ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

D-BSSE
Department of Biosystems
Science and Engineering

Approximate inference: Sampling and variational inference

Niko Beerenwinkel

Outline

- Markov chain Monte Carlo (MCMC)
- Metropolis-Hastings
- Gibbs sampling

- Structure and Order based DAG sampling
  (slides by Jack Kuipers)

- Factorial HMM
- Variational inference

Approximate inference via sampling

Bayesian learning of network structure

- MAP learning:

$$G^* = \underset{G}{\operatorname{argmax}}\, P(G \mid \mathcal{D})$$

- If several networks have similar posterior, we rather want

$$P(G \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid G)P(G)}{P(\mathcal{D})}$$

$$= \frac{P(\mathcal{D} \mid G)P(G)}{\sum_{G'} P(\mathcal{D} \mid G')P(G')}$$

Sampling from the posterior

- In general, sampling from the posterior

$$P(G \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid G)P(G)}{\sum_{G'} P(\mathcal{D} \mid G')P(G')}$$

is difficult, because the marginal likelihood can be intractable and the sum involves all network structures.

Sampling from a distribution

- More generally, we want to approximate an unknown distribution P(X) by a finite sample $X^{(1)}$, ..., $X^{(R)}$, such that

$$\left\{X^{(r)}\right\}_{r=1}^{R} \sim P(X)$$

- How can we obtain such a sample?

Markov Chain Monte Carlo (MCMC)

- Idea: Construct a Markov chain $X^{(n)}$ such that

$$\left\{X^{(n)}\right\}_n \supset \left\{X^{(r)}\right\}_{r=1}^{R} \sim P(X)$$

- Then any feature f of X can be estimated as

$$\mathrm{E}_P[f] = \int f(X)P(X)dX \approx \frac{1}{R}\sum_{r=1}^{R} f\left(X^{(r)}\right)$$

Markov chain

- The transition matrix T of the Markov chain has entries

$$T_{xy} = P\left(X^{(n+1)} = x \mid X^{(n)} = y\right)$$

- If the Markov chain is ergodic (aperiodic and irreducible), then

$$P\left(X^{(n+1)} = x\right) = \int_y T_{xy} P\left(X^{(n)} = y\right)$$

converges to a unique stationary distribution:

$$\lim_{n\to\infty} P\left(X^{(n)}\right) \to P_\infty(X)$$

Detailed balance

- If the detailed balance equations hold,

$$T_{xy}P(X = y) = T_{yx}P(X = x)$$

then the stationary distribution characterized by

$$P_\infty(X = x) = \int_y T_{xy}P(X = y)$$

is the target distribution P(X), because detailed balance implies

$$\int_y T_{xy}P(X = y) = \int_y T_{yx}P(X = x) = P(X = x)$$

Metropolis-Hastings

- Proposal distribution Q_{xy}
- Acceptance probability A_{xy}
- The transition probability is $T_{xy} = Q_{xy} A_{xy}$
- For T to fulfill detailed balance, we need to have

$$Q_{xy}A_{xy}P(X=y) = Q_{yx}A_{yx}P(X=x), \quad \text{or}$$

$$\frac{A_{xy}}{A_{yx}} = \frac{Q_{yx}P(X=x)}{Q_{xy}P(X=y)}$$

- For this, it is sufficient to set

$$A_{xy} := \min\left\{\frac{P(X=x)Q_{yx}}{P(X=y)Q_{xy}}, 1\right\}$$

Metropolis-Hastings algorithm

- Start with a random guess $X^{(0)}$
- For n = 1, ..., N
  - Generate a new point $X^{(n)}$ from the proposal distribution Q,

$$X^{(n)} \sim Q_{X^{(n)},X^{(n-1)}}$$

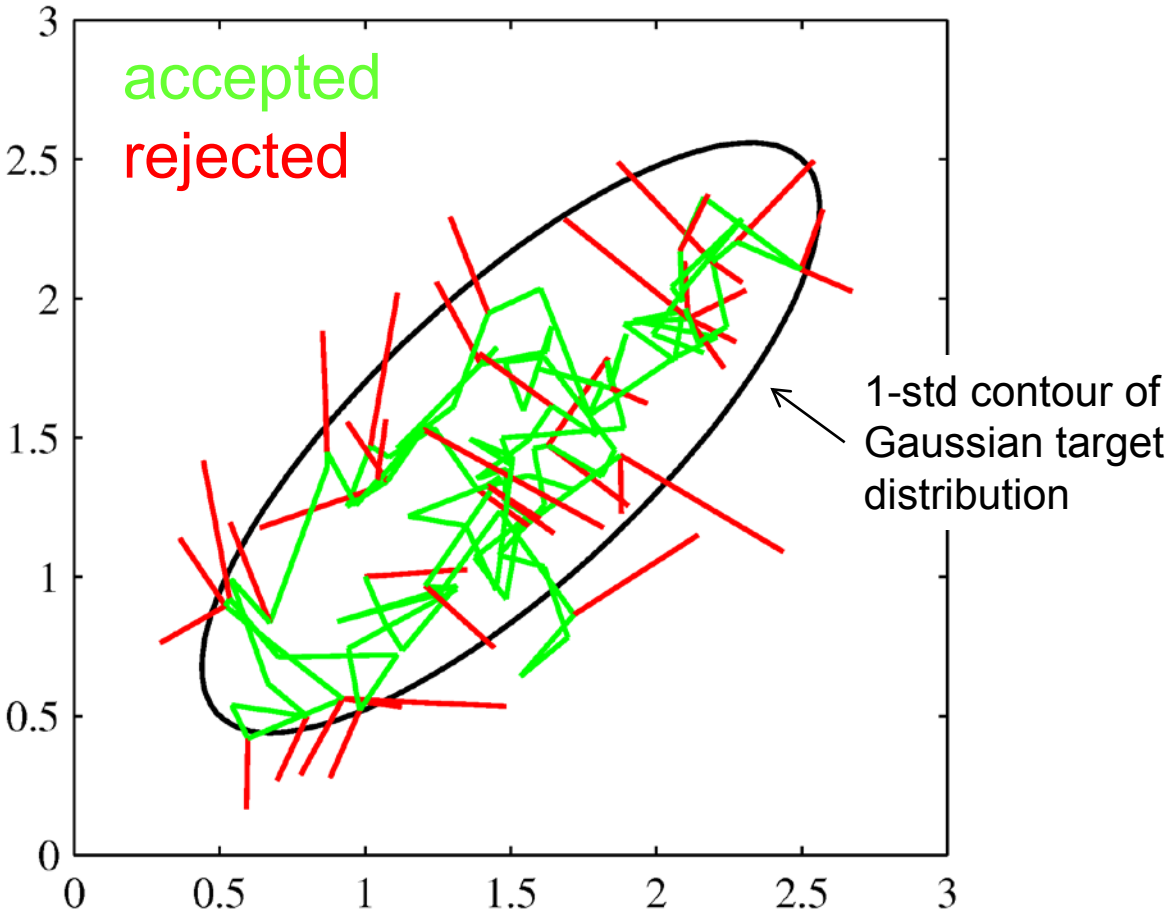
  - Accept the new value with probability

$$A\left(X^{(n)} = x \mid X^{(n-1)} = y\right) = A_{xy} = \min\left\{\frac{P_x Q_{yx}}{P_y Q_{xy}},\ 1\right\}$$

  otherwise, leave the value unchanged, $X^{(n)} = X^{(n-1)}$.
- Discard initial burn-in phase $X^{(0)}$, ..., $X^{(N-R)}$
- Compute expectations

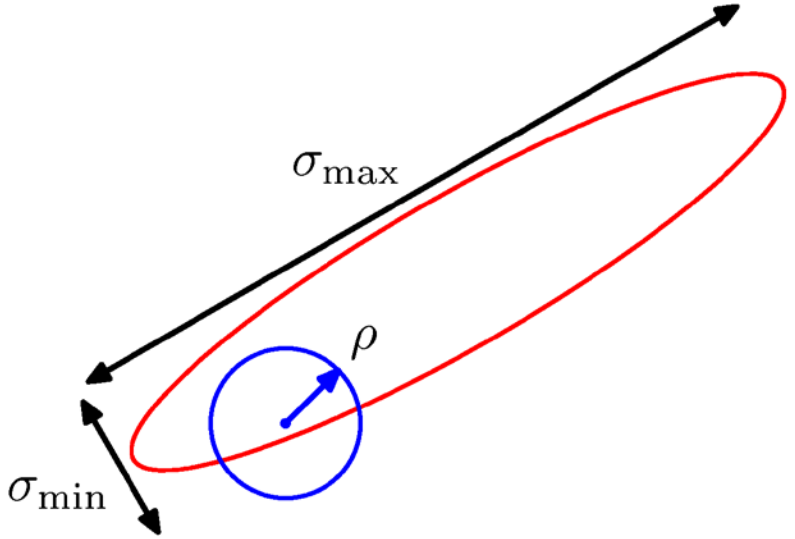
$$\hat{f} = \frac{1}{R}\sum_{r=N-R+1}^{N} f\left(X^{(r)}\right)$$

Example: Q = isotropic Gaussians



Example: Q = isotropic Gaussians

- ρ = scale of proposal distribution Q
- $\rho \approx \sigma_{\text{min}}$ results in a random walk and collecting samples every $(\sigma_{\text{max}} / \sigma_{\text{min}})^2$ steps gives (approximately) independent samples



Gibbs sampling

- Sample conditional probabilities of $P(X_1, ..., X_M)$ iteratively:

$$
\begin{aligned}
X_1^{(n+1)} &\sim P\left(X_1 \mid X_2^{(n)}, \ldots, X_M^{(n)}\right) \\
X_2^{(n+1)} &\sim P\left(X_2 \mid X_1^{(n+1)}, X_3^{(n)}, \ldots, X_M^{(n)}\right) \\
&\vdots \\
X_j^{(n+1)} &\sim P\left(X_j \mid X_1^{(n+1)}, \ldots, X_{j-1}^{(n+1)}, X_{j+1}^{(n)}, \ldots, X_M^{(n)}\right) \\
&\vdots \\
X_M^{(n+1)} &\sim P\left(X_M \mid X_1^{(n+1)}, \ldots, X_{M-1}^{(n+1)}\right)
\end{aligned}
$$

Gibbs sampling as an instance of Metropolis-Hastings

- Regard the conditionals as the proposal distributions Q.
- For the transition $X^{(n)} \rightarrow X^{(n+1)}$ involving variable k, we have

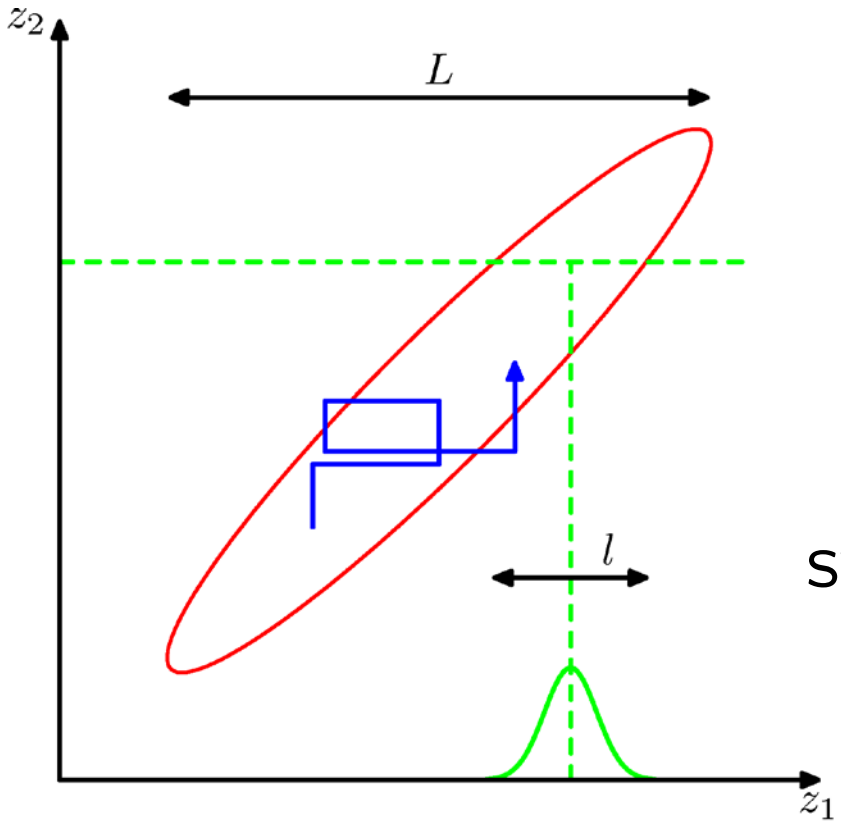
$$Q_{xy} = P\left(X_k^{(n+1)} = x_k \mid X_{\setminus k}^{(n)} = y_{\setminus k}\right)$$

and $$x_{\setminus k} = y_{\setminus k}$$

- Because $P(X^{(n)} = x) = P(x_k \mid x_{\setminus k}) P(x_{\setminus k})$, we find

$$A_{xy} = \frac{P_x Q_{yx}}{P_y Q_{xy}} = \frac{P(x_k \mid x_{\setminus k}) P(x_{\setminus k})\, P(y_k \mid x_{\setminus k})}{P(y_k \mid y_{\setminus k}) P(y_{\setminus k})\, P(x_k \mid y_{\setminus k})} = 1$$

Example: correlated Gaussian target

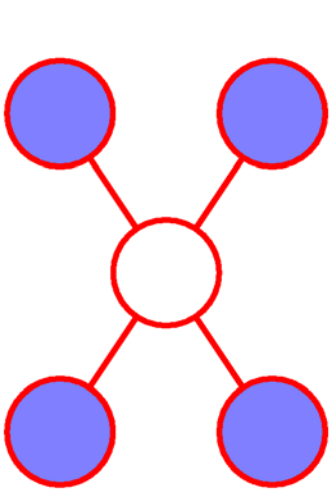


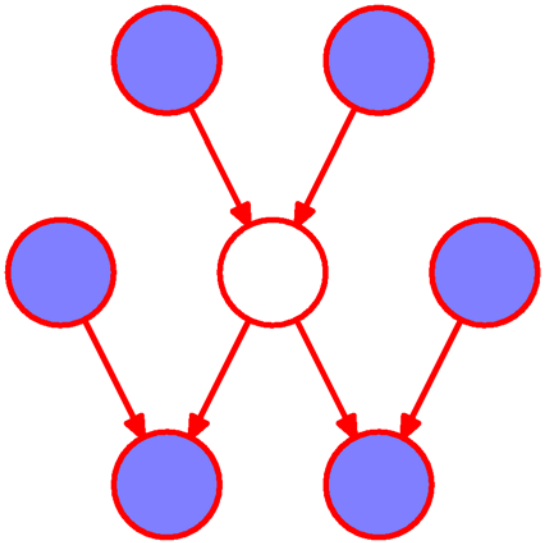
step size $= O(l)$

$O((L/l)^2)$ steps to obtain an independent sample

Gibbs sampling for graphical models

- Gibbs sampling is particularly useful, if it is much easier to sample from the conditionals $P(X_k \mid X_{\setminus k})$ than from the joint distribution $P(X_1, ..., X_M)$.
- For graphical models, $P(X_k \mid X_{\setminus k}) = P(X_k \mid X_{MB(k)})$.





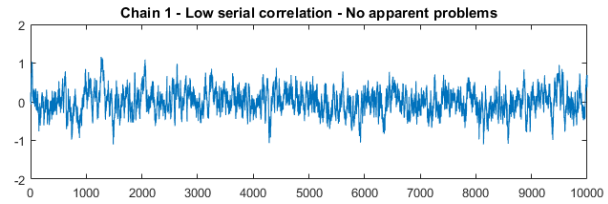
MCMC diagnostics

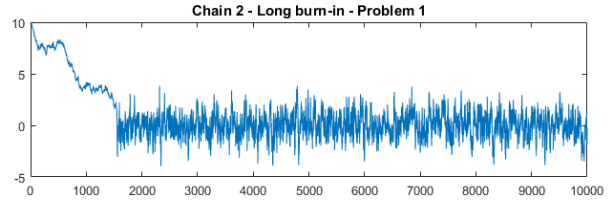
- How can we tell whether the Markov chain has converged?
- We can not. We can only try to spot obvious convergence problems:
  1. Large portions of the sample are drawn from different distributions.
  2. The (effective) sample size is too small.
- Measures to consider:
  - Segment MCMC sample and compare per-segment distributions (e.g., split and compare means; compute MC standard errors)
  - Run multiple chains and compare them (e.g., their means or other moments; Gelman-Rubin statistic)
  - Examine trace plots
  - Examine autocorrelation function (ACF) plots.

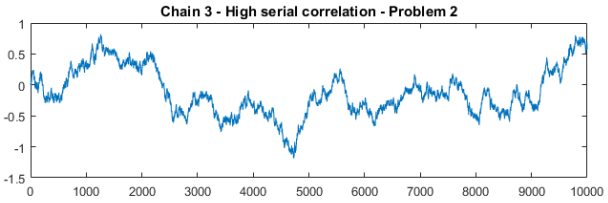
Trace and ACF plots

$$c_{XX}(k) = \frac{1}{N}\sum_{n=1}^{N-k}(X^{(n)} - \bar{X})(X^{(n+k)} - \bar{X}), \quad r_{XX}(k) = c_{XX}(k)/c_{XX}(0)$$

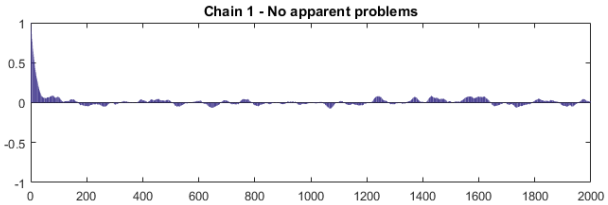
Trace

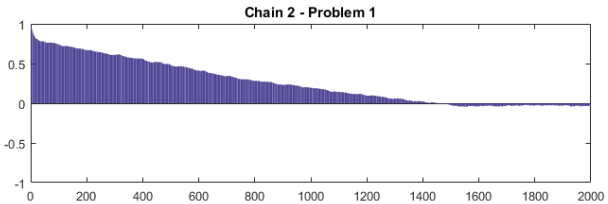


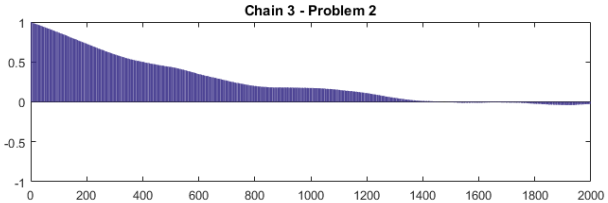




ACF

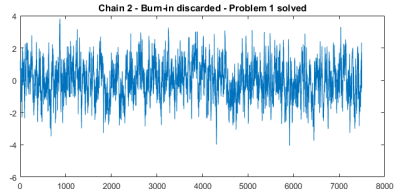




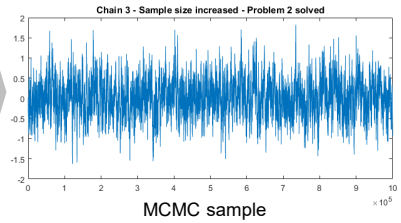


Trace

longer burn-in



larger sample



From: Taboga M, 2021, "MCMC diagnostics"

Sampling graph structures

Metropolis-Hastings for Bayesian networks: Sampling from P(G | $\mathcal{D}$)

- Start with a random DAG $G^{(0)}$
- For n = 1, ..., N
  - Generate a new DAG $G^{(n)}$ from the proposal distribution Q,

$$G^{(n)} \sim Q\left(G^{(n)} \mid G^{(n-1)}\right)$$

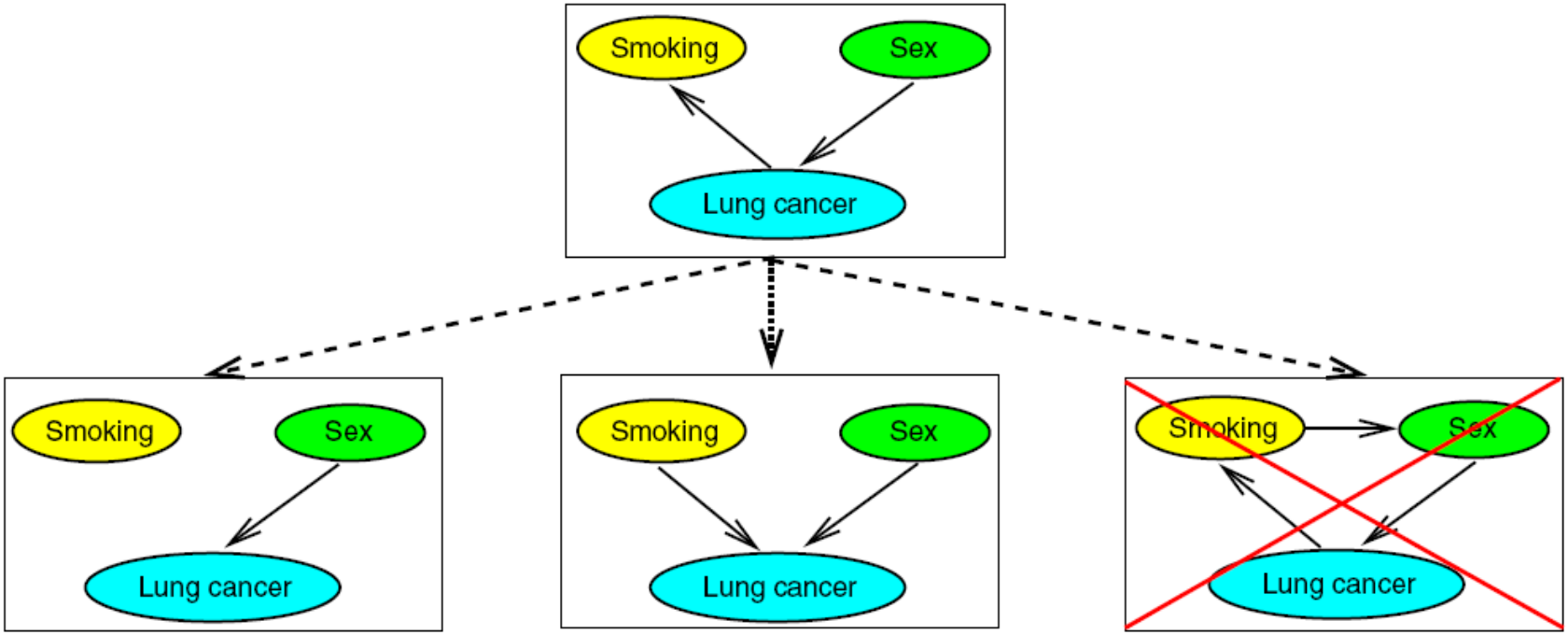
  - Accept the new graph with probability

$$A\left(G^{(n)} \mid G^{(n-1)}\right) = \min\left\{\frac{P(\mathcal{D} \mid G^{(n)})P(G^{(n)})Q(G^{(n-1)} \mid G^{(n)})}{P(\mathcal{D} \mid G^{(n-1)})P(G^{(n-1)})Q(G^{(n)} \mid G^{(n-1)})}, 1\right\}$$

    otherwise, leave the value unchanged, $G^{(n)} = G^{(n-1)}$.

- If $Q\left(G^{(n)} \mid G^{(n-1)}\right) = Q\left(G^{(n-1)} \mid G^{(n)}\right)$, then Q cancels out and the Hastings ratio is 1 in A, and the algorithm reduces to the Metropolis algorithm.

Elementary MCMC moves for DAGs

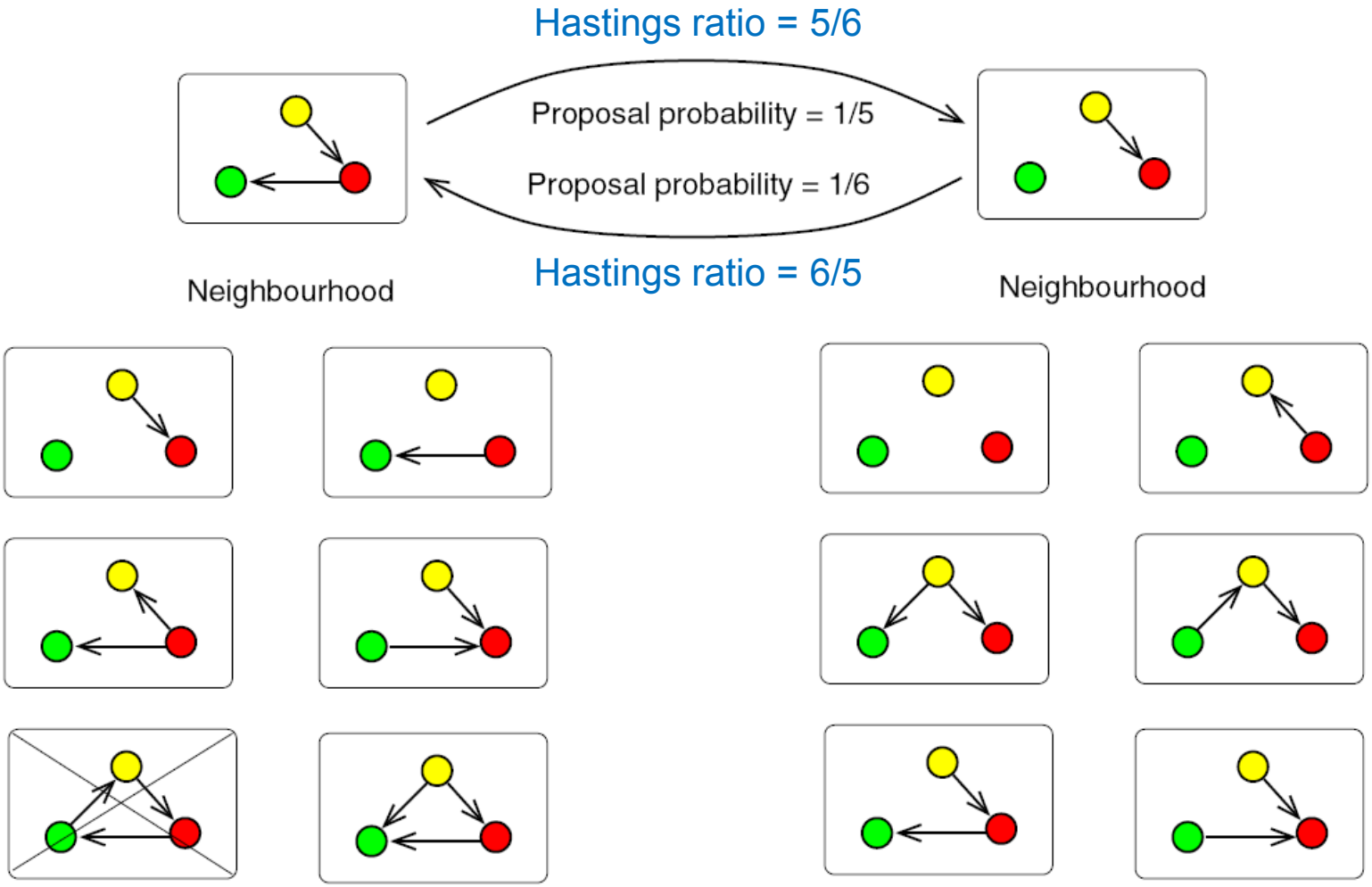


Remove edge

Reverse edge

Add edge

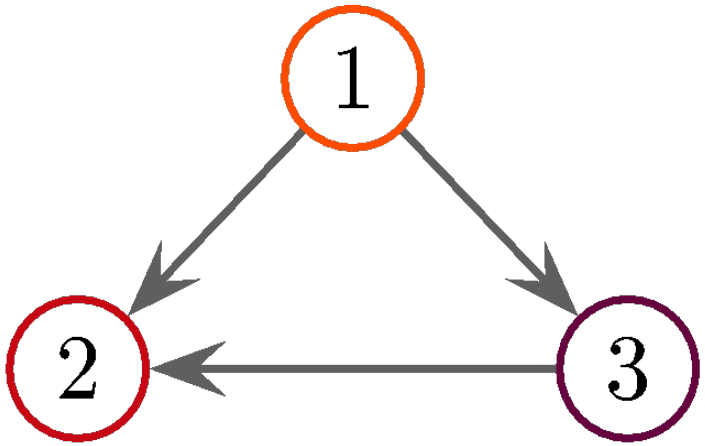
DAG neighborhoods and Hastings ratio



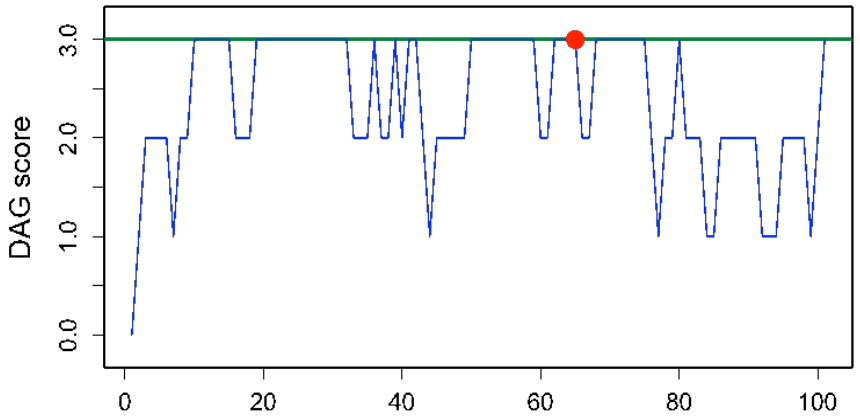
Structure MCMC example

Set $P(G \mid D) \propto e^{E}$ with E # edges

- 2 steps from one high scoring DAG to another



Madigan and York, ISR, 1995

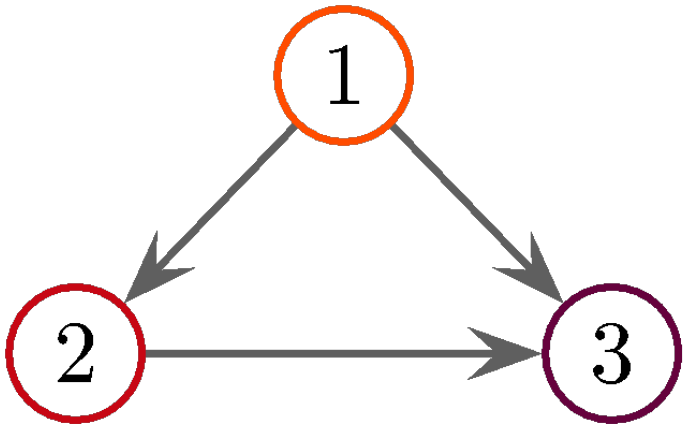


- fair amount of correlation
- slowish convergence

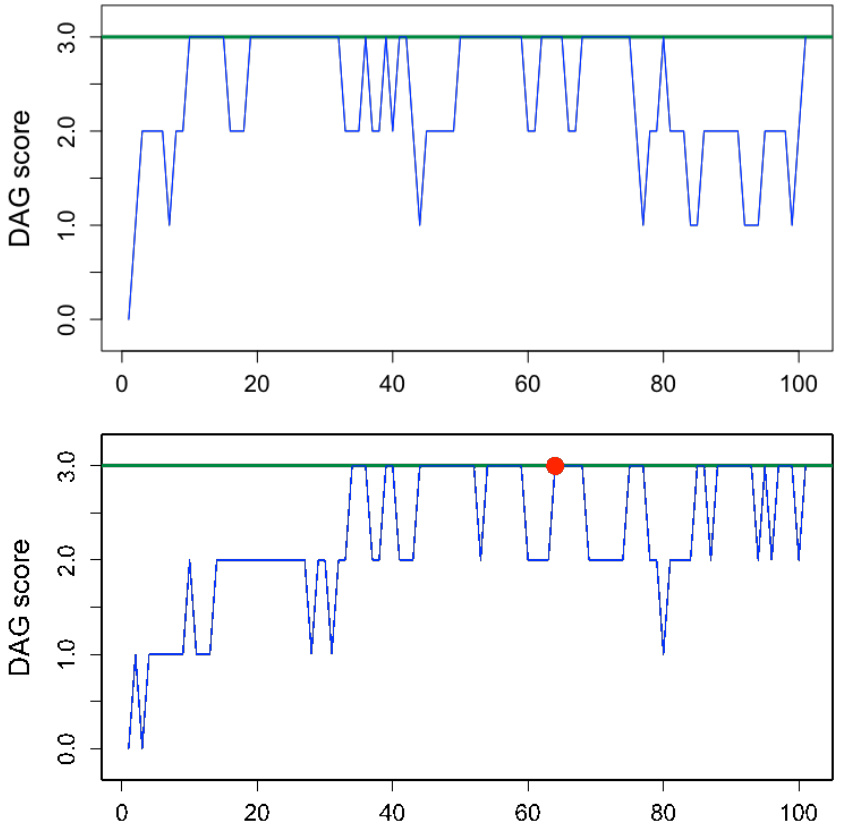
Structure MCMC edge reversal

Now allow edges to be reversed

- Combination of deleting and adding an edge



- Better convergence Giudici and Castelo, ML 2003
- Numerical speed-ups



Order MCMC

Define order on nodes as permutation π Friedmann and Koller, ML 2003

- Parents only further down chain

$$\mathrm{Pa}\{\pi(i)\} \subseteq \{\pi(j) \,|\, j > i\}$$

  - stops cycles

- Combine all DAGs consistent with order and sum scores

$$P(\pi \mid D) = \sum_{G}^{G \prec \pi} P(G \mid D)$$

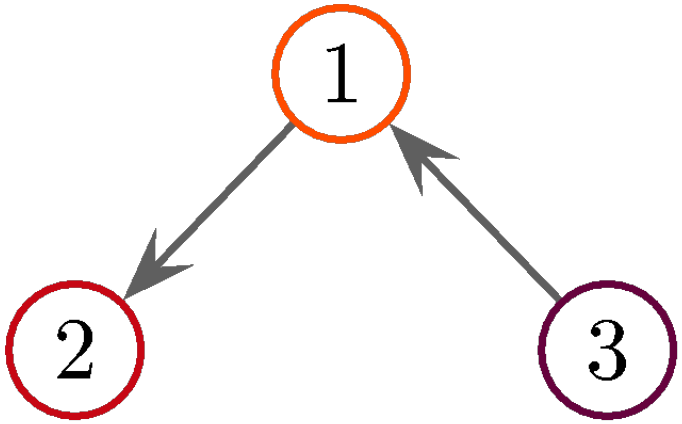
- Build MCMC chain on orders

  - propose a new order π' by swapping two elements
  - accept move with probability $\min\left(1, \frac{P(\pi'|D)}{P(\pi|D)}\right)$

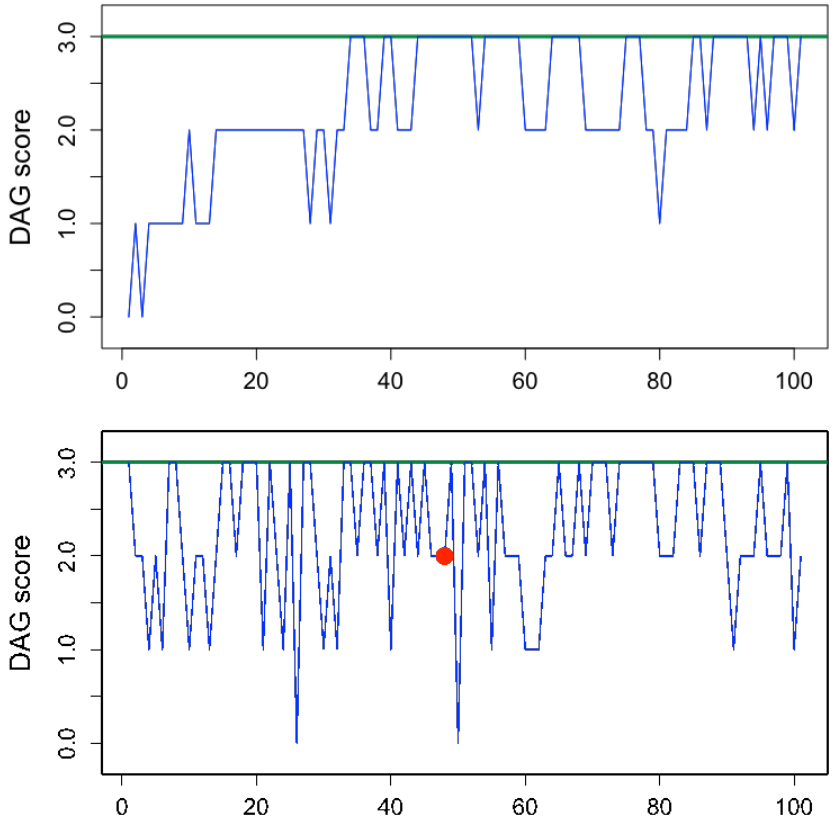
Order MCMC example

Sample DAG from the order

- sample node's parents from permissible scores



- Better convergence
  - combining smooths score landscape
  - smaller space



Friedmann and Koller, ML 2003

Order space

Order space is much smaller

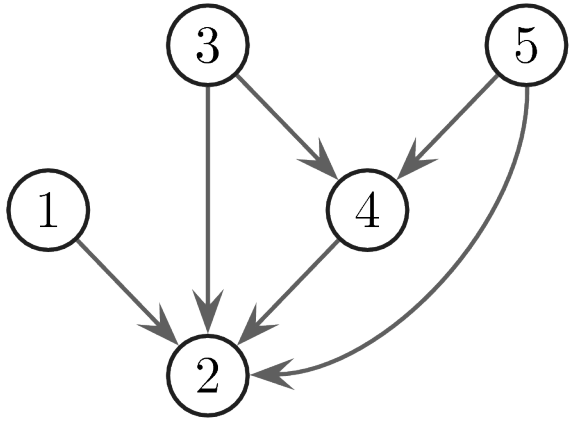
$$\#\,\text{Orders} = n! \qquad \#\,\text{DAGs} \approx n!\,\frac{2^{\frac{n(n-1)}{2}}}{(0.574)(1.48)^n}$$

But not the space of DAGs!

$$\#\,\text{DAGs per order} = 2^{\frac{n(n-1)}{2}}$$

- DAGs exponentially overcounted
- Order MCMC gives a biased sample

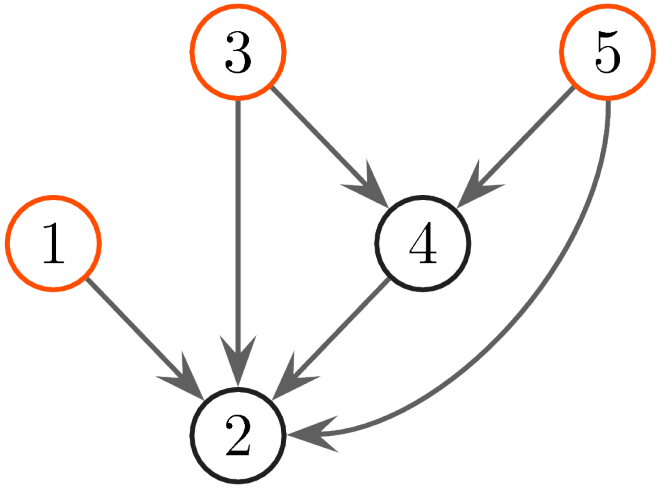
Permuted lower triangular matrices instead

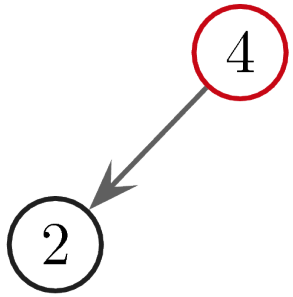


consistent with 8 orders

Outpoints and partitions

Outpoints have no incoming arcs





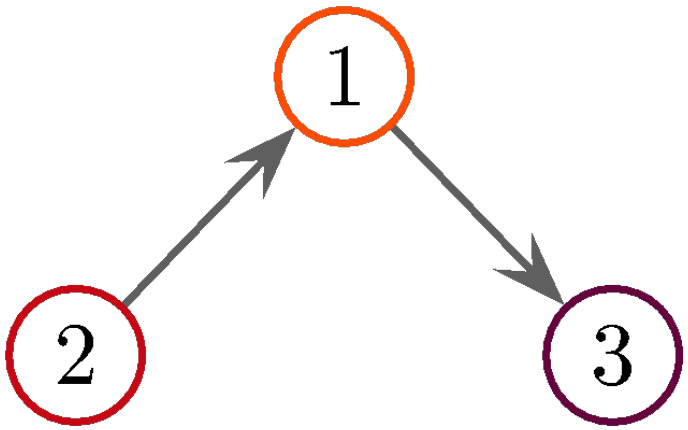
- removing outpoints leaves smaller DAG
- remove till no arcs remain
- sequence $\boldsymbol{k}$ of number removed is ordered partition of n

$$\sum k_i = n\,, \qquad \text{eg} \quad \boldsymbol{k} = [3,1,1]\,, \quad 3+1+1=5$$

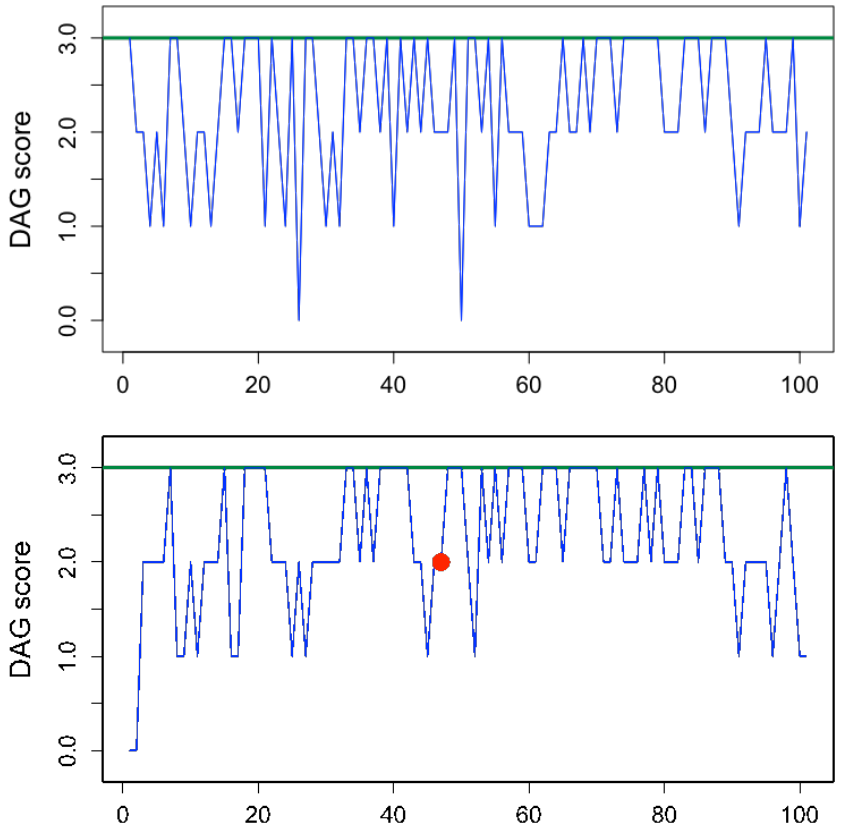
Partition MCMC

Unique representation in space of partitions

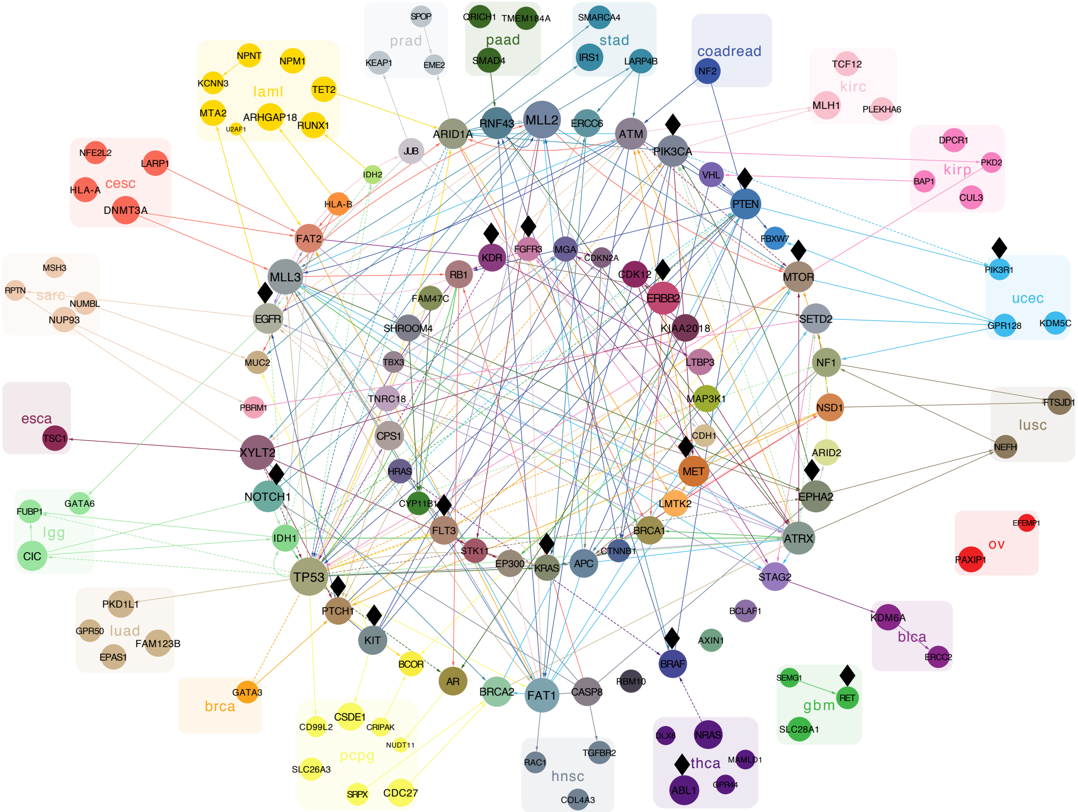
- join or break elements or swap nodes between them



- Convergence slightly worse than order
  - but unbiased
  - much better than structure

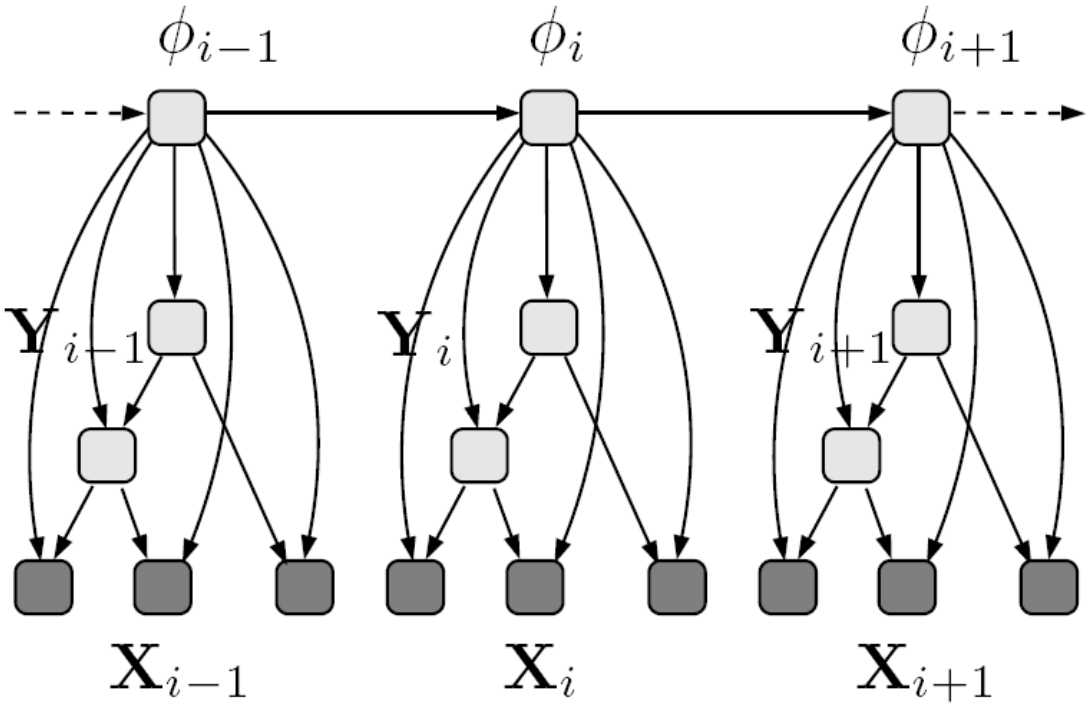


Kuipers and Moffa, JASA 2017


prad
SPOP
KEAP1
EME2
paad
QRICH1
TMEM184A
SMAD4
stad
SMARCA4
IRS1
LARP4B
coadread
NF2
kirc
TCF12
MLH1
PLEKHA6
laml
NPNT
NPM1
KCNN3
TET2
MTA2
ARHGAP18
U2AF1
RUNX1
cesc
NFE2L2
LARP1
HLA-A
DNMT3A
kirp
DPCR1
PKD2
BAP1
CUL3
sarc
MSH3
RPTN
NUMBL
NUP93
ucec
PIK3R1
GPR128
KDM5C
esca
TSC1
lusc
FTSJD1
NEFH
lgg
FUBP1
GATA6
CIC
ov
EFEMP1
PAXIP1
luad
PKD1L1
GPR50
FAM123B
EPAS1
blca
KDM6A
ERCC2
brca
GATA3
gbm
SEMG1
RET
SLC28A1
pcpg
CSDE1
CD99L2
CRIPAK
NUDT11
SLC26A3
SRPX
CDC27
hnsc
RAC1
TGFBR2
COL4A3
thca
DLX6
NRAS
MAMLD1
GPR44
ABL1
ARID1A
RNF43
MLL2
ERCC6
ATM
PIK3CA
VHL
PTEN
JUB
IDH2
HLA-B
FAT2
MLL3
EGFR
MUC2
PBRM1
KDR
FGFR3
MGA
CDKN2A
CDK12
ERBB2
FBXW7
MTOR
RB1
FAM47C
SHROOM4
KIAA2018
SETD2
TBX3
LTBP3
NF1
TNRC18
MAP3K1
NSD1
CPS1
CDH1
ARID2
XYLT2
HRAS
MET
EPHA2
NOTCH1
CYP11B1
LMTK2
FLT3
BRCA1
ATRX
IDH1
STK11
EP300
KRAS
APC
CTNNB1
TP53
STAG2
PTCH1
KIT
BCLAF1
AXIN1
BCOR
AR
BRCA2
FAT1
CASP8
RBM10
BRAF

Approximate inference via variational inference

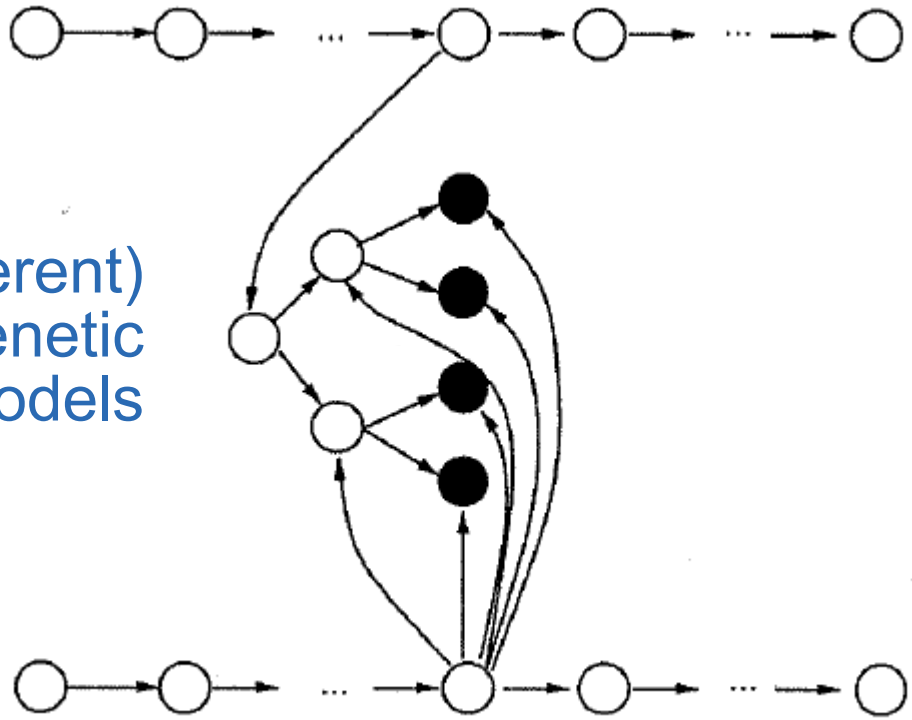
Recall the phylo-HMM



- The phylo-HMM performs poorly in distinguishing rate variation from topology change due to recombination.

Phylogenetic factorial HMM (Phylo-FHMM)

HMM 1



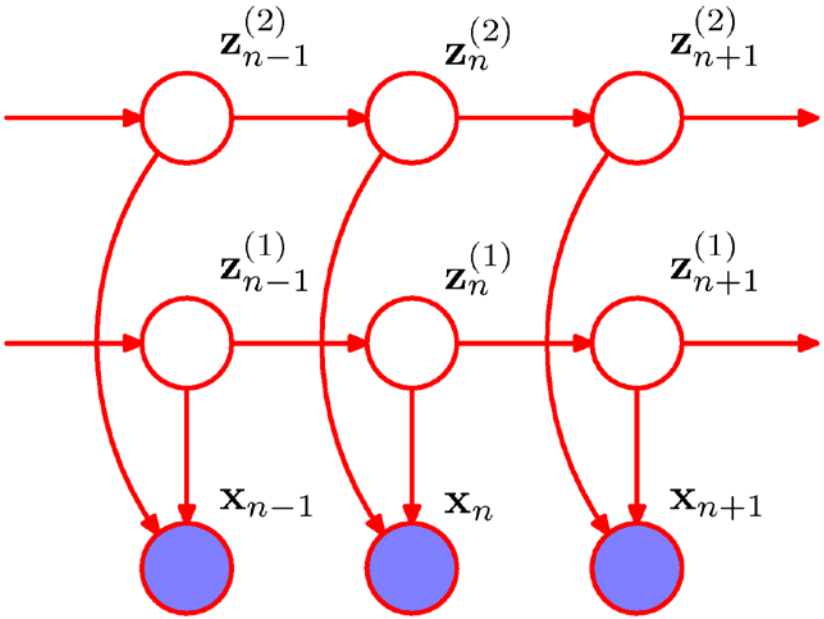
(Different) phylogenetic tree models

HMM 2

indicates changes in topology (recombination)

indicates changes in the rate of evolution (selection pressure)

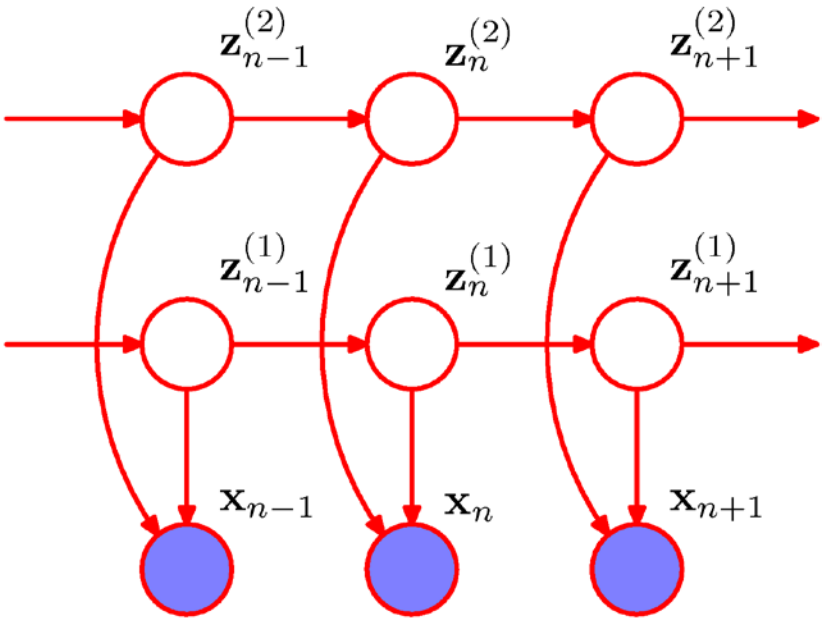
Factorial HMM



- Distributed state space:
  - Let K be the number of states per random variable.
  - Let M be the number of hidden Markov chains.
  - In the FHMM, we need M transition matrices each of dimension K^2, rather than one $K^M \times K^M$ transition matrix in a single HMM.

$$P(X,Z) = \prod_{t=1}^{N} \prod_{m=1}^{M} P\left(Z_t^{(m)} \mid Z_{t-1}^{(m)}\right) P\left(X_t \mid Z_t\right)$$

Linear Gaussian observations



- If P(X | Z) is Gaussian with
  - mean vector μ; each μ_t is a linear combination of the hidden variables, one from each chain,

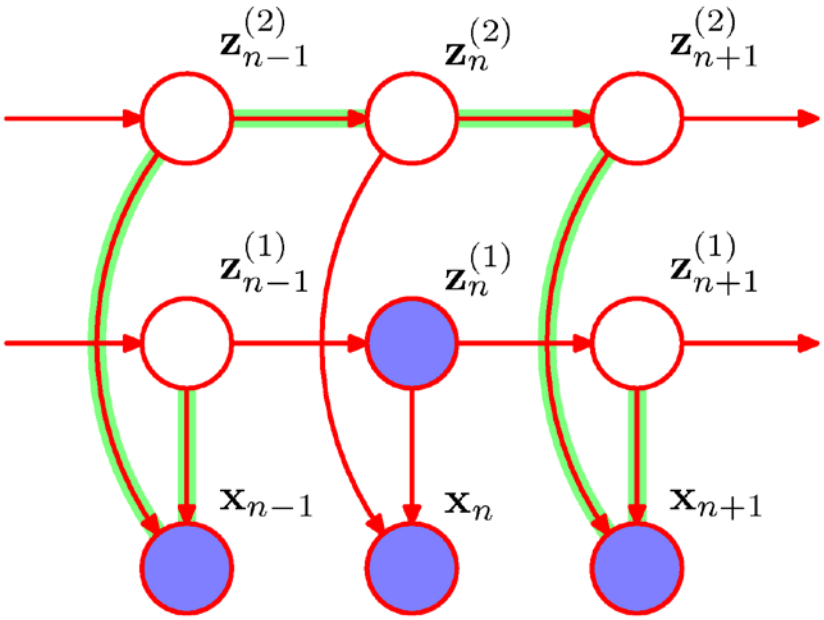
$$\mu_t = \sum_{m=1}^{M} W^{(m)} Z_t^{(m)}$$

  - covariance matrix C,

then the marginals $P(X_t)$ are Gaussian mixture models

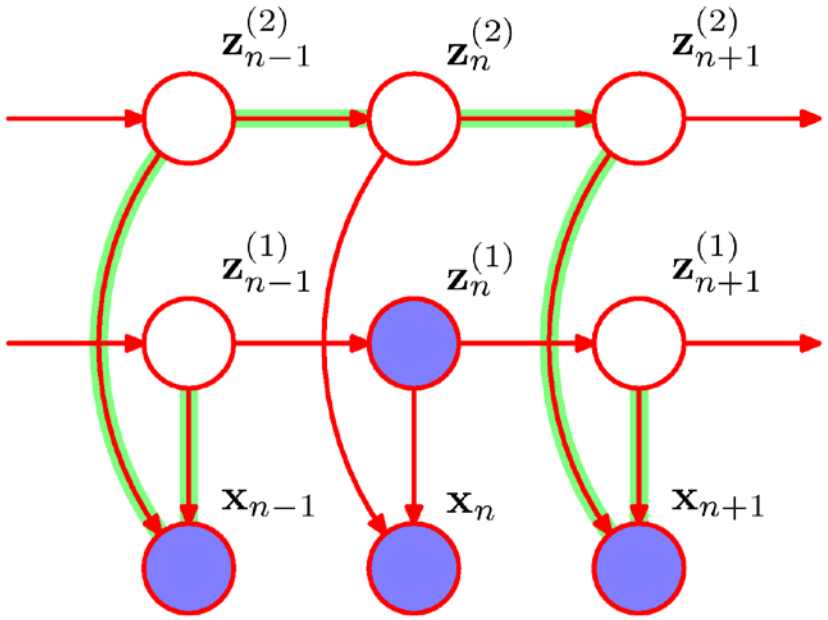
$$P(X_t \mid Z_t) = |C|^{-1/2}(2\pi)^{-D/2} \exp\left\{-\frac{1}{2}(X_t - \mu_t)^t C^{-1}(X_t - \mu_t)\right\}$$

Conditional dependencies



- Observation of X introduces dependencies among chains.
- The green path is not blocked, hence $Z_{n+1}^{(1)}$ is not independent of $Z_{n-1}^{(1)}$ given $Z_{n}^{(1)}$.

Exact inference is computationally expensive



- Exact inference is exponential in M
  - Naïve implementation (forward-backward on big chain): $O(NK^{2M})$
  - Junction tree algorithm: $O(NMK^{M+1})$

→ approximate inference

Gibbs sampling for the FHMM

$$Z_t^{(m)} \sim P\left(Z_t^{(m)} \mid Z_t^{(\setminus m)}, Z_{t-1}^{(m)}, Z_{t+1}^{(m)}, X_t\right)$$

$$\propto P\left(Z_t^{(m)} \mid Z_{t-1}^{(m)}\right) P\left(Z_t^{(m)} \mid Z_{t+1}^{(m)}\right) P\left(X_t \mid Z_t\right)$$

- O(NMK) per sampling step.
- The MCMC sample can be used in the EM algorithm to compute expectations.

Recall the EM algorithm:

- Iterative maximization of the lower bound

$$\log P(X \mid \theta) = F(q, \theta) + \mathrm{D}_{\mathrm{KL}}(q \parallel P)$$

$$\geq F(q, \theta) = \sum_{Z} q(Z) \log \frac{P(X, Z \mid \theta)}{q(Z)}$$

- In the E step, the posterior distribution over the hidden variables, q(Z) = P(Z | X, θ), is computed.
  - This is the (hard) inference problem for FHMMs.
- In the M step, F is maximized with respect to the model parameters.
  - Easy for FHMMs.

Variational approximation

- Because the lower bound holds for any q, we select a simpler family of distributions q(Z | λ),

$$\log P(X \mid \theta) \quad \geq \quad F(q(Z \mid \lambda), \theta)$$

  and maximize with respect to the *variational parameters* λ.

- The complexity of exact inference is determined by the conditional independence relations, not by the parameters.

- Thus, we make simplifying assumptions on the dependency structure, typically removing some of the dependencies in the original model.

Mean field approximation for the FHMM

- The hidden variables Z_t encode K states as binary vectors,

$$Z_{t,k}^{(m)} \in \{0,1\} \quad \text{and} \quad \sum_{k=1}^{K} Z_{t,k}^{(m)} = 1$$

- The mean field approach assumes complete independence

$$q(Z \mid \lambda) = \prod_{t=1}^{N} \prod_{m=1}^{M} q(Z_t^{(m)} \mid \lambda_t^{(m)})$$

- The variational parameters are the means of the latent variables, and

$$q(Z_t^{(m)} \mid \lambda_t^{(m)}) = \prod_{k=1}^{K} \left(\lambda_{t,k}^{(m)}\right)^{Z_{t,k}^{(m)}}$$

Tighten the bound

- To maximize the lower bound F, we minimize the KL divergence $D_{KL}(q(Z \mid \lambda) \,||\, P)$.
- We obtain the fixed point equation

$$\lambda_t^{(m)\,\text{new}} = \varphi\left\{W^{(m)\prime} C^{-1} \bar{X}_t^{(m)} - \frac{1}{2}\Delta^{(m)} + (\log T^{(m)})\lambda_{t-1}^{(m)} + (\log T^{(m)})'\lambda_{t+1}^{(m)}\right\}$$

where

- $\bar{X}_t^{(m)} = X_t - \sum_{\ell \neq m} W^{(\ell)} \lambda_t^{(\ell)}$ is the prediction error,
- $\Delta^{(n)} = \text{diag}(W^{(m)\prime}\, C^{-1}\, W^{(m)})$,
- φ is the softmax operator, $\varphi(A)_i = \exp(A_i) / \sum_j \exp(A_j)$,
- and $\log T^{(m)}$ is the elementwise logarithm of the transition matrix.

Mean field equation, O(NMK²)

$$\lambda_t^{(m)\,\text{new}} = \varphi\left\{W^{(m)\prime}C^{-1}\bar{X}_t^{(m)} - \frac{1}{2}\Delta^{(m)} + (\log T^{(m)})\lambda_{t-1}^{(m)} + (\log T^{(m)})'\lambda_{t+1}^{(m)}\right\}$$

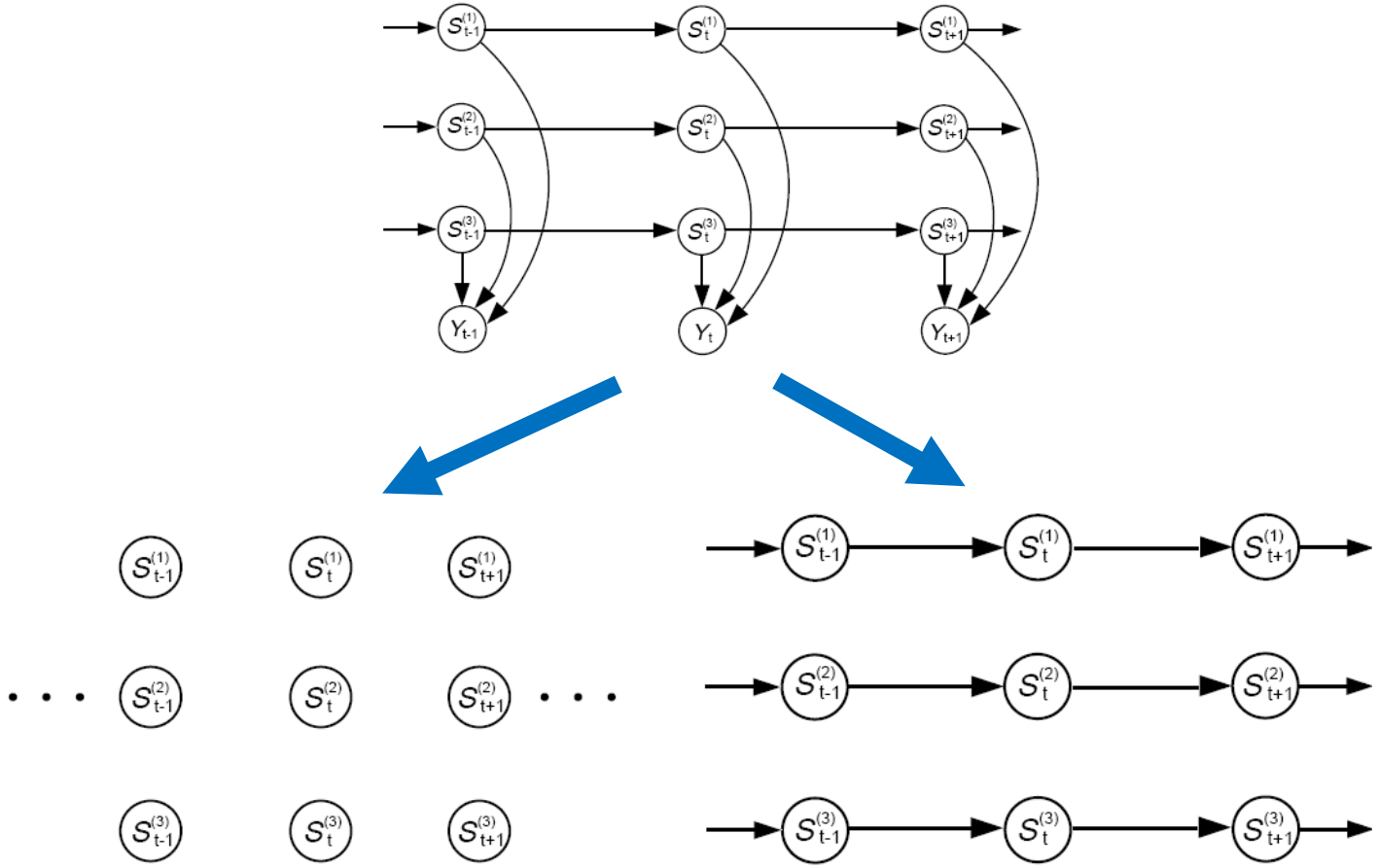
Projection of the error in reconstructing the observations onto the weights of the state vector: the more a state vector reduces this error, the larger the associated variational parameter

No second order correlations under the variational distribution

backward and forward time dependencies among variational parameters

- The fixed point equations introduce dependencies among variational parameters across the Markov blanket.
- The stochastic coupling of the Markov chains is approximated by the deterministic coupling of their means.

FHMM variational inference



"Mean field" "Structured" variational inference

Summary

- **Markov chain Monte Carlo** (MCMC) is a powerful method for drawing samples from distributions that are difficult to assess. Popular MCMC algorithms are Metropolis-Hastings and, especially for graphical models, the Gibbs sampler.
- **Order based samplers** combine large groups of graphical models for more efficient inference**.**
- **Variational inference** refers to approximating the marginal distribution of interest by a simpler parametric family of distributions. In the mean field approach, this distribution is fully factorized. Minimizing the KL divergence gives fixed point equations for the variational parameters.

References

- Bishop CM. Pattern Recognition and Machine Learning. Springer, 2007.
- Jordan MI (ed.). Learning in Graphical Models. Kluwer Academic Publishers, 1998.
  - MacKay DJC. Introduction to Monte Carlo methods.
  - Jordan MI, Ghahramani Z, Jaakkola TS, Saul LK. An introduction to variational methods for graphical models.
- Ghahramani Z, Jordan MI (1997). Factorial hidden Markov models. *Machine Learning* 29, 245−273.