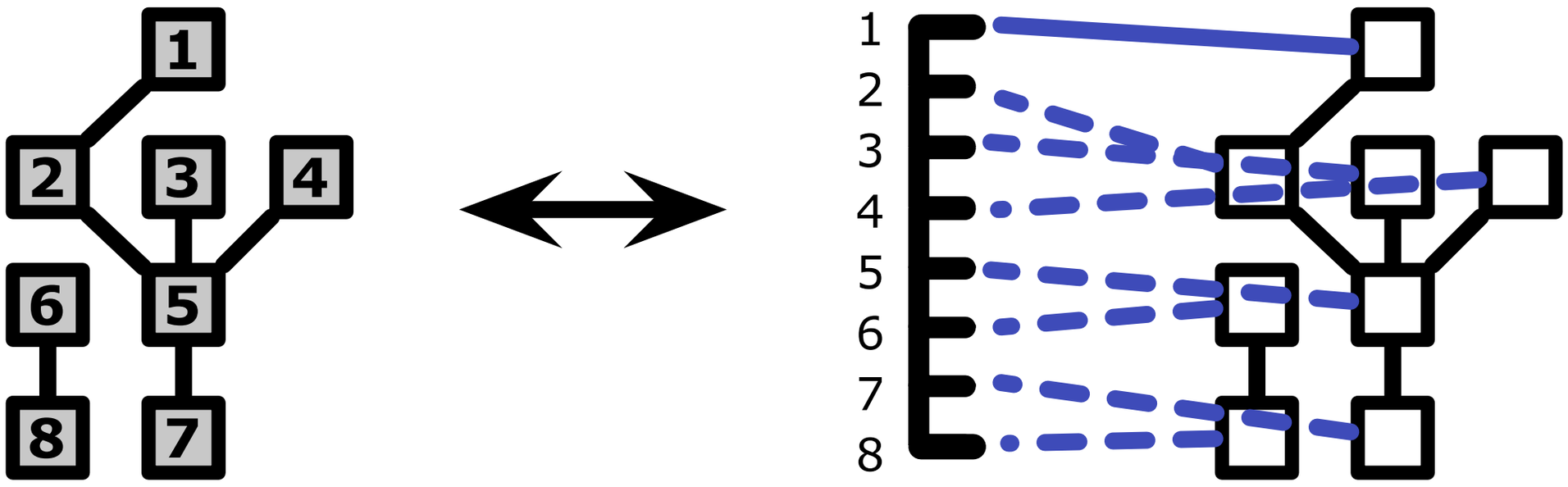






Examples of tractable measure factorizations

More advanced example: linearizations of partial orders



Under the hood

Background on exponential families:

Sufficient statistic

Parameter

$$\mathbb{P}(\boldsymbol{X}_{\boldsymbol{\theta}} \in B) = \sum_{x \in B} \exp\{\langle T(x), \boldsymbol{\theta} \rangle - A(\boldsymbol{\theta})\}\nu(x),$$

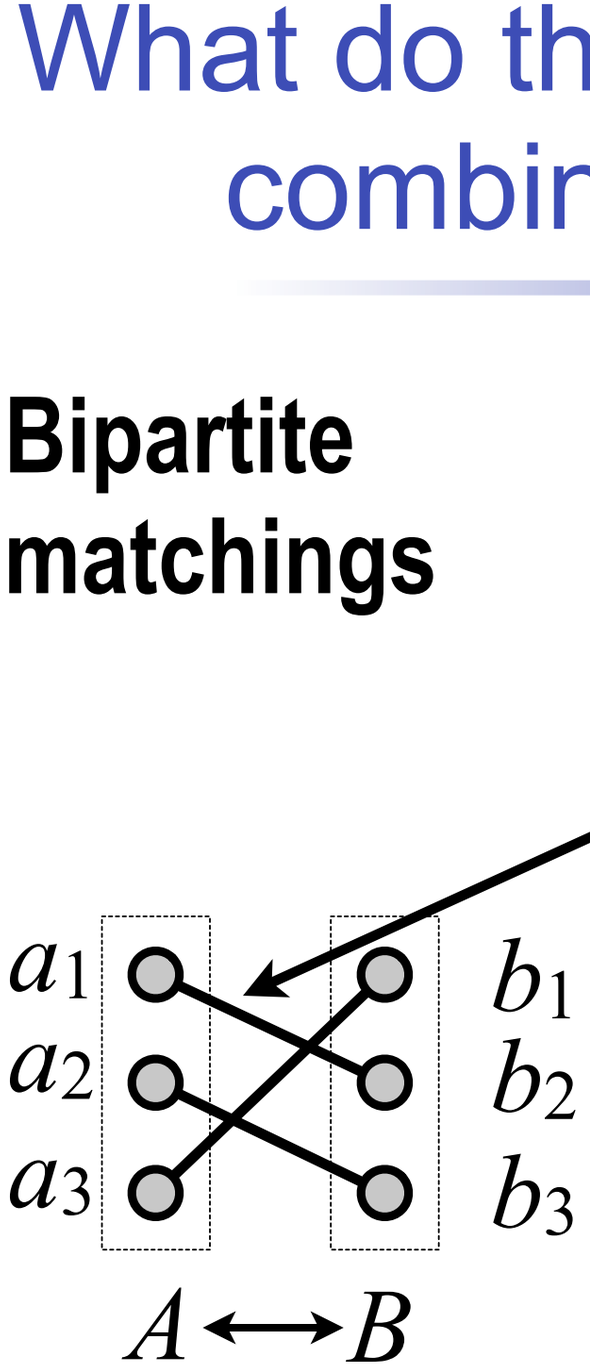
$$A(\boldsymbol{\theta}) = \log \sum_{x \in \boldsymbol{\mathcal{X}}} \exp\{\langle T(x), \boldsymbol{\theta} \rangle\}\nu(x),$$

Log partition function

The base measure is an indicator over the combinatorial space C of interest

What do the sufficient statistic look like in the combinatorial spaces we have seen?

Bipartite matchings



$$T(x) = \begin{bmatrix} \mathbf{1}[a_1 \text{ linked to } b_1] \\ \mathbf{1}[a_1 \text{ linked to } b_2] \\ \mathbf{1}[a_1 \text{ linked to } b_3] \\ \vdots \end{bmatrix} \begin{bmatrix} \theta_{1,1} \\ \theta_{1,2} \\ \theta_{1,3} \\ \vdots \end{bmatrix} \Bigg\} J$$

Rich sufficient statistic condition: for all J-dimensional vector s, there is at most one configuration x with $T(x) = s$

Important property

The gradient of the log partition function is equal to the moments:

$$\nabla A(\boldsymbol{\theta}) = \mathbb{E}[T(\boldsymbol{X}_{\boldsymbol{\theta}})]$$

The hessian of the log partition function is equal to the covariance matrix:

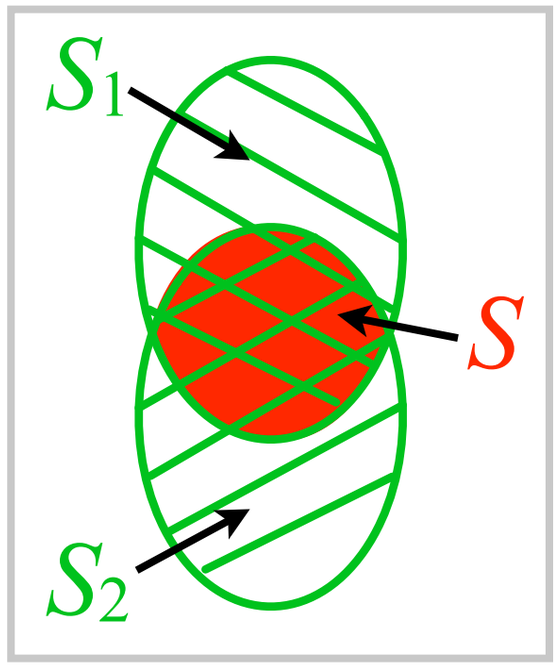
$$H(A(\boldsymbol{\theta})) = \mathbf{V}ar[T(\boldsymbol{X}_{\boldsymbol{\theta}})].$$

Consequence: A is a convex function

Connection with measure factorizations

Inference on the original problem:

$$X \sim \exp\left\{\langle T(x), \theta\rangle - A(\theta)\right\} \prod_i \mathbf{1}[x \in S_i]$$



Inference on a single factor *i*:

$$X_i \sim \exp\left\{\langle T(x), \theta\rangle - A_i(\theta)\right\} \mathbf{1}[x \in S_i]$$

Tractable assumption: can compute (e.g. using DP)

$$\nabla A_i(\theta) = \mathbb{E}[T(X_i)]$$

$$= \sum_x T(x) \exp\left\{\langle T(x), \theta\rangle - A_i(\theta)\right\} \mathbf{1}[x \in S_i]$$

Example of an algorithm derived using our framework: Measure Factorization Belief Prop.

BPMF$(\boldsymbol{\theta}, A_1, \ldots, A_I)$

1: $\boldsymbol{\zeta}_{i,j}^{(1)} = 0$
2: **for** $t = 1, 2, \ldots, T$ **do**
3: $\quad \bar{\boldsymbol{\xi}}_i^{(t)} = \boldsymbol{\theta} + \sum_{i':i' \neq i} \boldsymbol{\zeta}_{i'}^{(t-1)}$
4: $\quad \boldsymbol{\zeta}_i^{(t)} = \text{logit}\left(\nabla A_i\left(\bar{\boldsymbol{\xi}}_i^{(t)}\right)\right) - \bar{\boldsymbol{\xi}}_i^{(t)}$
5: **end for**
6: **return** $\hat{\boldsymbol{\mu}} = \text{logistic}\left(\boldsymbol{\theta} + \sum_i \boldsymbol{\zeta}_i^{(T)}\right)$

Form a set of transformed parameters (one for each factor)

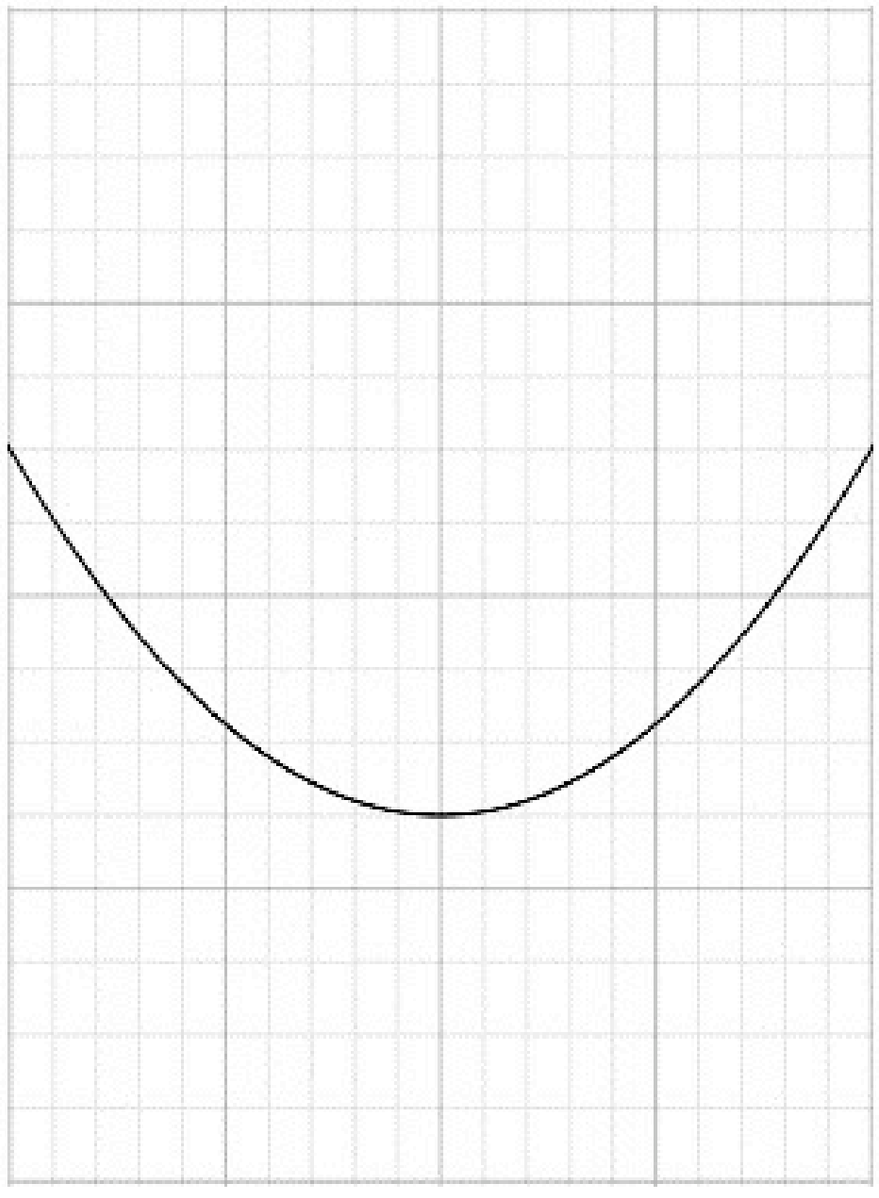
Compute moments for each factor using the new transformed parameters

How did we get to this algorithm?

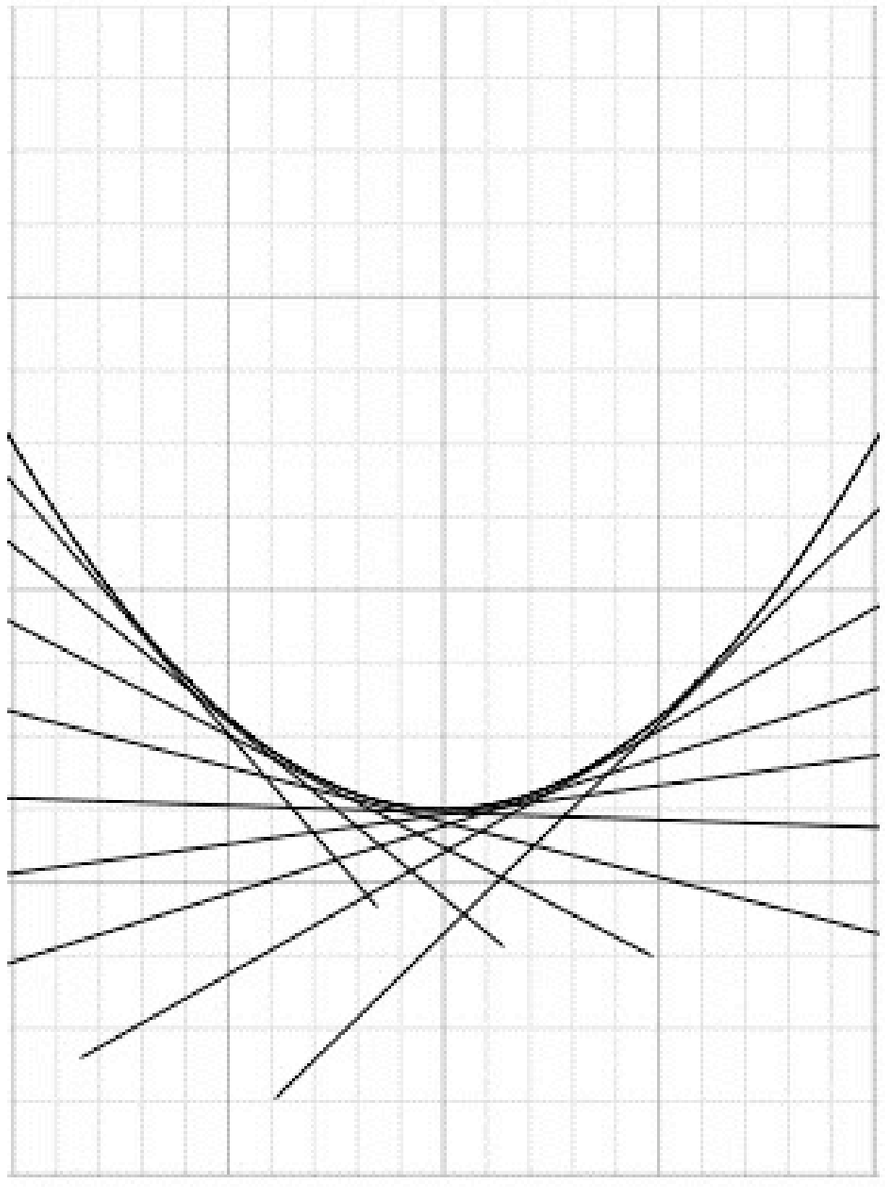
Plan:

- Review of the variational framework for graphical models
- How to cast combinatorial problems into something that fits this framework

Representation of convex functions



Standard / pointwise encoding



Encoded by intercepts of the supporting tangents

Connexion: Legendre-Fenchel transformation

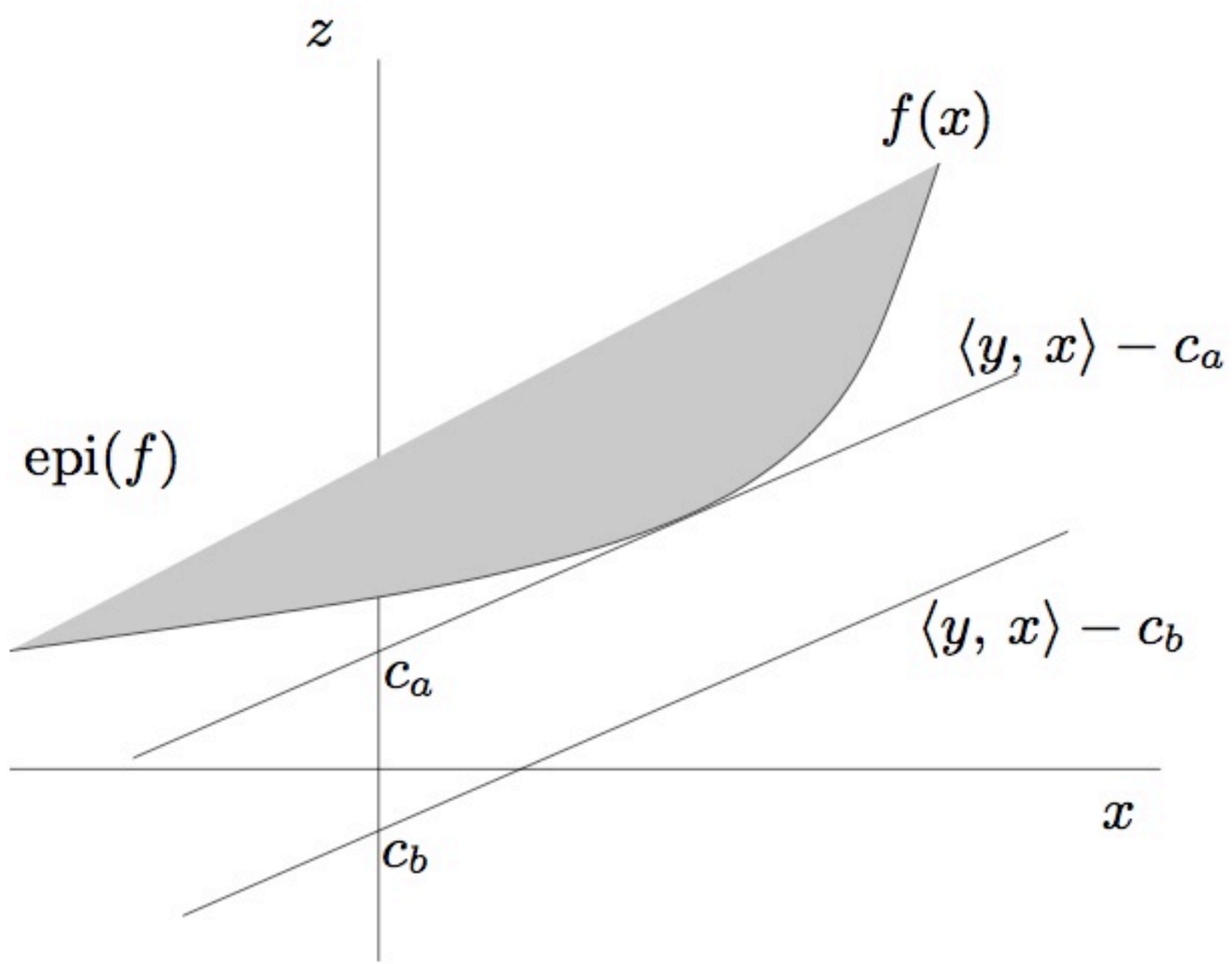
An operator (a function that takes a function and transforms it into another function) denoted by *

$$f^*(y) := \sup_{x \in \mathrm{dom}(f)} \{\langle y, x \rangle - f(x)\},$$

Warning: for pedagogical reasons, assume for now that f is univariate, twice differentiable and strictly convex (can be made more general!!)

Intuition

Suppose I give you a tangent/supporting plane. Encoding a convex function can be done by giving the intercept c_a



Why this particular 'encoding'?

Theorem:
When f is convex (and lower semi-continuous): $f^{**} = f$

Consequence: the log partition function satisfies $A^{**} = A$

What we will do with this: First, apply the definition of Fenchel dual to the function A^*, getting:

$$A^{**}(\boldsymbol{\theta}) = \sup\{\langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle - A^*(\boldsymbol{\mu}) : \boldsymbol{\mu} \in \mathscr{M}\},$$

||

$A(\theta)$

This is just the domain of A^*

Done?

Convex function are easy to optimize, right?

$$A(\boldsymbol{\theta}) = \sup\{\langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle - A^*(\boldsymbol{\mu}) : \boldsymbol{\mu} \in \mathscr{M}\},$$

Problem: there are exponentially many constraints here

Constraints: realizable moments

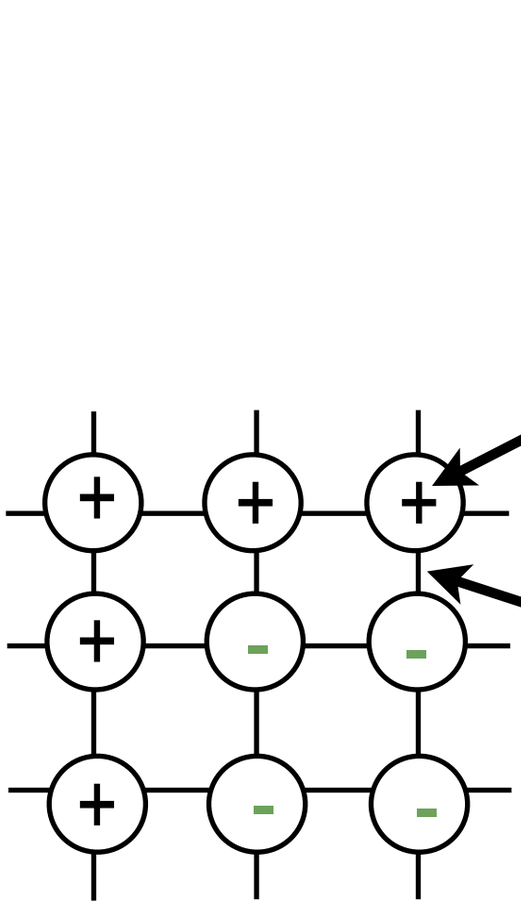
Suppose I give you a J-dimensional vector μ and I claim it is the moment of a distribution for some parameters θ (which I don't give you θ, but the sufficient statistics are known)

I.e. claim there is a θ such that:

$$\mu = \mathrm{E}[T(X_\theta)]$$

What could you check?

Sufficient statistic for graphical models



One node

Pairs of nodes

$$T(x) = \begin{bmatrix} \mathbf{1}[x_{1,1} = +] \\ \mathbf{1}[x_{1,1} = \text{-}] \\ \mathbf{1}[x_{1,2} = +] \\ \vdots \\ \mathbf{1}[x_{1,1} = +, x_{1,2} = +] \\ \mathbf{1}[x_{1,1} = +, x_{1,2} = \text{-}] \\ \vdots \end{bmatrix} \begin{bmatrix} \theta_{1,1,+} \\ \theta_{1,1,\text{-}} \\ \theta_{1,2,+} \\ \vdots \\ \theta_{1,1,+;1,2,+} \\ \theta_{1,1,+;1,2,\text{-}} \\ \vdots \end{bmatrix} J$$

Constraints: realizable moments

Suppose I give you a J-dimensional vector μ and I claim it is the moment of a distribution for some parameters θ (which I don't give you θ, but the sufficient statistics are known)

I.e. claim there is a θ such that:

$$\mu = \mathrm{E}[T(X_\theta)]$$

What could you check?

$$\mu_{1,1,+} = \sum_{x \in \{+,-\}} \mu_{1,1,+;1,2,x} \quad \text{etc.}$$

$$\begin{bmatrix} \mu_{1,1,+} \\ \mu_{1,1,-} \\ \mu_{1,2,+} \\ \vdots \\ \mu_{1,1,+;1,2,+} \\ \mu_{1,1,+;1,2,-} \\ \vdots \end{bmatrix}$$

Constraints: realizable moments

Theorem: for *trees*, μ is a realizable moment if and only if *pairwise* marginalization conditions are met

In *cyclic* graphs, higher order marginalization constraints needed!

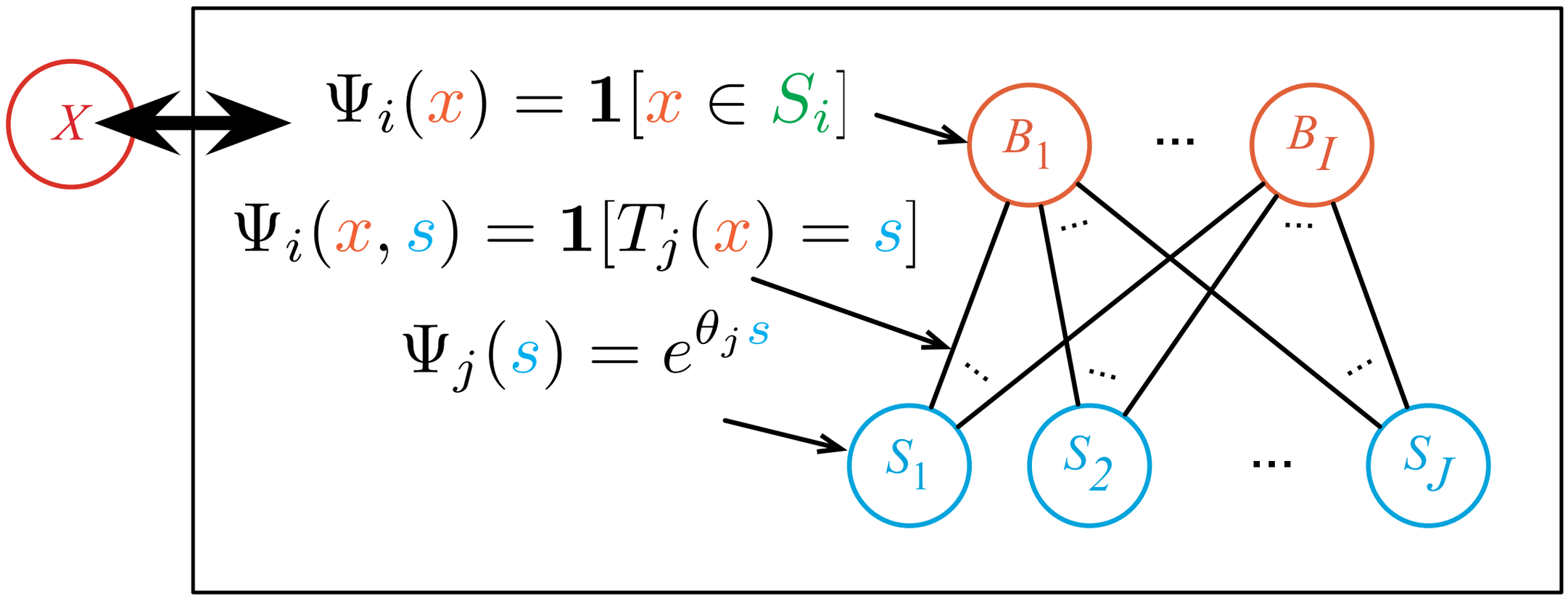
Belief propagation

Main idea: even if there are cycles, use only pairwise marginalization constraints (a relaxation of the optimization problem)

It can be shown that optimizing this relaxed problem yields the familiar BP algorithm (actually, the objective also needs to be simplified a little bit)

Back to combinatorial spaces...

First step: we showed that under the rich sufficient statistic condition, the following two distributions are equivalent:

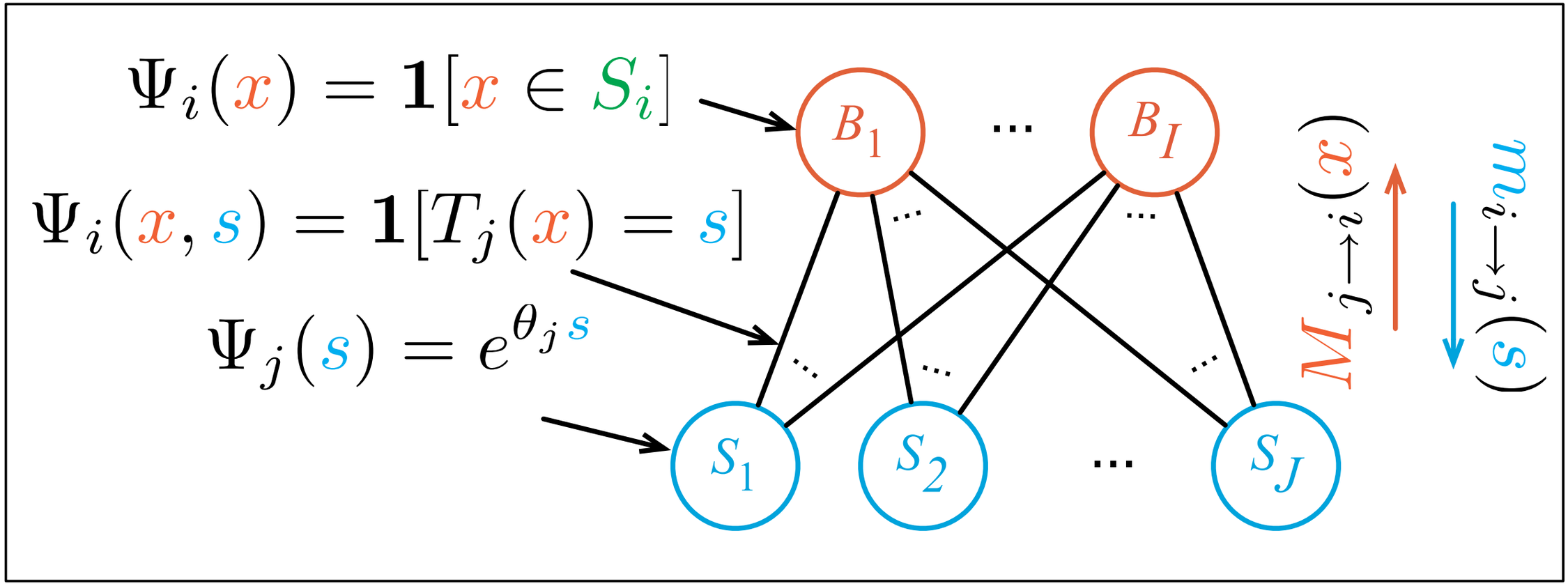


$$X \sim \exp\Big\{\langle T(x), \theta\rangle - A(\theta)\Big\} \prod_i \mathbf{1}[x \in S_i]$$

$$X_i \sim \exp\Big\{\langle T(x), \theta\rangle - A_i(\theta)\Big\}\ \mathbf{1}[x \in S_i]$$

Back to combinatorial spaces

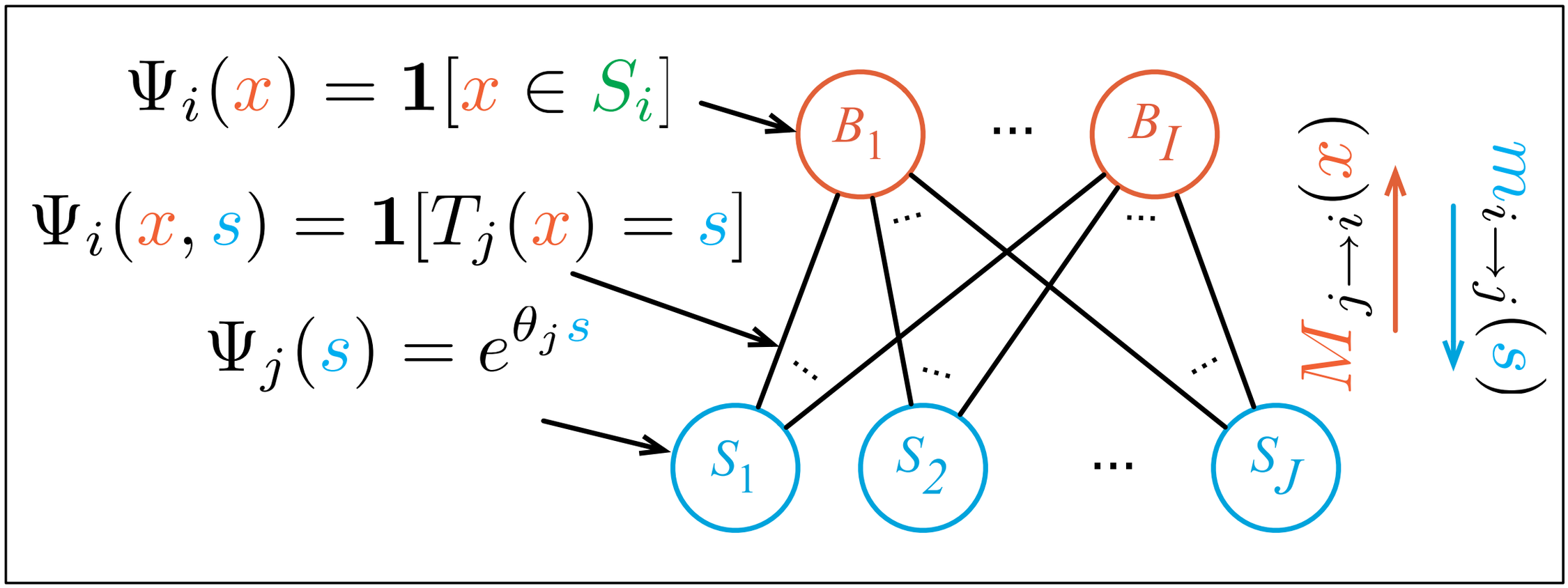
Second step: pick a variational algorithm an look at the messages sent in the equivalent bipartite graph



Back to combinatorial spaces

Third step: address one last problem...

Storing a single message $M(x)$ would take exponential memory!
(Recall that x is an element of a large combinatorial space)



Representing the messages $M(x)$

Key idea: the messages $M_{j\to i}(x)$ can be shown to belong to the same exponential family the one associated to factor i

$$\exists \hat{\theta} \;\text{ s.t. }\; M_{j\to i}(x) = \exp\left\{\langle T(x), \hat{\theta}\rangle - A_i(\hat{\theta})\right\} \mathbf{1}[x \in S_i]$$

This is what gets calculated in the pseudo-code shown earlier

BPMF($\boldsymbol{\theta}, A_1, \ldots, A_I$)

1: $\zeta_{i,j}^{(1)} = 0$
2: **for** $t = 1, 2, \ldots, T$ **do**
3: $\quad \bar{\boldsymbol{\xi}}_i^{(t)} = \boldsymbol{\theta} + \sum_{i':i'\neq i} \boldsymbol{\zeta}_{i'}^{(t-1)}$
4: $\quad \boldsymbol{\zeta}_i^{(t)} = \text{logit}\left(\nabla A_i\left(\bar{\boldsymbol{\xi}}_i^{(t)}\right)\right) - \bar{\boldsymbol{\xi}}_i^{(t)}$

Experiments

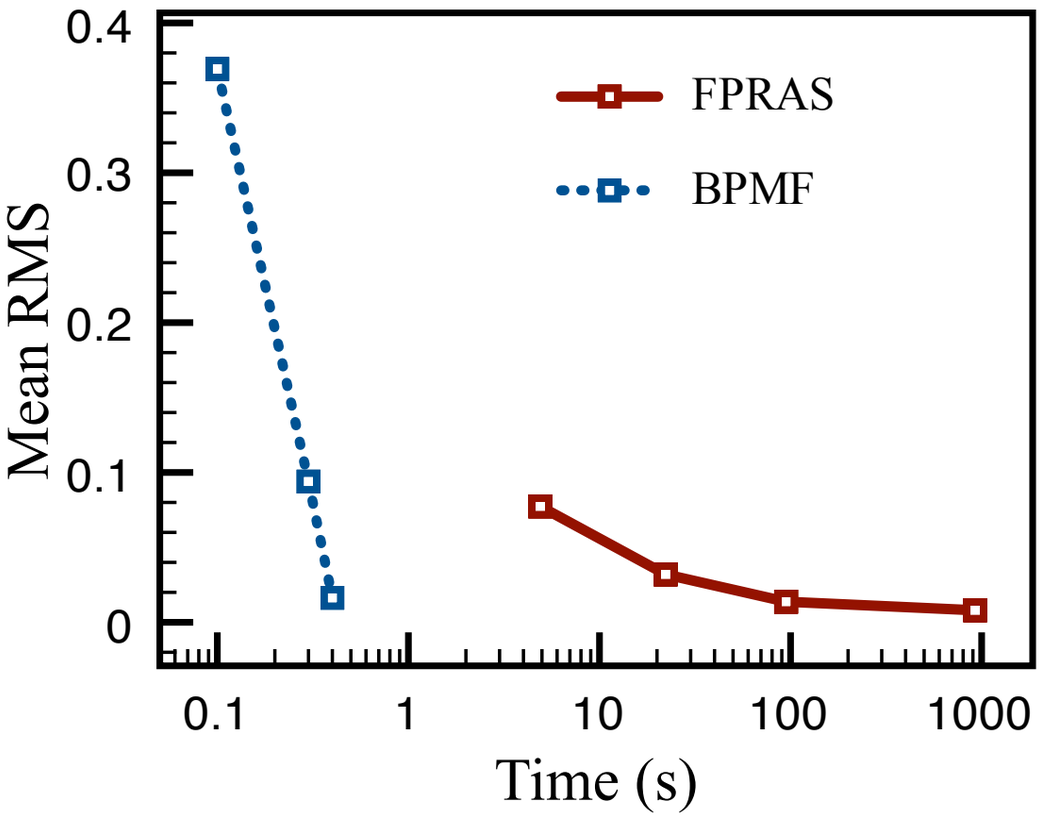
Matching: comparing to RMS to true posterior in small graphs, and reconstructing matchings from noisy observations in large graphs (averaged over 100 runs)

Synthetic data: random bipartite graphs with edge appearance probability p

Competitor: a fully polynomial randomized algorithm (FPRAS) [Rasmussen]

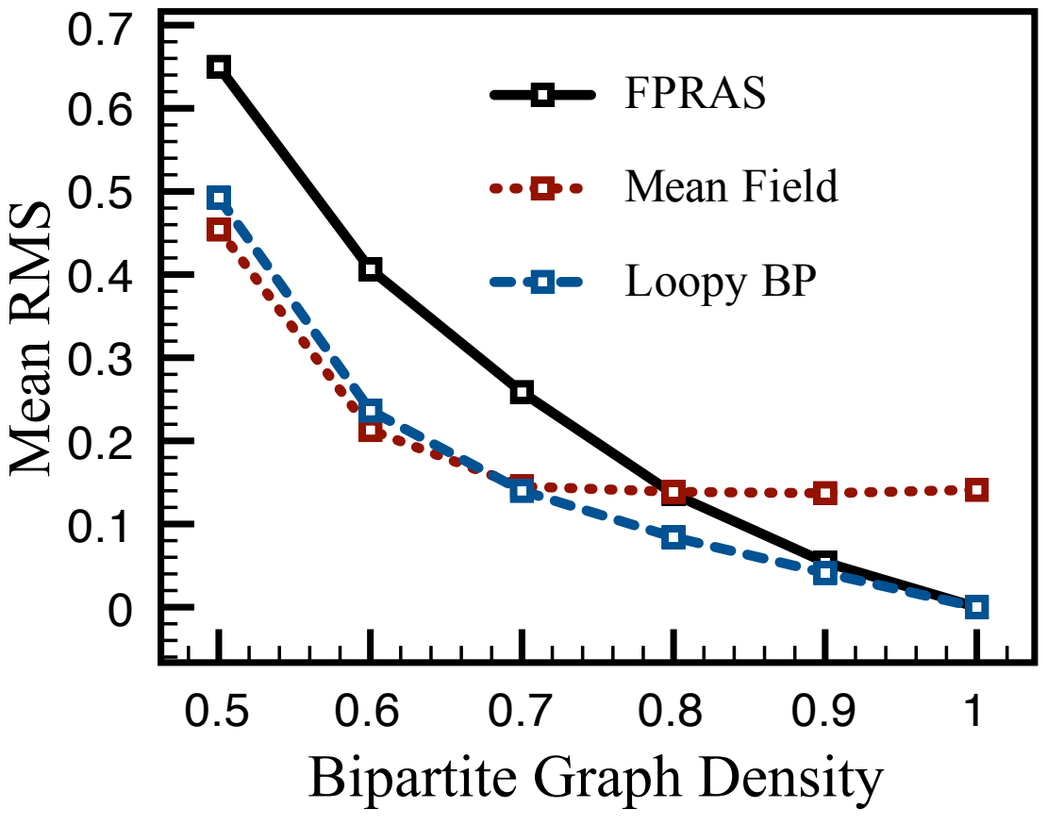
Results

Performance on p=9/10



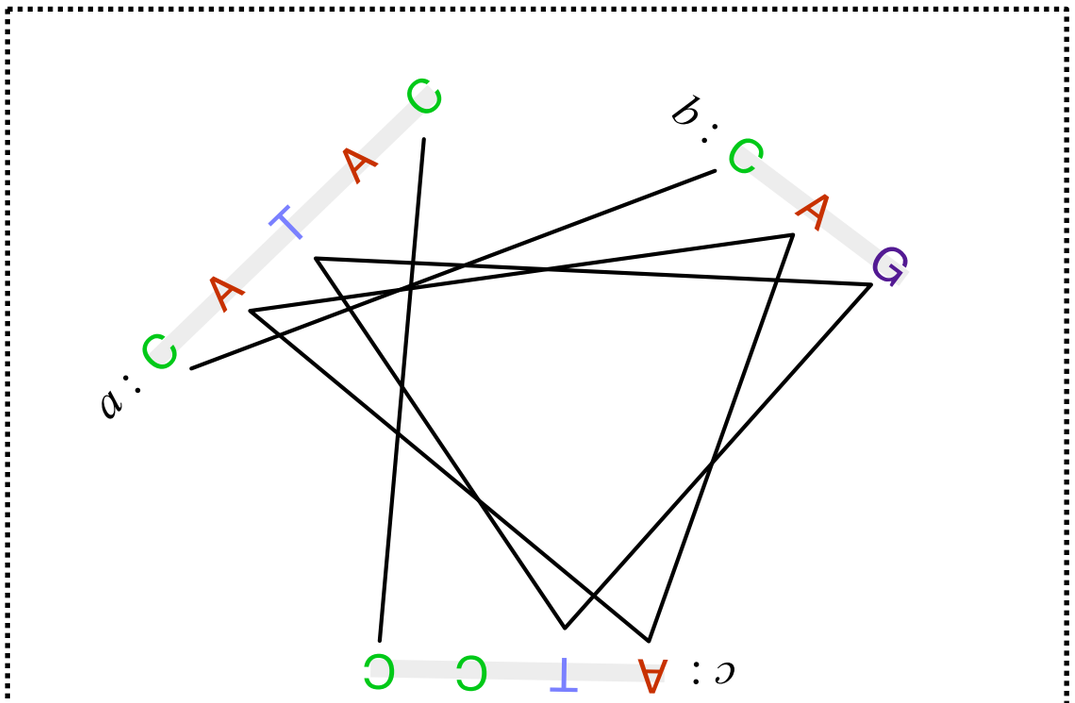
Logscale

Performance as a function of p



Application to multiple sequence alignment

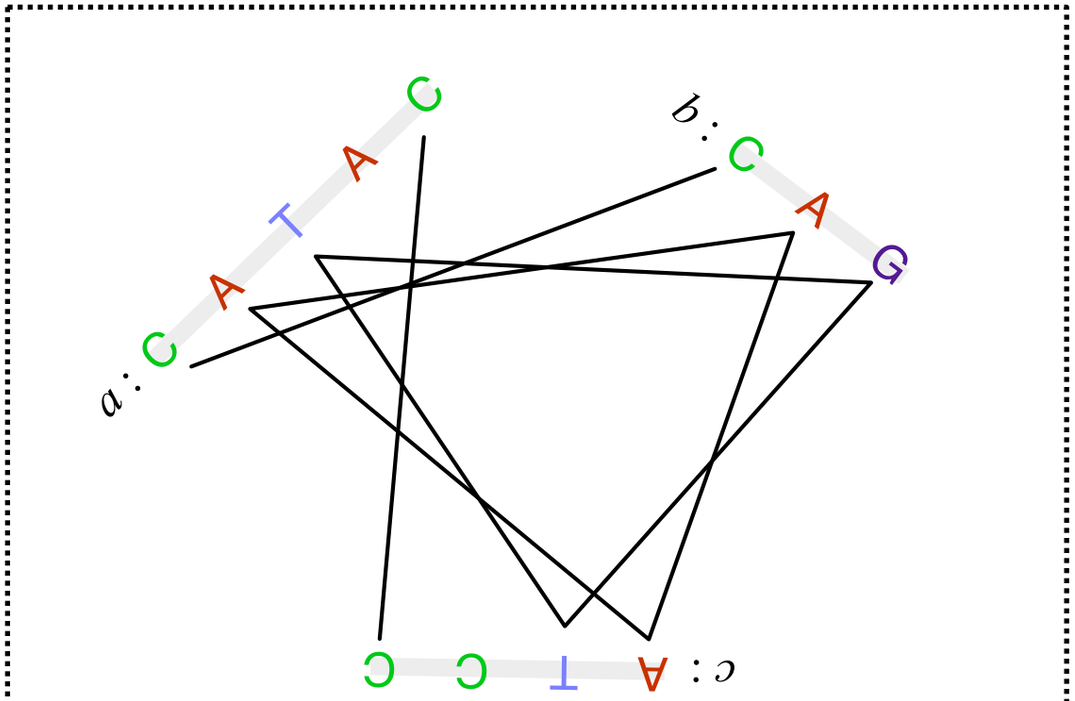
Model of alignment: k-partite transitive matchings with ordering constraints

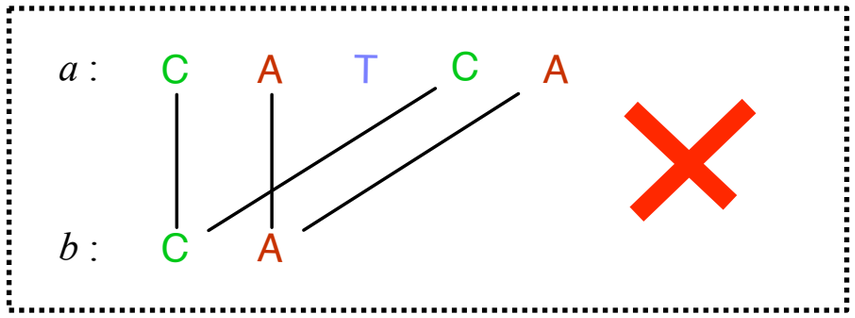


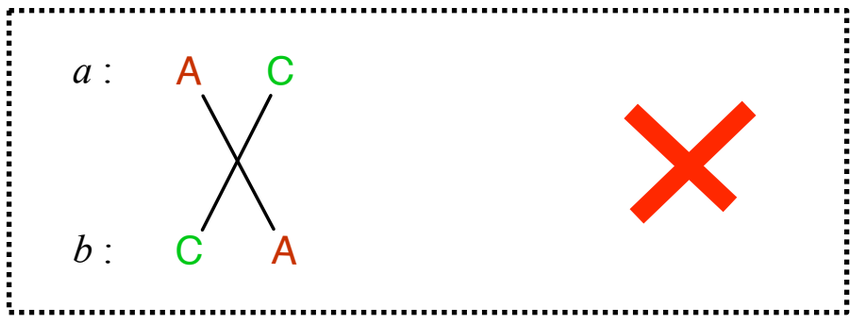
a :	C	A	-	T	A	C
b :	C	A	-	G	-	-
c :	-	A	T	C	-	C

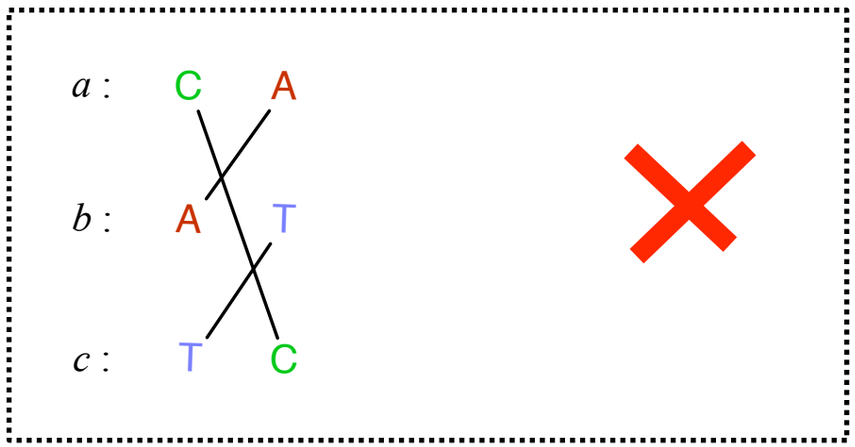
Application to multiple sequence alignment

Model of alignment: k-partite transitive matchings with ordering constraints



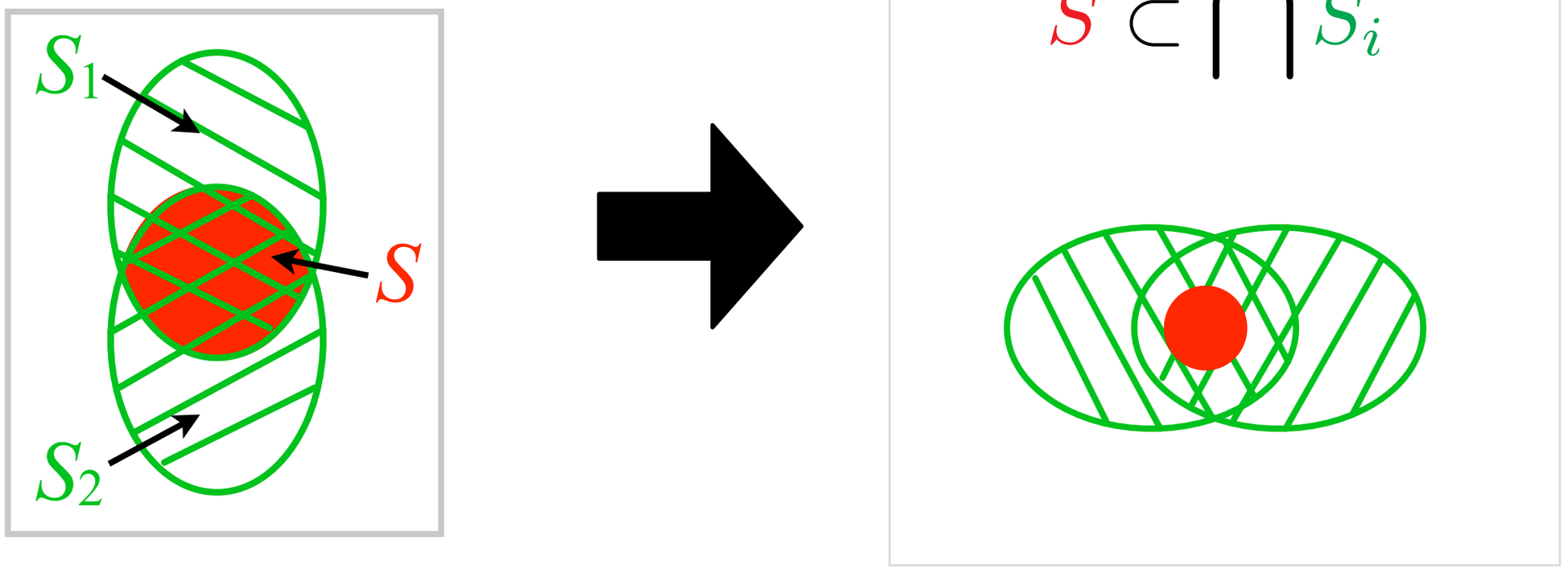






Too many factors!

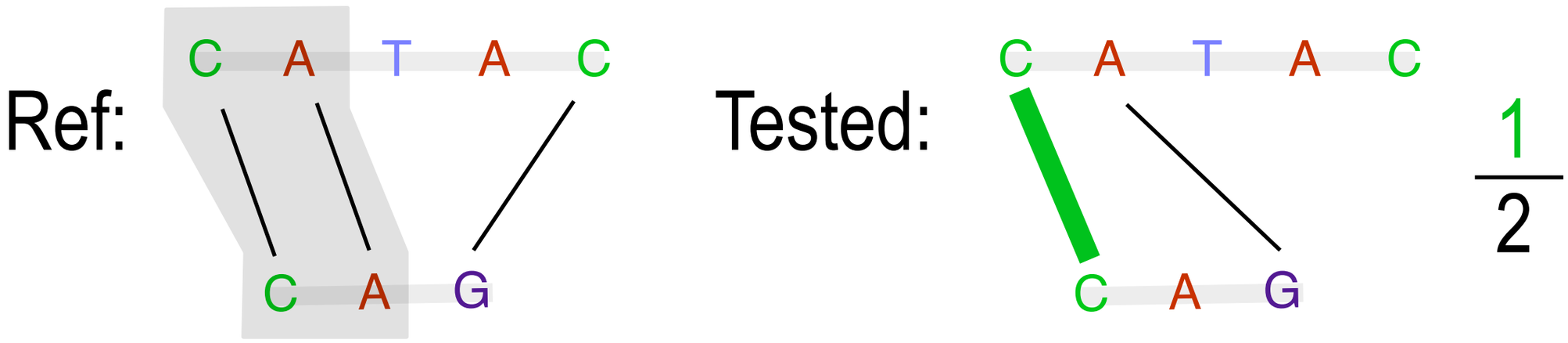
Generalized setup:



Effect on the approximation: can be characterized as an upper bound on the true partition function

Evaluation

Alignment: Sum of Pair (SP): Fraction of aligned core block residues also aligned in the reference



Dataset: BAliBASE (manually annotated proteins alignments)

[Thomson et al., ’99]

Results

BAliBASE protein group	Sum of Pairs score (SP)				
	BPMF-1	BPMF-2	BPMF-3	Clustal [24]	ProbCons [25]
short, $< 25\%$ identity	0.68	0.74	**0.76**	0.71	0.72
short, 20% — 40% identity	0.94	**0.95**	**0.95**	0.89	0.92
short, $> 35\%$ identity	0.97	**0.98**	**0.98**	0.97	**0.98**
All	0.88	**0.91**	**0.91**	0.88	0.89

Competitors:

- ***Clustal*** (most commonly used alignment package) [Higgins et al, '88]
- ***Probcons*** (a state-of-the-art system) [Do et al, '05]

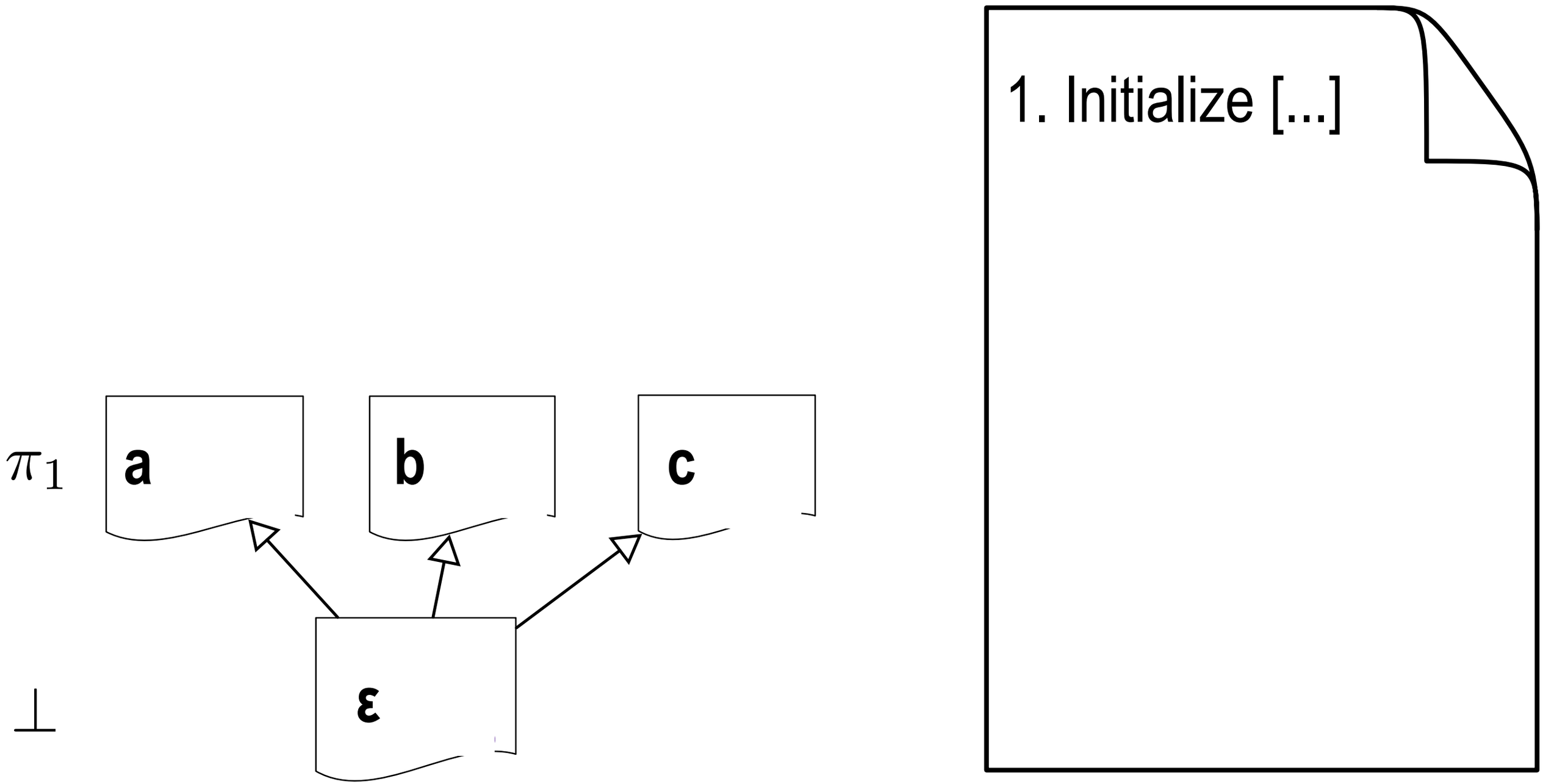
Current caveat: runs in cubic time

Outline: two frameworks for probabilistic inference over combinatorial spaces

- Variational inference by Measure Factorization
  - User's guide
  - Theoretical foundations
  - Experiments on multiple sequence alignment
- **Poset Sequential Monte Carlo**
  - User's guide
  - Theoretical foundations
  - Experiments on phylogenetic tree inference

Background: Sequential Monte Carlo (SMC)

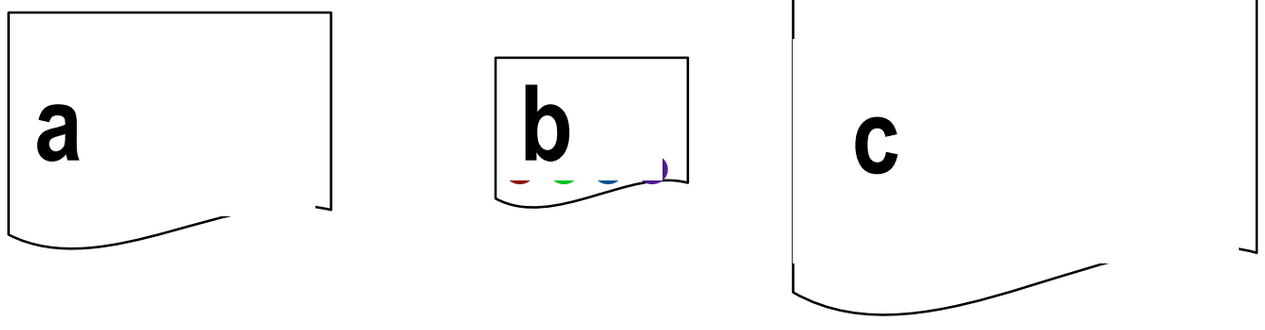
Standard SMC: An algorithm to sample sequences of length n



Sequential Monte Carlo (SMC)

π_2

π_1



SMC : Approximation for π

1. Initialize [...]

2. Iterate :

i. Sample partial states

$$p'_i \sim \pi_1 \times q$$

Sequential Monte Carlo (SMC)

SMC : Approximation for π

1. Initialize [...]

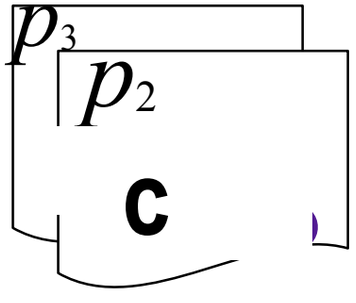
2. Iterate :

i. Sample partial states

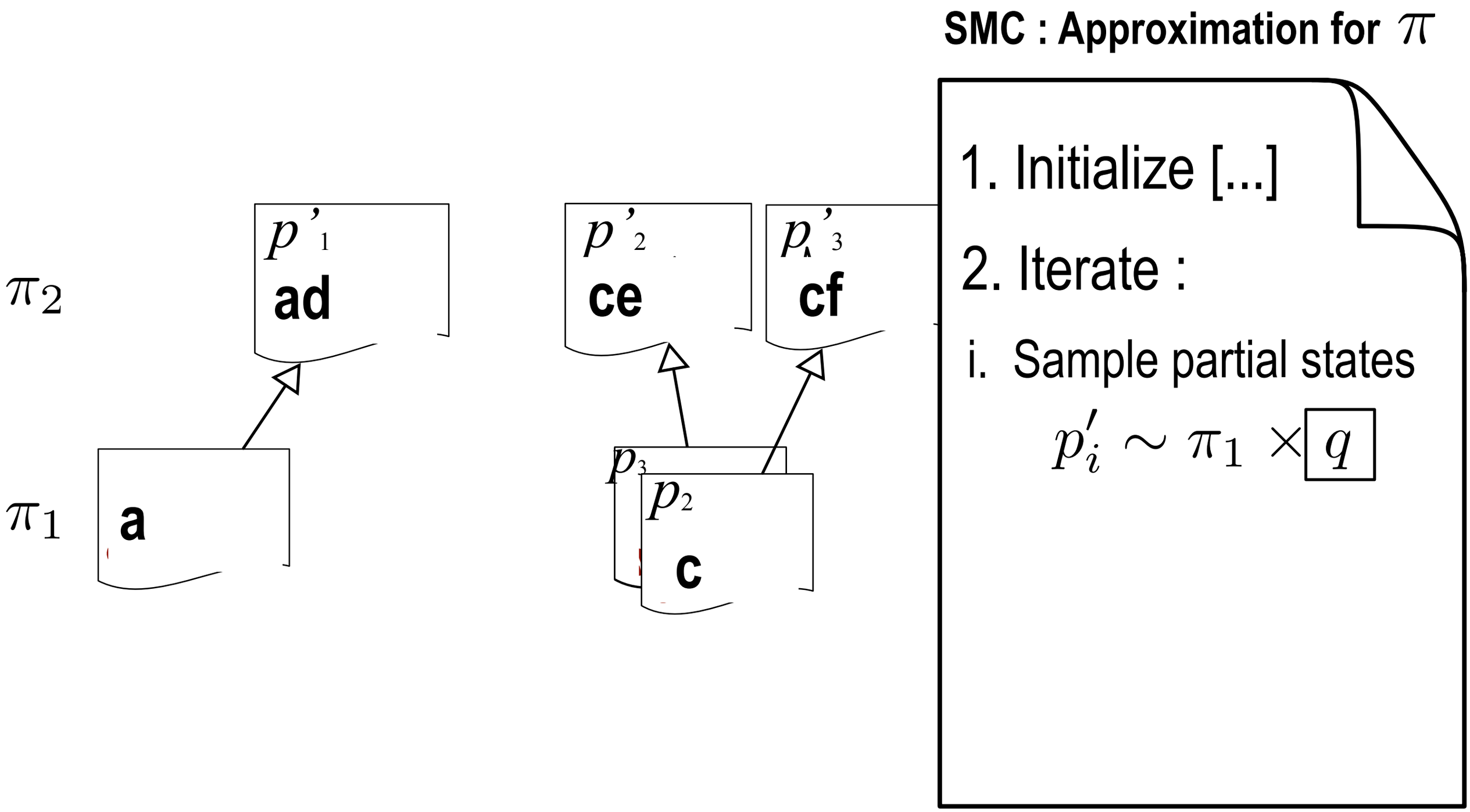
$$p'_i \sim \boxed{\pi_1} \times q$$

π_2

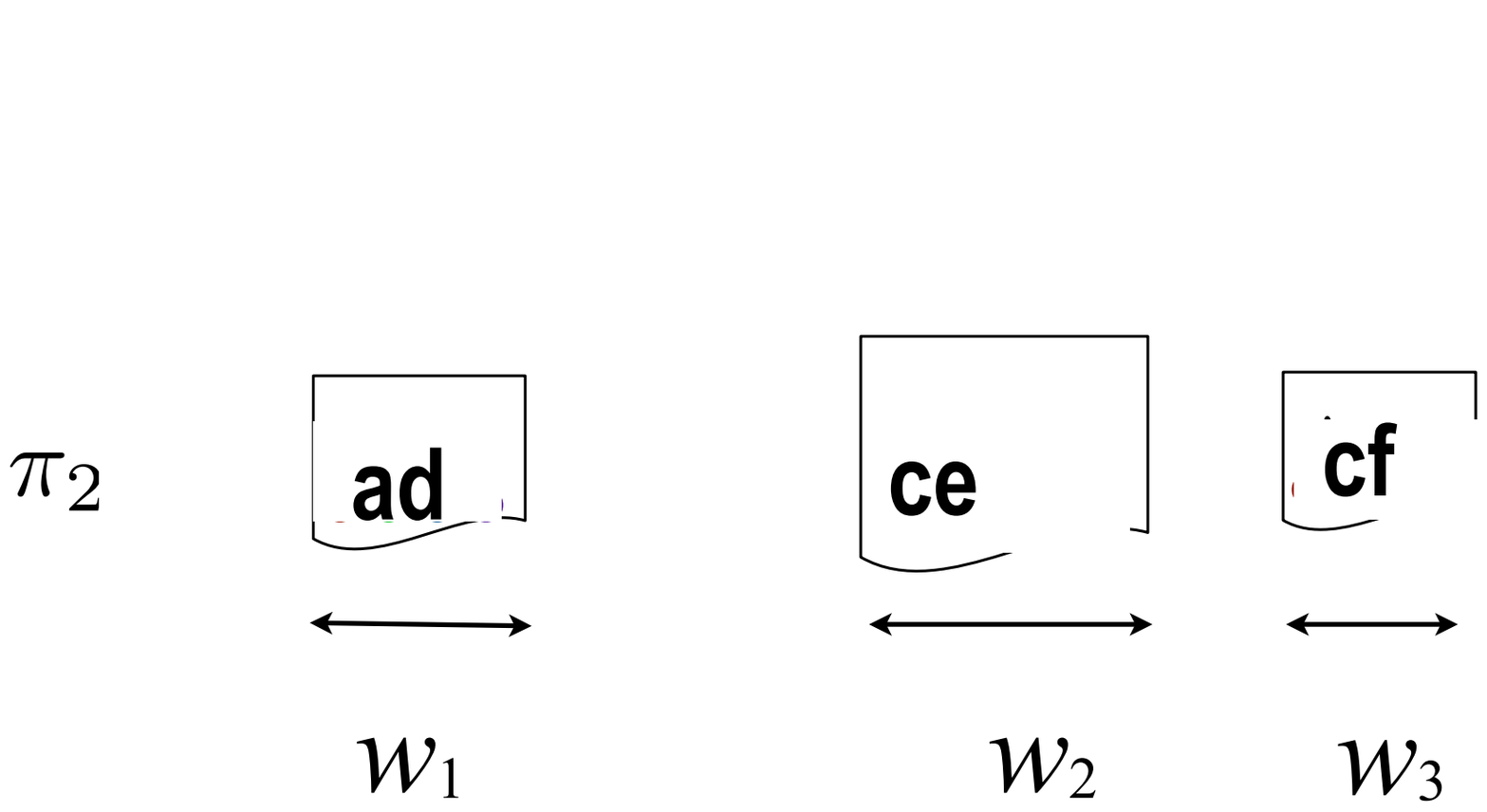
π_1



Sequential Monte Carlo (SMC)



Sequential Monte Carlo (SMC)



SMC : Approximation for π

1. Initialize [...]

2. Iterate :

i. Sample partial states

$$p'_i \sim \pi_1 \times q$$

ii. Compute weights

$$w_i = \frac{\gamma(p'_i)}{\gamma(p_i)} \frac{1}{q(p_i \rightarrow p'_i)}$$

iii. Normalize weights

Advantages of SMC

Efficiency: gives a reasonable approximation very quickly

Scale: trivial to parallelize across cores or clusters

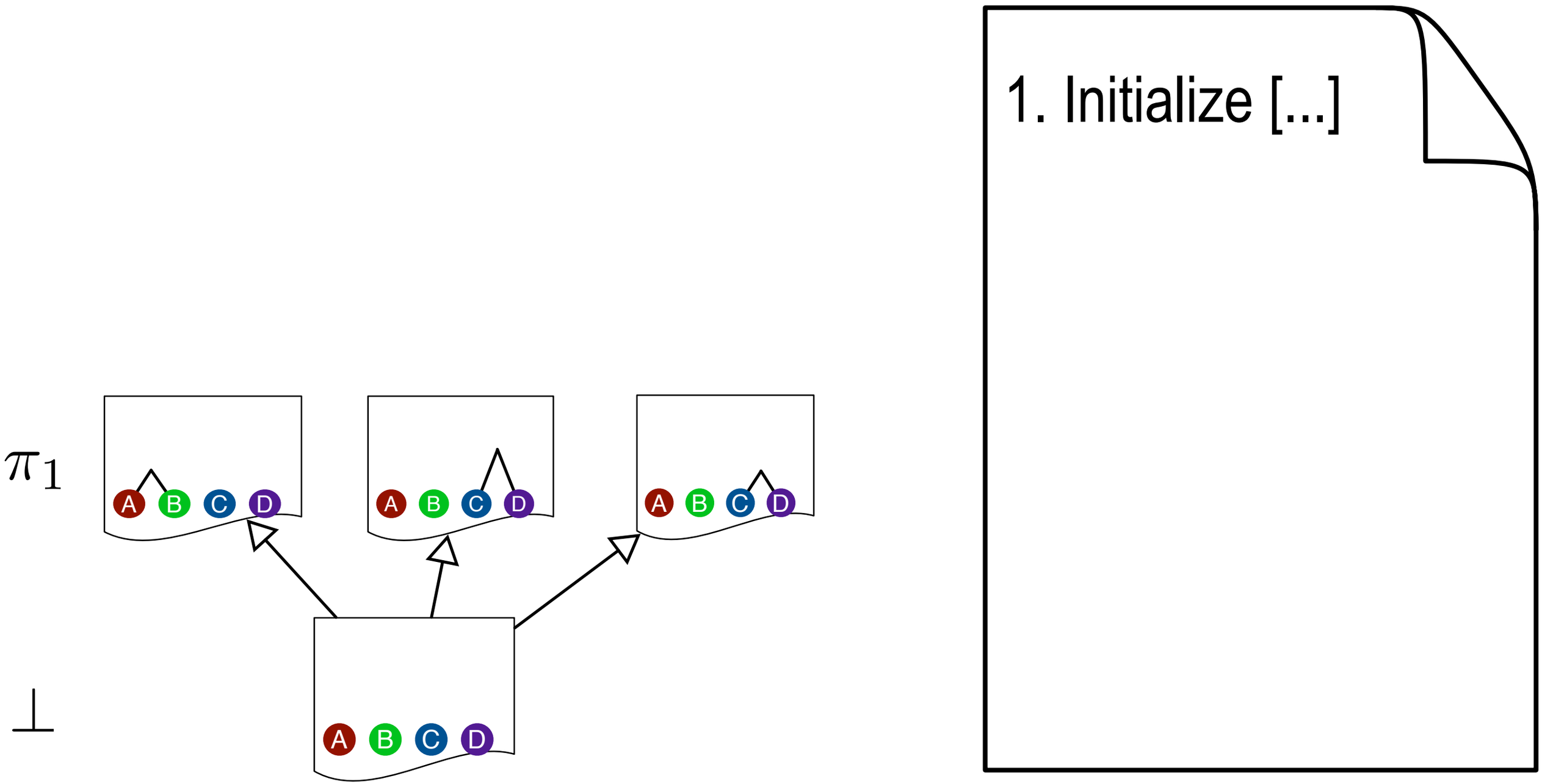
Can be combined with MCMC

Using SMC on combinatorial spaces: user's guide

Idea: sample the combinatorial structure incrementally

Example: building phylogenetic trees incrementally

Sequential Monte Carlo (SMC)



Sequential Monte Carlo (SMC)

SMC : Approximation for π

1. Initialize [...]

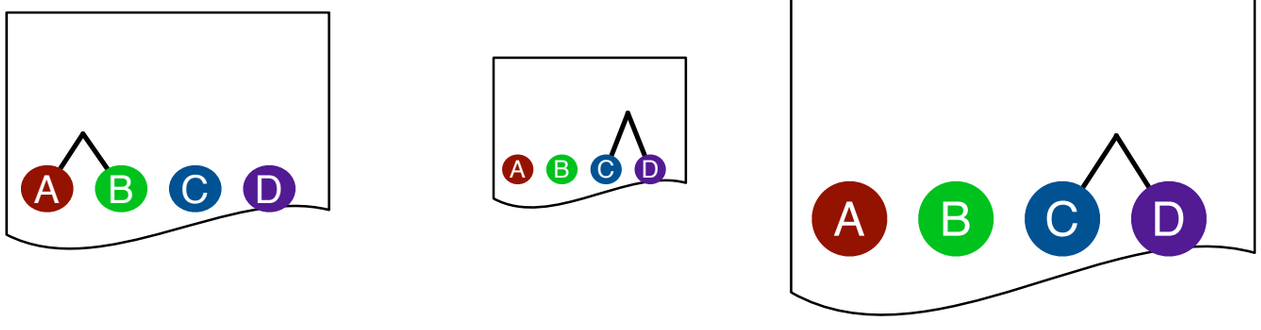
2. Iterate :

   i. Sample partial states

$$p'_i \sim \pi_1 \times q$$

π_2

π_1



Sequential Monte Carlo (SMC)

SMC : Approximation for π

1. Initialize [...]

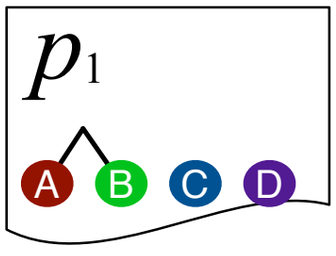
2. Iterate :

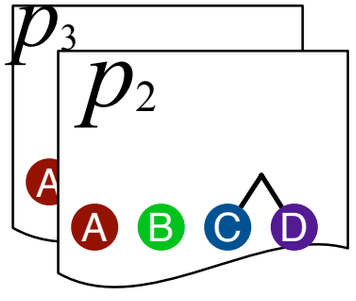
i. Sample partial states

$$p'_i \sim \boxed{\pi_1} \times q$$

π_2

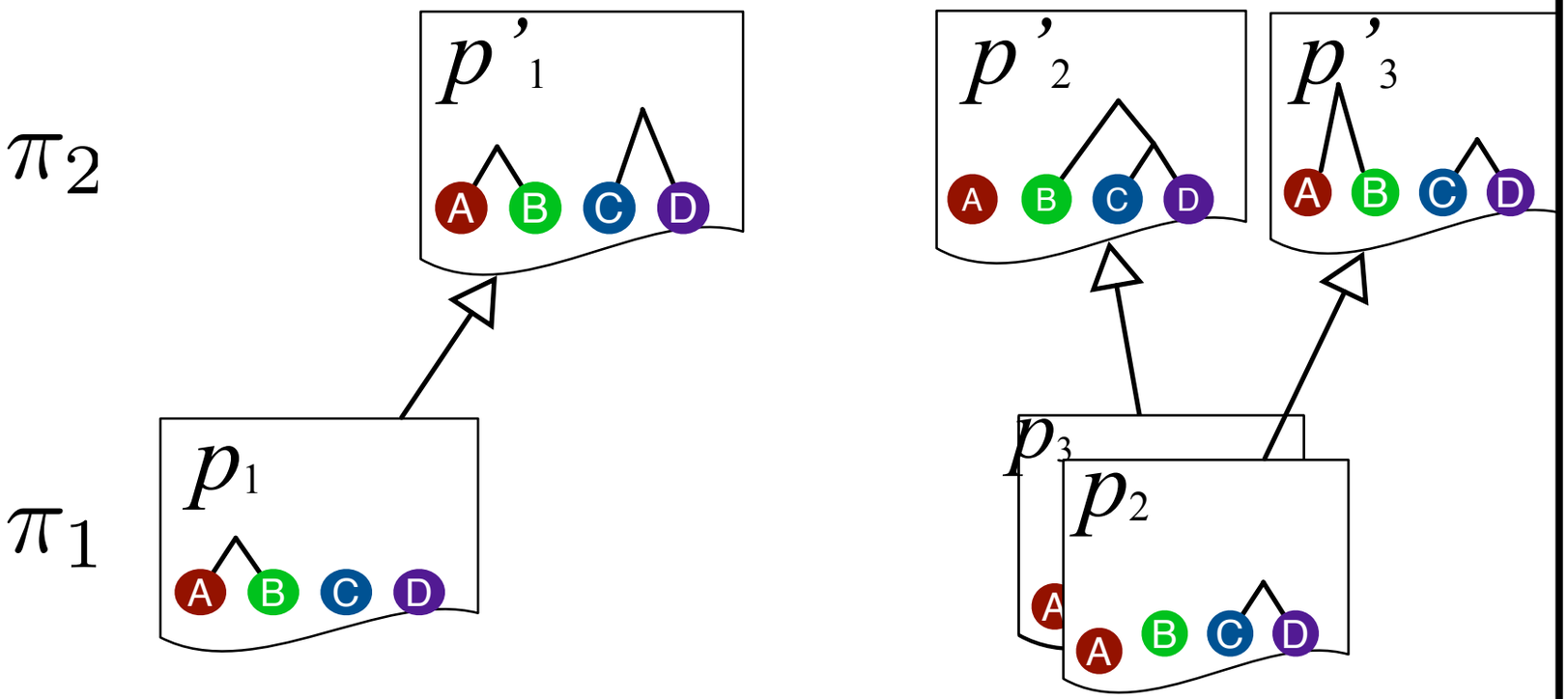
π_1





Sequential Monte Carlo (SMC)

SMC : Approximation for π



1. Initialize [...]

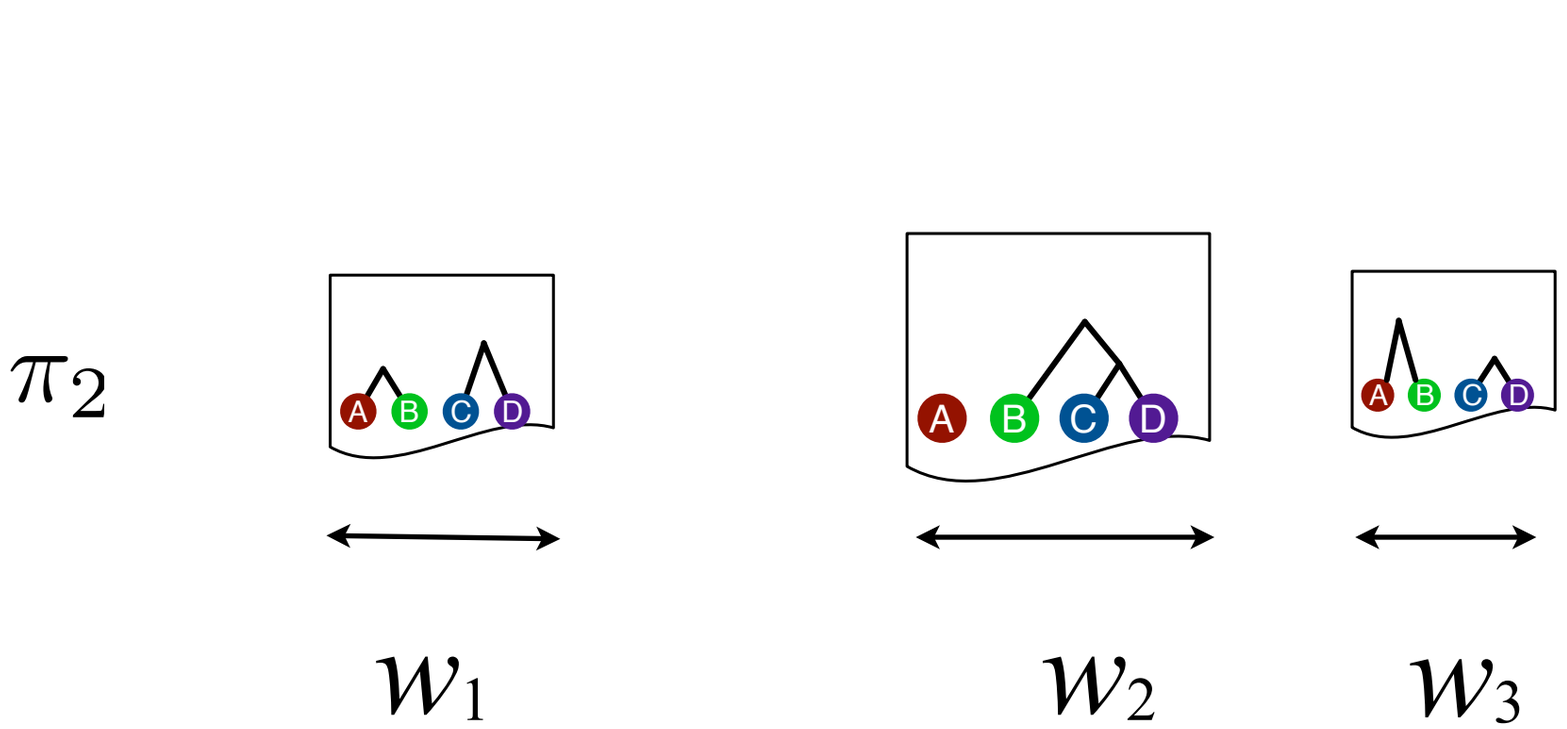
2. Iterate :

i. Sample partial states

$$p'_i \sim \pi_1 \times \boxed{q}$$

Sequential Monte Carlo (SMC)

SMC : Approximation for π



1. Initialize [...]

2. Iterate :

i. Sample partial states

$$p'_i \sim \pi_1 \times q$$

ii. Compute weights

$$w_i = \frac{\gamma(p'_i)}{\gamma(p_i)} \frac{1}{q(p_i \rightarrow p'_i)}$$

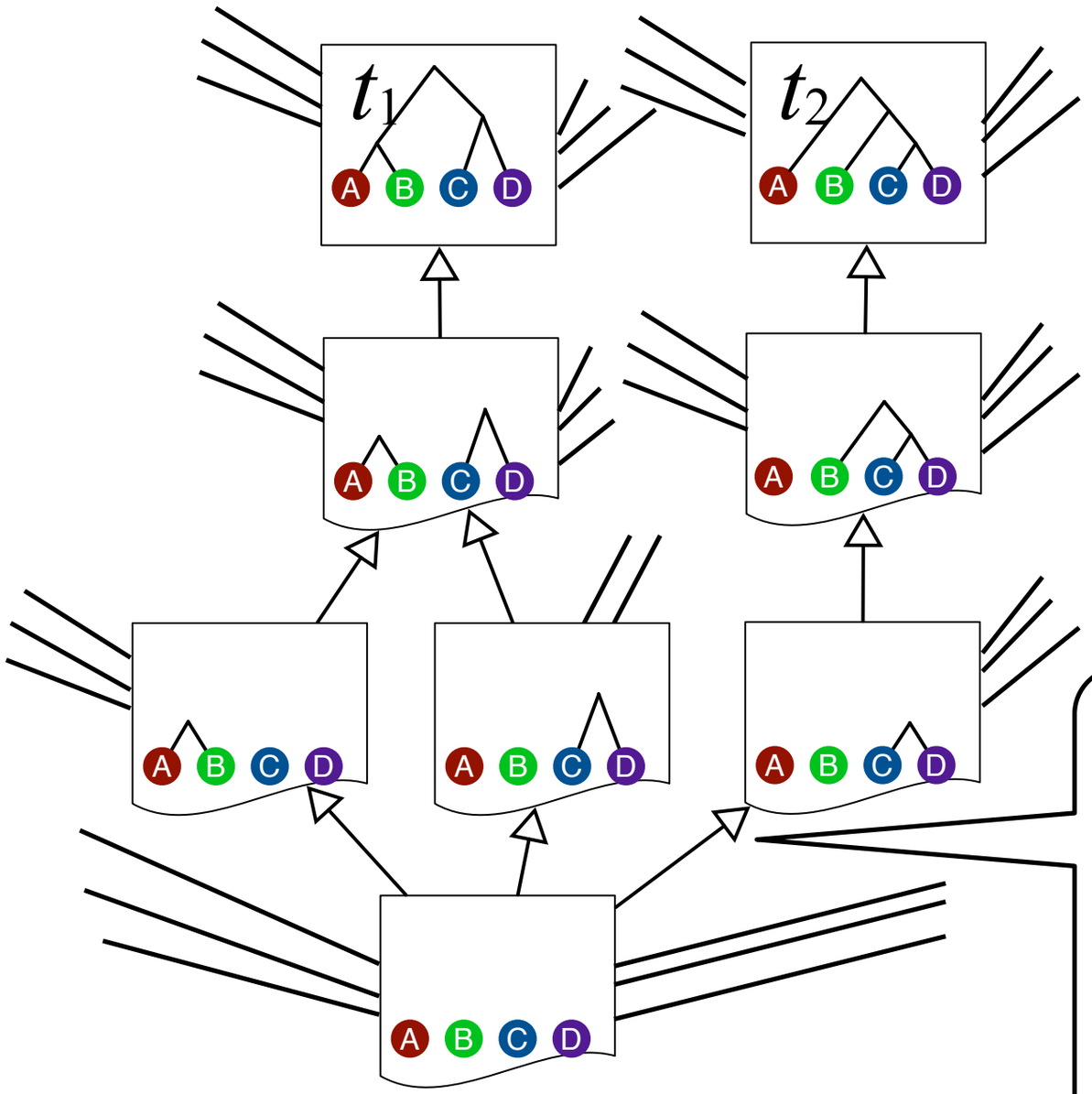
iii. Normalize weights

Is that it?

There is one issue that arise here (and was not a problem in standard SMC)

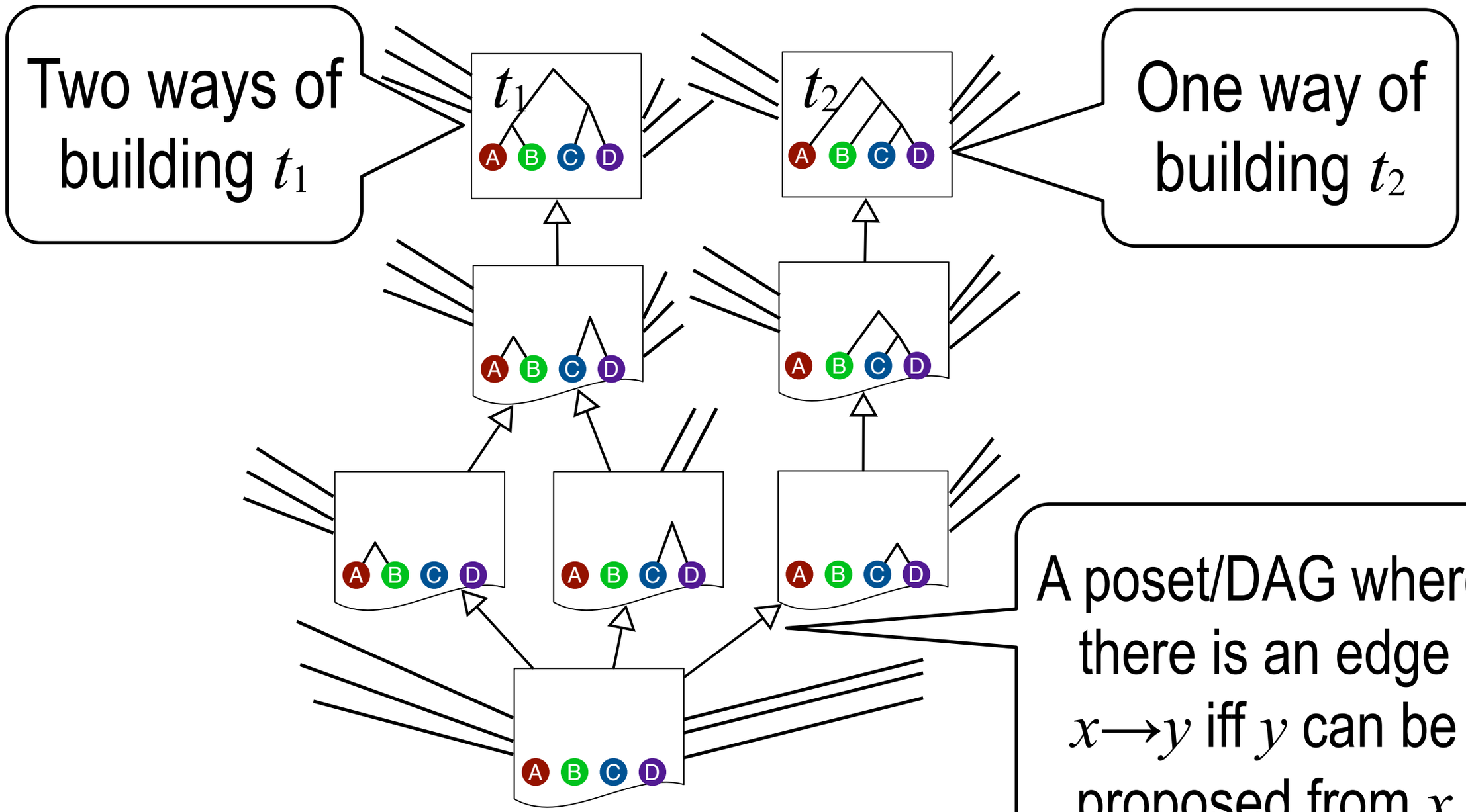
Imagine all possible particles that can be proposed...

Proposal poset for combinatorial spaces

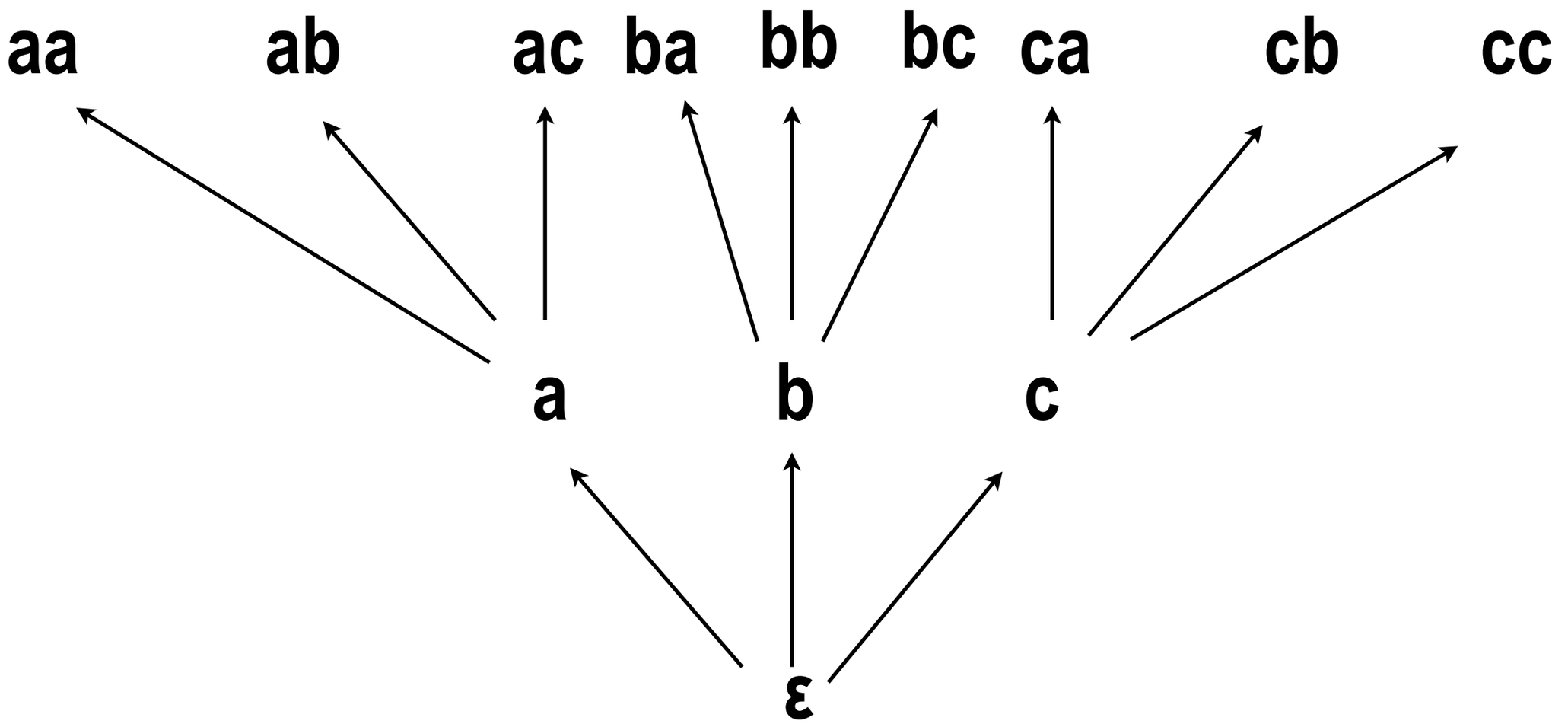


A poset/DAG where there is an edge $x \rightarrow y$ iff y can be proposed from x

Proposal poset for combinatorial spaces

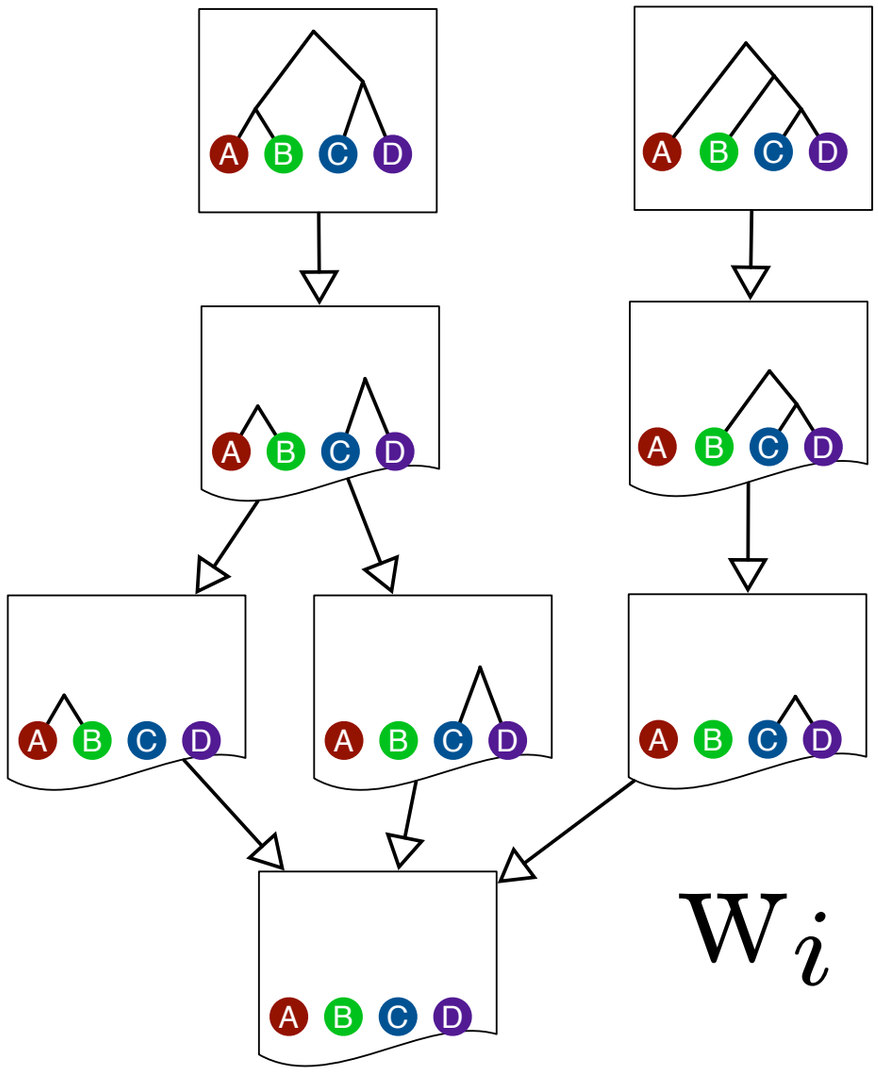


Proposal poset for standard SMC



Correcting over-counting posets

Simply define a *backward* proposal density q_- with support on the reversed proposal poset



and use it to correct the weights:

$$\mathbf{w}_i = \frac{\gamma(p_i')}{\gamma(p_i)} \frac{q_-(p_i' \rightarrow p_i)}{q(p_i \rightarrow p_i')}$$

Theoretical guarantee

Theorem:
Poset SMC is consistent, i.e. for all bounded test function f

$$\lim_{N\to\infty} \sum_{i=1}^{N} \bar{w}_i f(p_i) = \int f(p)\pi(\mathrm{d}p) \quad \text{(a.s.)}$$

Application to phylogenetic inference

Goal: comparison against MCMC

Competitor: standard MCMC sampler, 4 tempering chains, shared sum-product implementation

Metric: symmetric clade difference of the Minimum Bayes Risk reconstructed tree to the generating tree

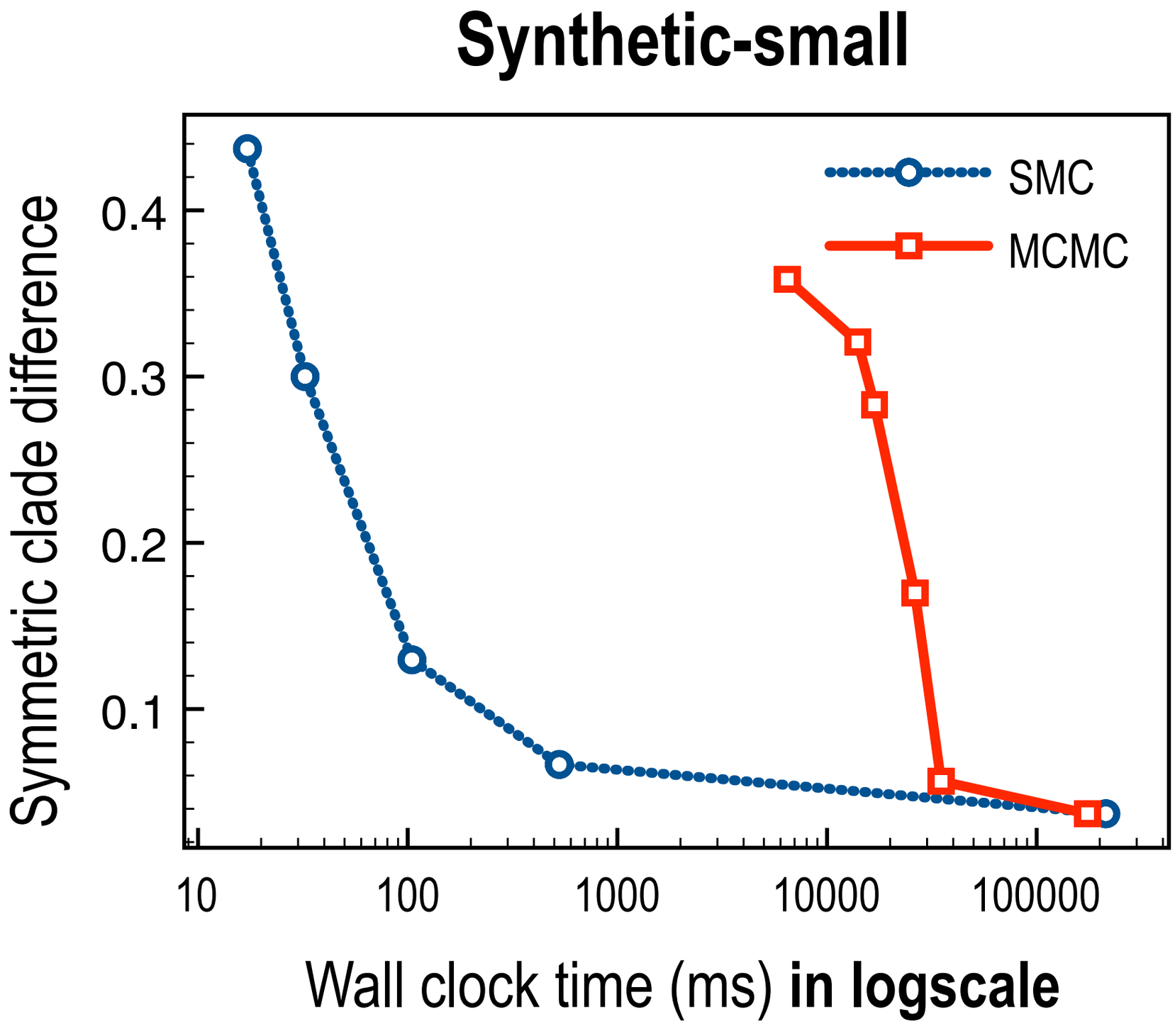
Datapoints computed by increasing the number of particles (for SMC) and the number of sampling steps (for MCMC)

Experiments: setup

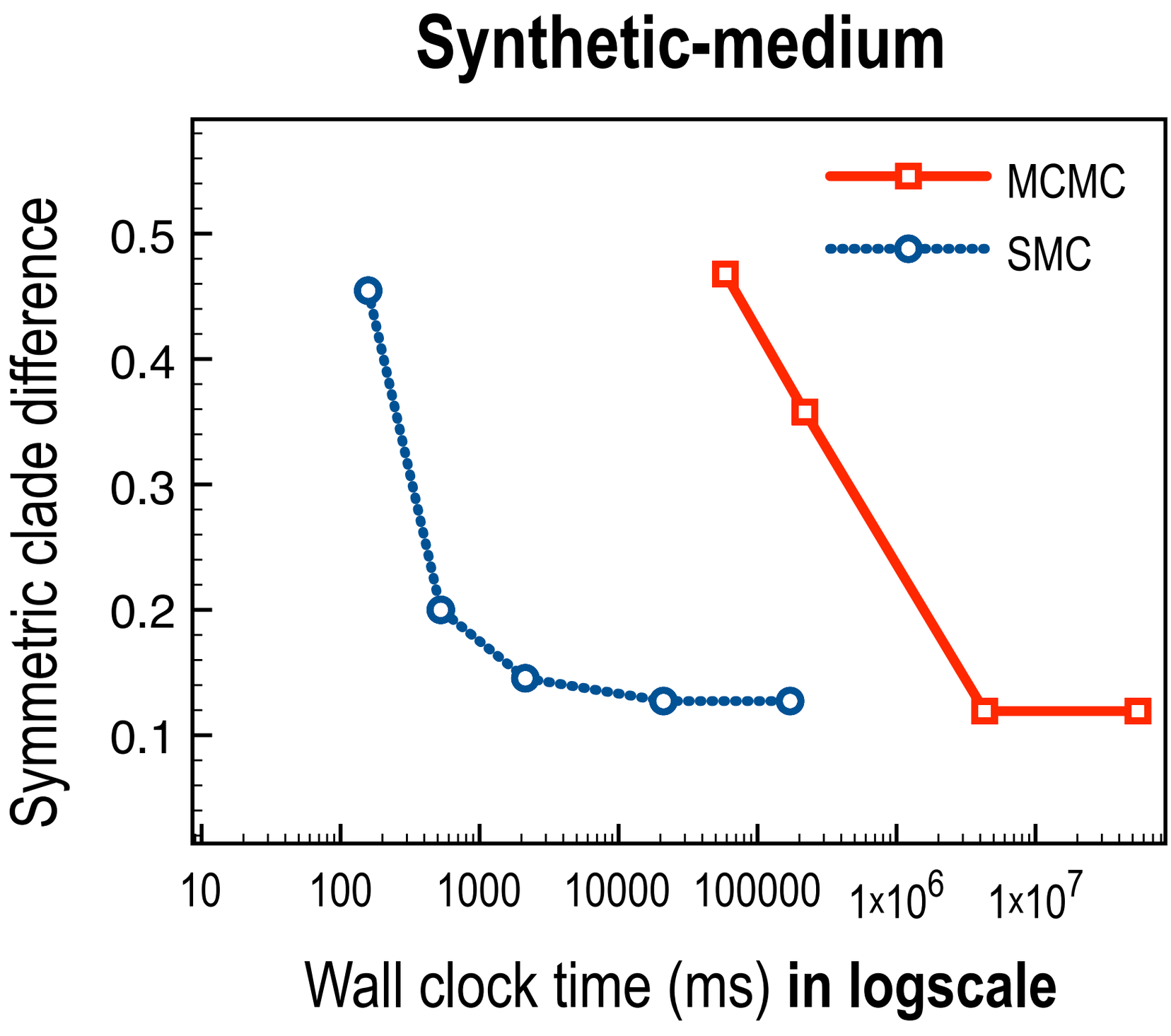
	Synthetic-small	Synthetic-med	Real data
Source	Generated from the model		Subset of HGDP
Likelihood model	Brownian motion on frequencies		
Number of sites	100		11 511
Number of nodes	25	51	25
Number of leaves	13	26	13

See Sys bio paper for more experiments

Comparison with MCMC



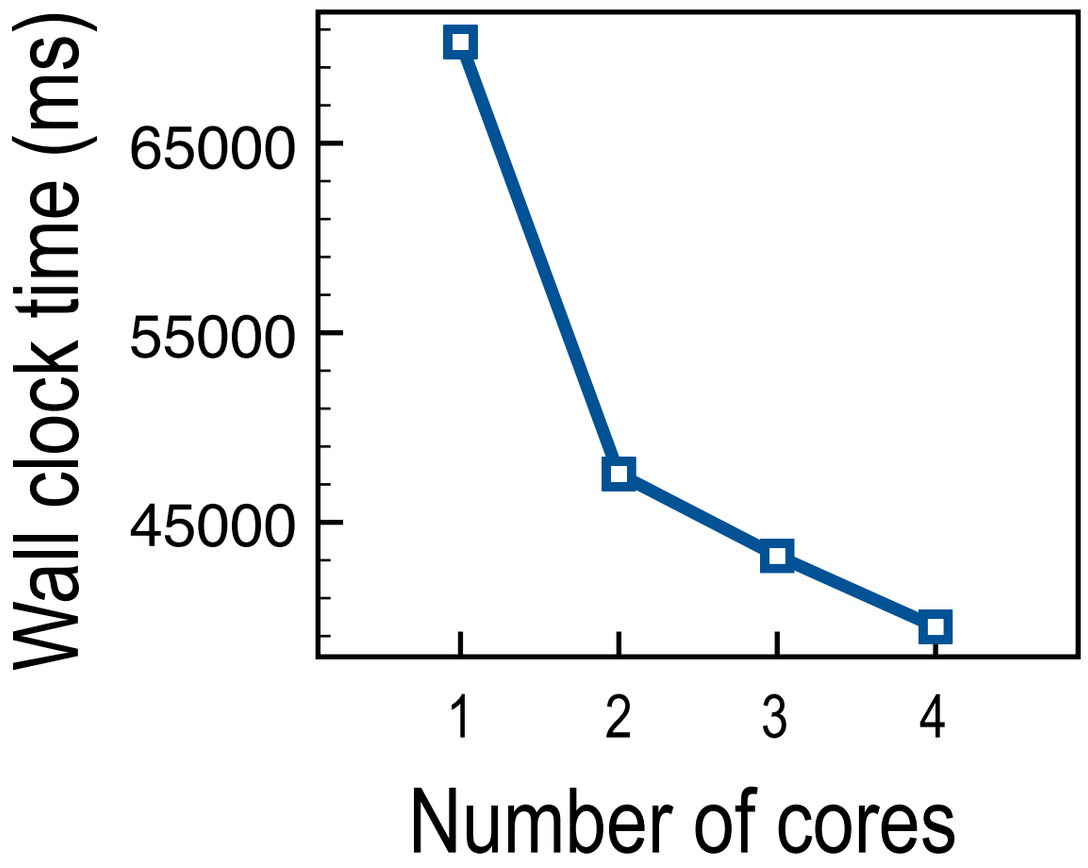
Comparison with MCMC



Experiments on real data

Goal: show that the method scales to large number of sites

Number of particle (10,000) determined using synthetic experiments, timing experiments with different numbers of cores:



Conclusion

- Variational inference by Measure Factorization
  - Cons: Approximations are not incremental (yet), MSA system is still too slow for broad usability
  - Pros: Good accuracy, bounds

[Bouchard et al. NIPS 2010]

- Poset Sequential Monte Carlo
  - Cons: Memory intensive--MCMC eventually overperforms
  - Pros: Quickly gets to a good guess, easy to parallelize, makes a great combo with MCMC

[Bouchard et al. Sys bio 2012]

Acknowledgments

- My collaborators: Liangliang Wang, Arnaud Doucet, Michael I. Jordan, Sriram Sankararaman

- Funding by NSERC, FQRNT, and the Siebel Foundation