Approximate Lifted Inference on Relational Models

Statistical Relational AI

Tutorial at BTW 2019

Tanya Braun, University of Lübeck

UNIVERSITÄT ZU LÜBECK

Thanks to Kristian Kersting

Lifted Approximate Inference

One possibility to get an approximate lifted inference approach

- Replace "conditioning" by "sampling" in recursive conditioning approaches
  [see e.g. Gogate, Jha, Venugopal NIPS'12; Venugopal, Sarkhel, Gogate AAAI'15]

Lifted Belief Propagation [Jaimovich-UAI07, Singla-AAAI08, Kersting-UAI09]

Lifted Bisimulation/Mini-buckets [Sen-VLDB08, Sen-UAI09]

Lifted Importance Sampling [Gogate-UAI11, Gogate-AAAI12]

Lifted Relax, Compensate & Recover (Generalized BP) [VdB-UAI12]

Lifted MCMC [Niepert-UAI12, Niepert-AAAI13, Venugopal-NIPS12]

Lifted Variational Inference [Choi-UAI12, Bui-StarAI12]

Lifted MAP-LP [Mladenov-AISTATS14, Apsel-AAAI14] and many more …

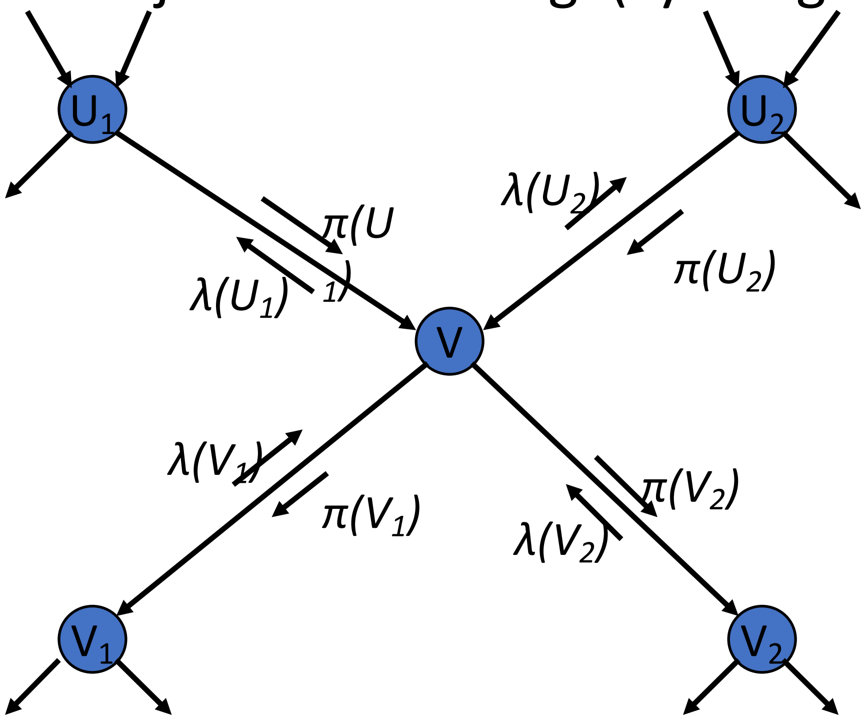
Lifted Approximate Inference

One possibility to get an approximate lifted inference approach

- Replace "conditioning" by "sampling" in recursive conditioning approaches
  [see e.g. Gogate, Jha, Venugopal NIPS'12; Venugopal, Sarkhel, Gogate AAAI'15]
- Algebraic, group-theoretical view on approximate lifted inference
  - For general understanding across different families of inference algorithms
- To do so, we start by lifting (loopy) belief propagation

A Bit of History…

- Pearl's Belief propagation [Pearl 1982]
  - Messages on Bayes net
  - Exact for polytrees (no cycles in undirected graph!)
  - Precursor of junction tree alg. (cycles go into clusters)



J. Pearl, Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach. Proc. AAAI-82. 1982.

Loopy Belief Propagation

- Pass messages on graph
  - If no cycles: exact
  - Else: approximate

- Lifted (loopy) belief propagation
  - Exploit computational symmetries
  - Compress graph whenever nodes would send identical messages
  - Send messages on compressed graph

[Singla, Domingos AAAI'08; Kersting, Ahmadi, Natarajan UAI'09; Ahmadi, Kersting, Mladenov, Natarajan MLJ'13]

Lifted Loopy Belief Propagation

- K. Kersting:

If exchanging two variables preserves optimality, group them together

Big Model

automatically compressed

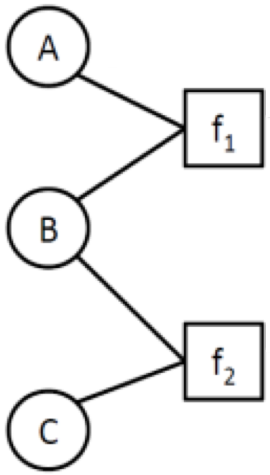
Small Model

Run
Loopy Belief Propagation

Run a modified
Loopy Belief Propagation

[Singla, Domingos AAAI'08; Kersting, Ahmadi, Natarajan UAI'09; Ahmadi, Kersting, Mladenov, Natarajan MLJ'13]

Compression: Pass the colours around*



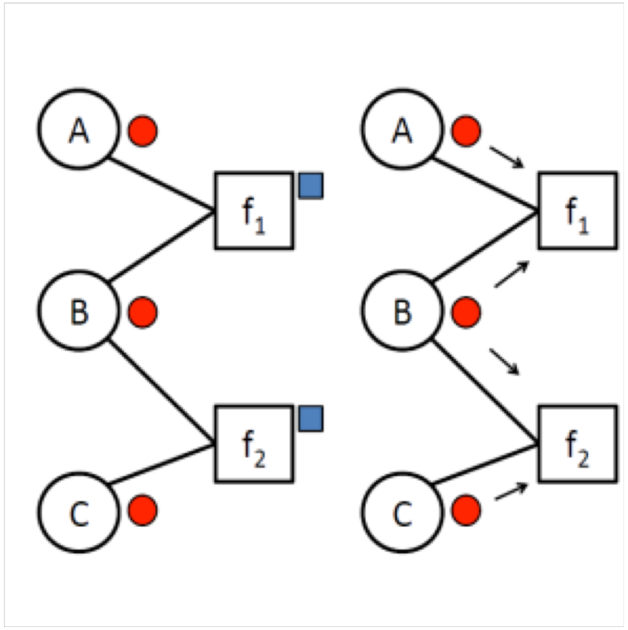
- **Colour nodes according to the evidence you have**
  - No evidence, say **red**
  - State „one", say **brown**
  - State „two", say **orange**
  - ...
- **Colour factors distinctively according to their equivalences** For instance, assuming f_1 and f_2 to be identical and B appears at the second position within both, say **blue**

*can also be done at the „lifted", i.e., relational level

[Singla, Domingos AAAI'08; Kersting, Ahmadi, Natarajan UAI'09; Ahmadi, Kersting, Mladenov, Natarajan MLJ'13]

Compression: Pass the colours around

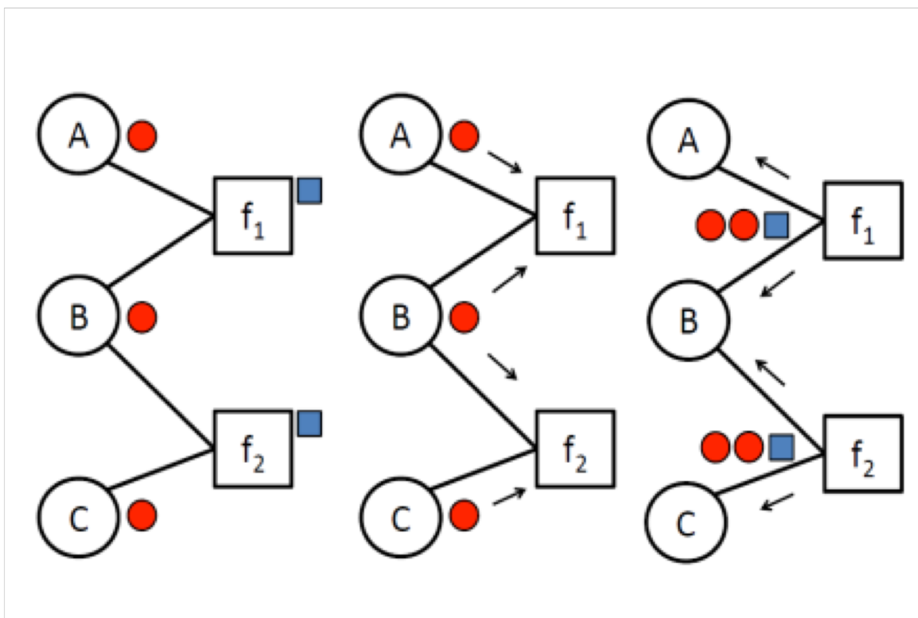
1. Each factor collects the colours of its neighbouring nodes



[Singla, Domingos AAAI'08; Kersting, Ahmadi, Natarajan UAI'09; Ahmadi, Kersting, Mladenov, Natarajan MLJ'13]

Compression: Pass the colours around

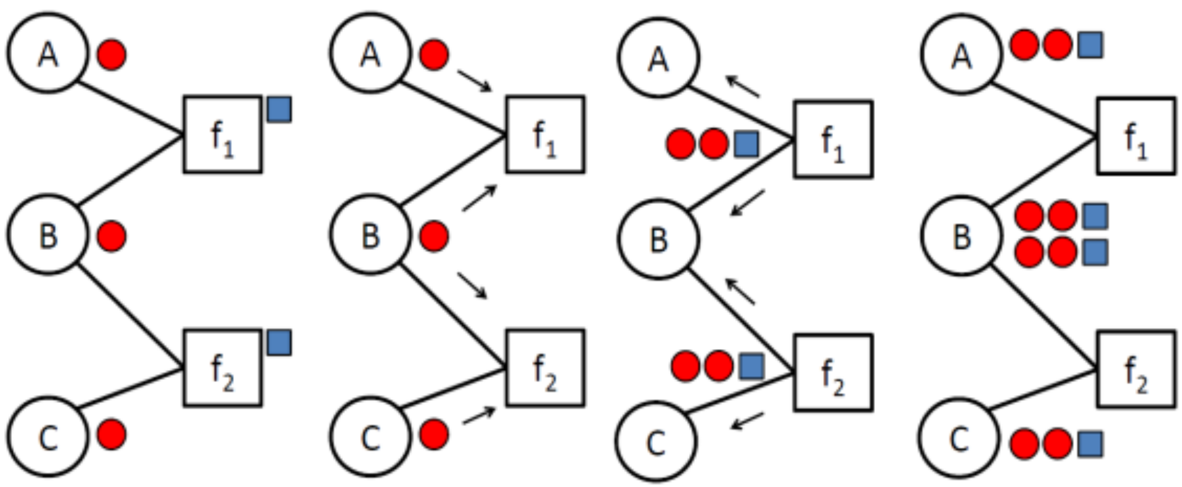
1. Each factor collects the colours of its neighbouring nodes
2. Each factor „signs“ its colour signature with its own colour



[Singla, Domingos AAAI’08; Kersting, Ahmadi, Natarajan UAI’09; Ahmadi, Kersting, Mladenov, Natarajan MLJ’13]

Compression: Pass the colours around

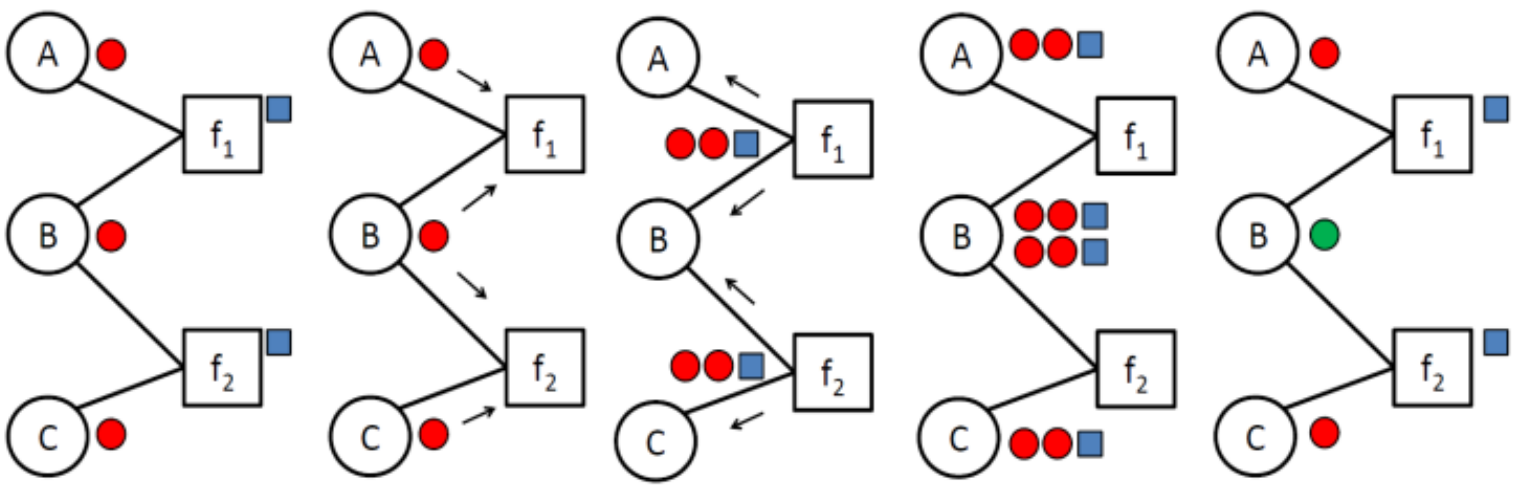
1. Each factor collects the colours of its neighbouring nodes
2. Each factor „signs“ its colour signature with its own colour
3. Each node collects the signatures of its neighbouring factors



[Singla, Domingos AAAI’08; Kersting, Ahmadi, Natarajan UAI’09; Ahmadi, Kersting, Mladenov, Natarajan MLJ’13]

Compression: Pass the colours around

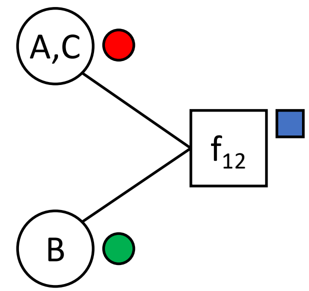
1. Each factor collects the colours of its neighbouring nodes
2. Each factor „signs“ its colour signature with its own colour
3. Each node collects the signatures of its neighbouring factors
4. Nodes are recoloured according to the collected signatures

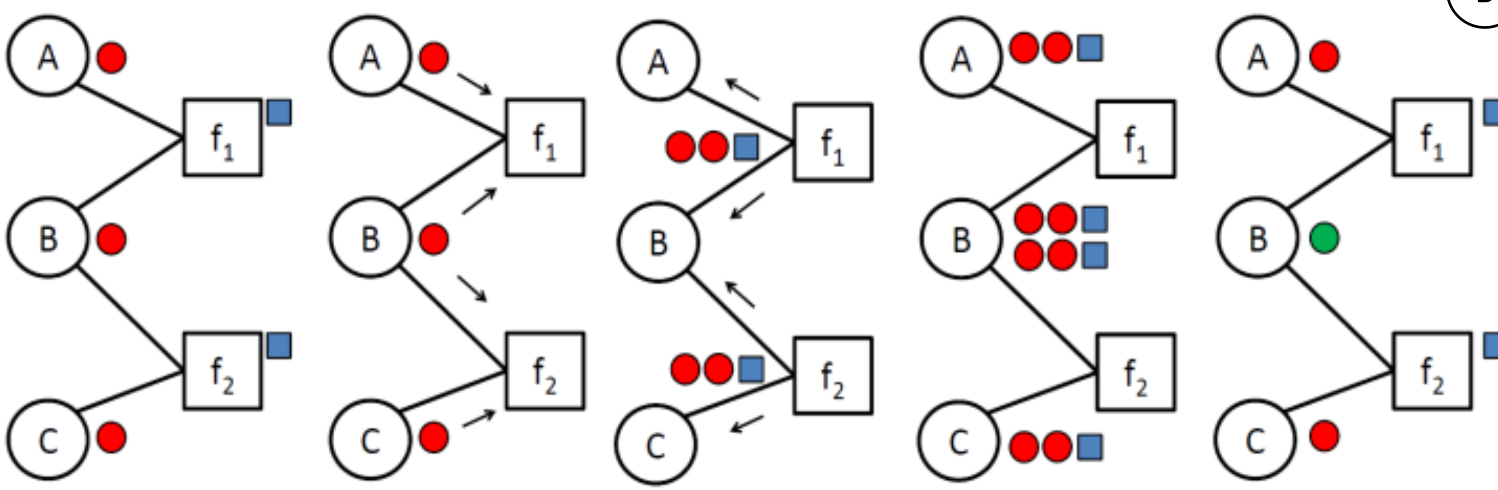


[Singla, Domingos AAAI’08; Kersting, Ahmadi, Natarajan UAI’09; Ahmadi, Kersting, Mladenov, Natarajan MLJ’13]

Compression: Pass the colours around

1. Each factor collects the colours of its neighbouring nodes
2. Each factor „signs“ its colour signature with its own colour
3. Each node collects the signatures of its neighbouring factors
4. Nodes are recoloured according to the collected signatures
5. If no new colour is created stop, otherwise go back to 1





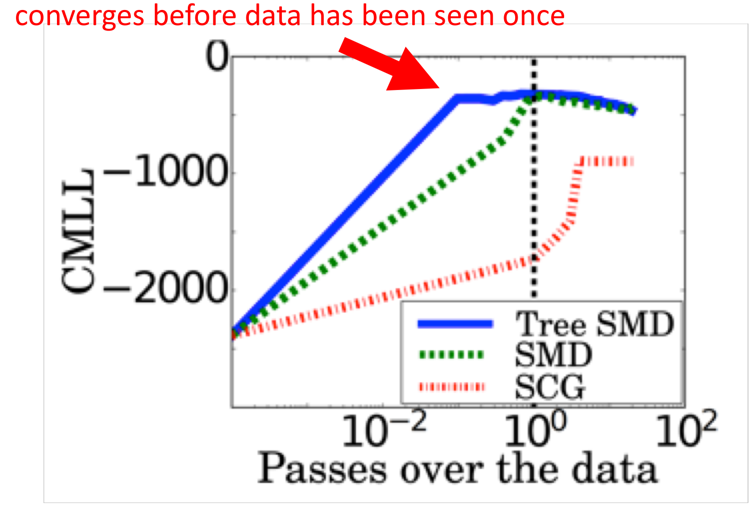
[Singla, Domingos AAAI’08; Kersting, Ahmadi, Natarajan UAI’09; Ahmadi, Kersting, Mladenov, Natarajan MLJ’13]

Considerable Speedup

- Probabilistic inference

Domain	Time (in seconds)						No. of (Super) Features	
	Construction		BP		Total			
	Ground	Lifted	Ground	Lifted	Ground	Lifted	Ground	Lifted
Cora	263.1	1173.3	12368.4	3997.7	12631.6	5171.1	2078629	295468
UW-CSE	6.9	22.1	1015.8	602.5	1022.8	624.7	217665	86459
Friends & Smokers	38.8	89.7	10702.2	4.4	10741.0	94.2	1900905	58

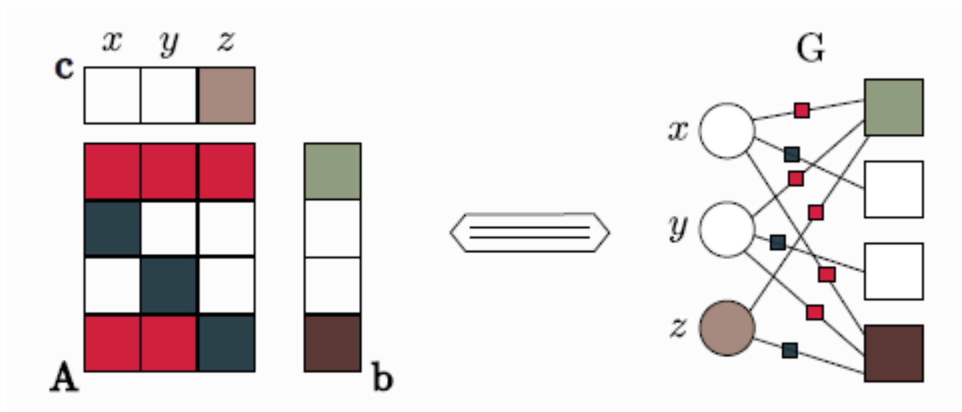
- Parameter training using a lifted stochastic gradient
    - CORA entity resolution



Colour Passing in Graph Theory

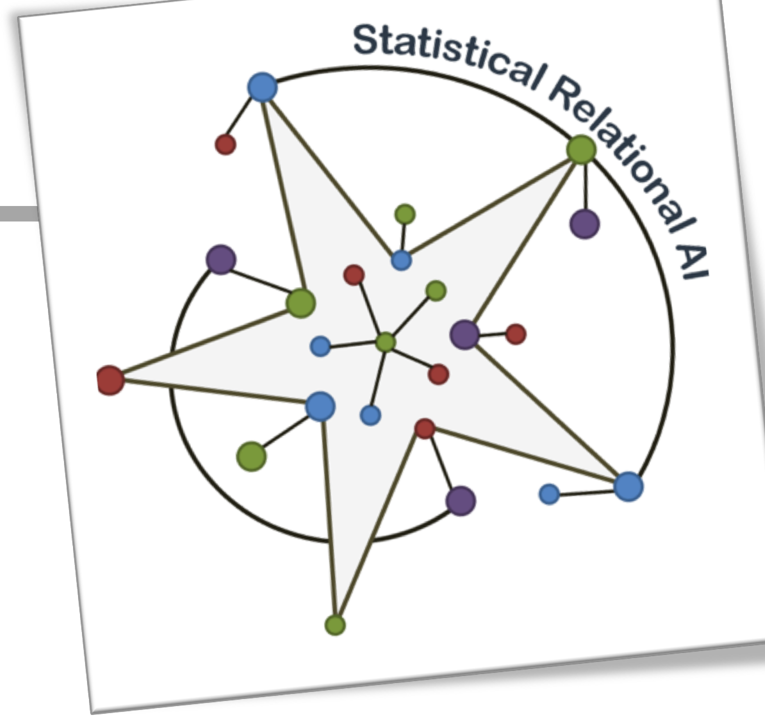
$$\max_{[x,y,z]^T \in \mathbb{R}^3} \quad 0x + 0y + 1z$$

$$\text{s.t.} \quad \begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \leq \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$



- Weisfeiler-Lehman Algorithm ≙ Colour Passing
  - Computes (fractional) automorphisms of mathematical programs
- Lifted Mathematical Programming
  - View mathematical program as coloured graph
  - Reduce program by running Weisfeiler-Lehman on graph
  - Solve reduced program using any solver

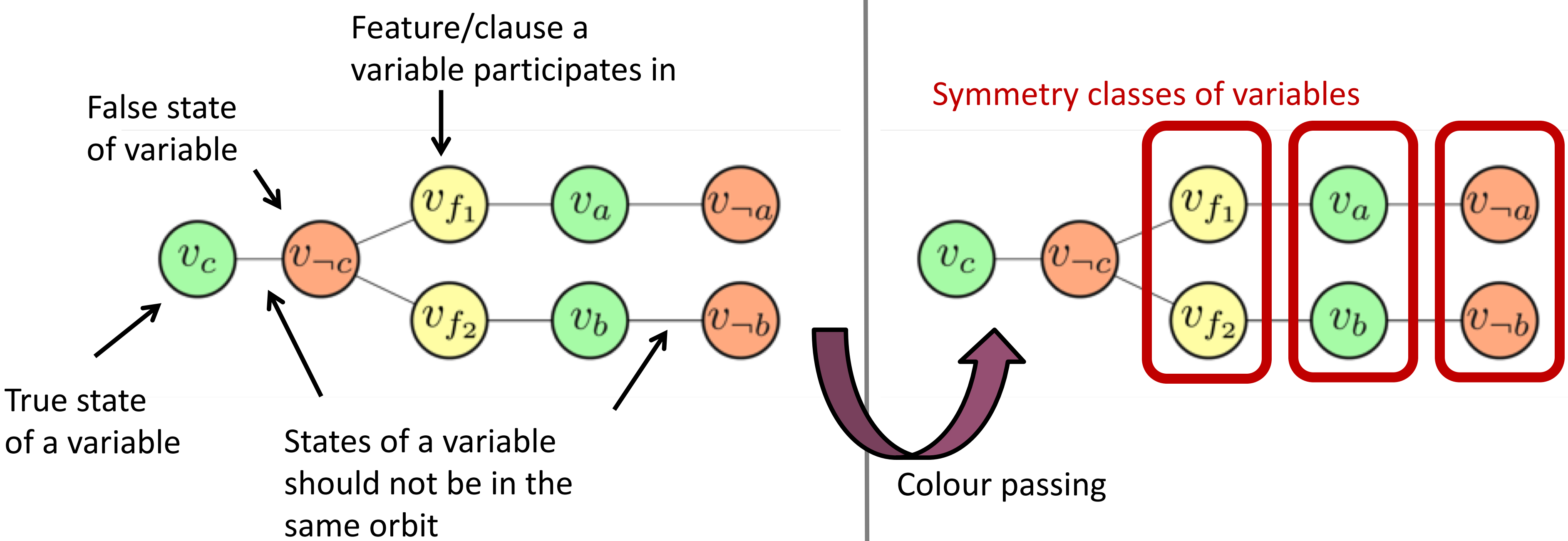
[Mladenov, Ahmadi, Kersting AISTATS ´12, Grohe, Kersting, Mladenov, Selman ESA ´14, Mladenov, Globerson, Kersting UAI ´14, AISTATS ´14, Mladenov, Kersting UAI ´15, Kersting, Mladenov, Tokmatov AIJ ´17]



Symmetries can also be exploited to speed up sampling

Orbital Markov Chain Monte Carlo

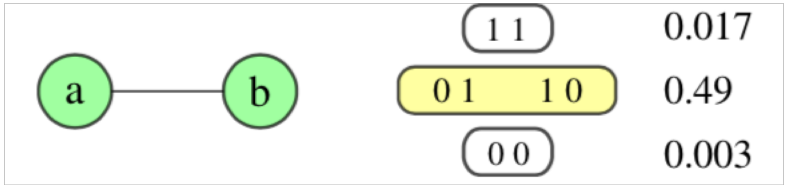
true and false states have the same color, and all clauses/features that have the same weight



Jump between symmetric states uniformly

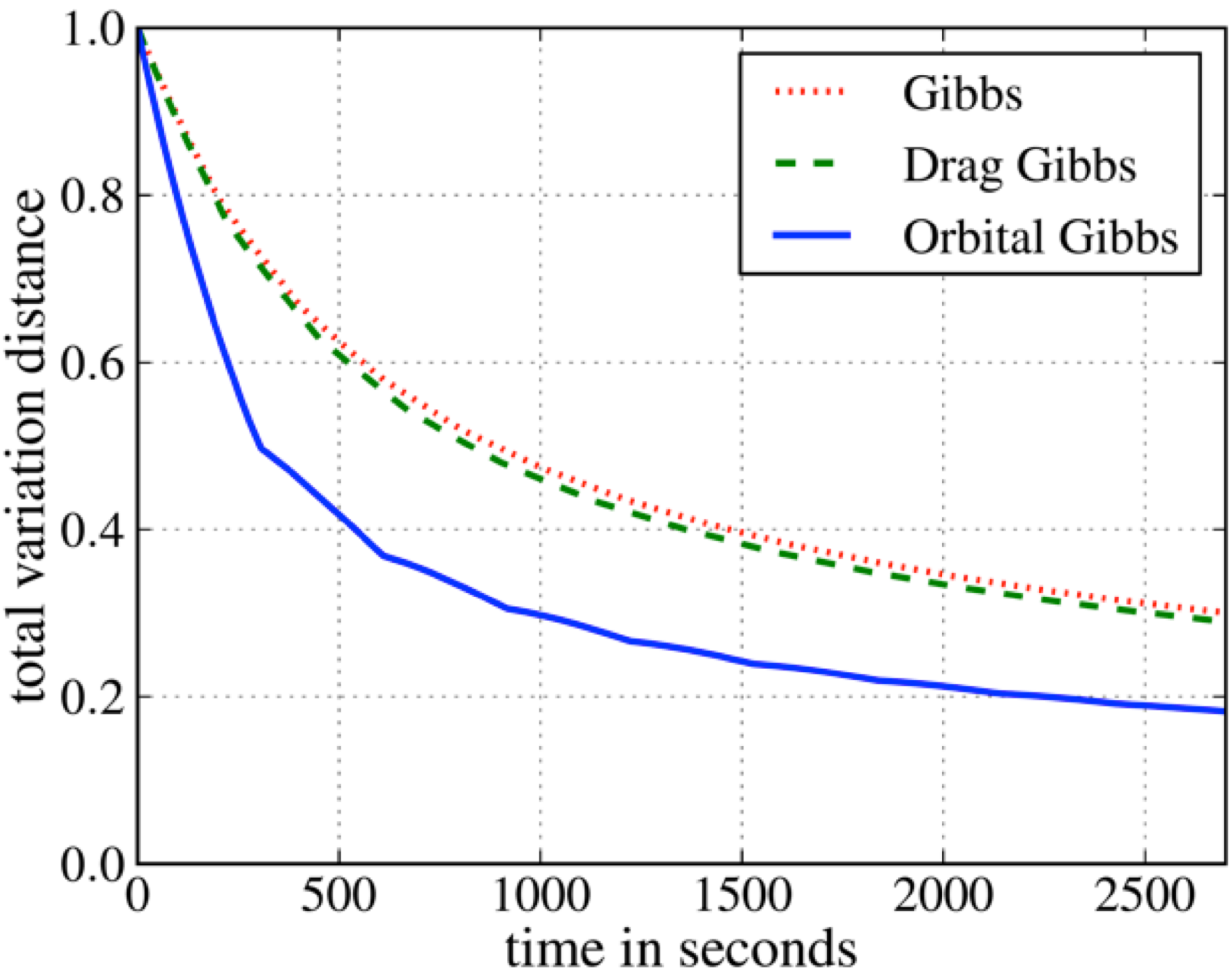
Orbital MCMC Sampling

- Two Markov chains,
  - One ordinary M’
  - One orbital M (based on symmetry groups)
- In each sampling iteration
  1. Run a step of traditional MCMC, chain M’
     - Select a variable V uniformly at random
     - Sample a value for V based on the current states
  2. Sample the state of M uniformly at random from the orbit of the new state of V,
     i.e., select an equivalent state uniformly at random



[Niepert UAI 2012, Van den Broeck, Niepert AAAI 2015]

Orbital MCMC on a 6x6 Ising grid



[Niepert UAI 2012, Van den Broeck, Niepert AAAI 2015]

Lifted Metropolis-Hastings

Given an orbital Metropolis chain A:

- Given symmetry group G (approx. symmetries)
- Orbit $\boldsymbol{x}^G$ contains all states approx. symmetric to $\boldsymbol{x}$
- In state $\boldsymbol{x}$
  1. Select $\boldsymbol{y}$ uniformly at random from $\boldsymbol{x}^G$
  2. Move from $\boldsymbol{x}$ to $\boldsymbol{y}$ with probability $\min\left\{\frac{Pr(\boldsymbol{x})}{Pr(\boldsymbol{y})}, 1\right\}$
  3. Otherwise: stay in $\boldsymbol{x}$ (reject)
  4. Repeat

Colour passing

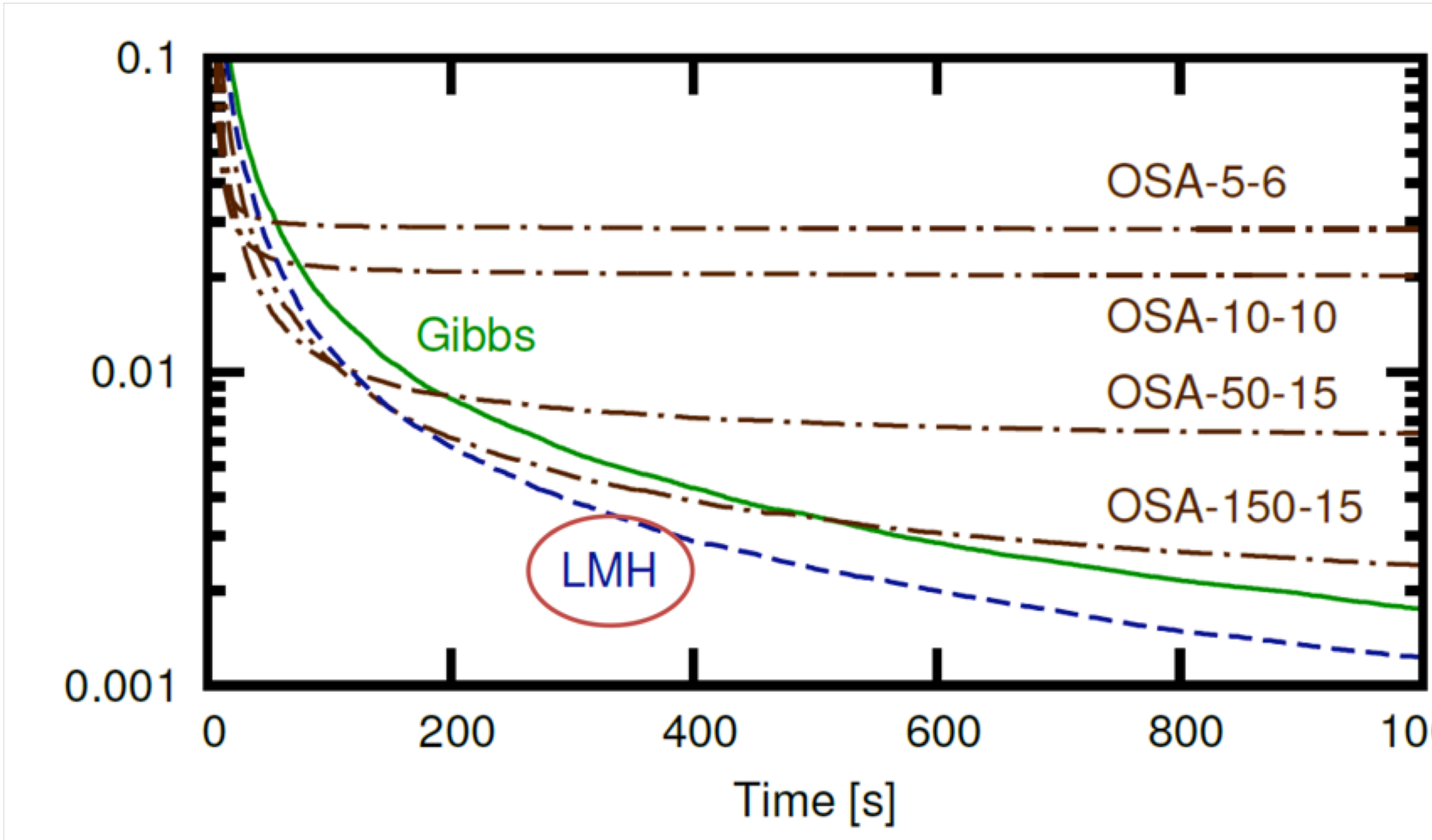
and an ordinary (base) Markov chain B

- With prob. α follow B
- With (1-α) follow A

Account for evidence that may break symmetries, using e.g. approx. symmetries

[Niepert UAI 2012, Van den Broeck, Niepert AAAI 2015]

Lifted Metropolis-Hastings on WebKB



[Niepert UAI 2012, Van den Broeck, Niepert AAAI 2015]

Wrap-up Approximate Lifted Inf.

- Lifted inference exploits (fractional) symmetries
- Fractional symmetries can be computed in quasi-linear time
- Symmetries allow one to study lifted inference in an algebraic way, i.e., independent of the underlying algorithm
- Essentially, the whole family of approximate inference methods is liftable
- Lifted inference of interest to Optimization, ML, and AI in general (SVMs, RL, IRL, Deep Networks, …)

Mission and Schedule of the Tutorial*

Providing an overview and a synthesis of StaR AI

- Introduction 10 min ✓
  - StaR AI
- Overview: Probabilistic relational modeling 40 min ✓
  - Semantics (grounded-distributional, maximum entropy)
  - Inference problems and their applications
  - Algorithms and systems
  - Scalability (limited expressivity, knowledge compilation, approximation)
- Scalability by lifting
  - Exact lifted inference 40+50 min ✓
  - Approximate lifted inference 30 min ✓
- Summary 10 min

*Thank you to the SRL/StaRAI crowd for all their exciting contributions! The tutorial is necessarily incomplete. Apologies to anyone whose work is not cited