

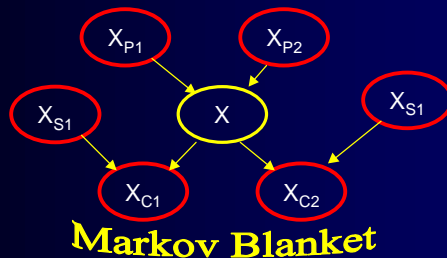
Today

- Simulation
 - More Gibbs
 - AA Algorithm
 - Backward Sampling (not)
- Structural Approximation
 - Boyen and Koller
 - Talk courtesy of Xavier Boyen, xb@cs.stanford.edu

Gibbs Sampling

- For each observed variable E_i , set $E_i = e_i$.
- Use any sampling technique (usually forward-sampling or importance-sampling) to set X_i to some random value.
- Repeat
 - Pick some unobserved variable X_i
 - Sample $X_i[j] \sim P\{X_i | X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n\}$

Transition Probability



- Transition Probability

$$P\{X_i | X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_N\} = P\{X_i | MB(X_i)\}$$

$$P\{X_i | MB(X_i)\} = \alpha P\{X_i | pa(X_i)\} \prod_{j \in ch(X_i)} P\{X_j | (pa(X_j) \setminus X_i), X_i\}$$

Markov Chain

- Markov Chain Monte Carlo (MCMC)
- Transition Matrix

$$P\{X_i[t+1] | MB(X_i)[t]\}$$

- Stationary distribution
 - Irreducible, Aperiodic
 - (Infinite state space) Positive recurrent.
- Convergence thought to be governed by rate of *mixing*:

$$\sup_{A,B} |P\{x[n] \in A, x[0] \in B\} - \pi\{x[n] \in A\}\pi\{x[0] \in B\}|$$

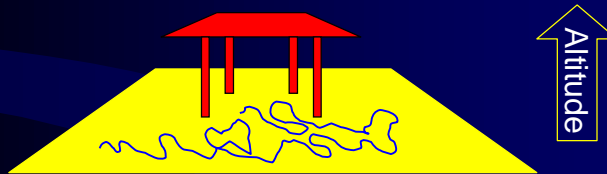
Procedures

- Parallel:
 - “Burn-in” by Gibbs for some period of time. Sample.
 - Sample is in $P\{X|E\}$
- Serial:
 - If $x[i]$ is from $P(X|E)$, then, so is $x[i+1]$.
 - Say that x_1 is flipped:

$$\begin{aligned}P\{X[i+1]\} &= \sum_{x_1} P\{x'_1 | x_2, \dots, x_n\} P\{x_1, x_2, \dots, x_n\} \\ &= \sum_{x_1} P\{x'_1 | x_2, \dots, x_n\} P\{x_1 | x_2, \dots, x_n\} P\{x_2, \dots, x_n\} \\ &= P\{x'_1 | x_2, \dots, x_n\} P\{x_2, \dots, x_n\} \sum_{x_1} P\{x_1 | x_2, \dots, x_n\} \\ &= P\{x'_1, x_2, \dots, x_n\}\end{aligned}$$

Gibbs Sampling Convergence

- Estimate average altitude of surfaces in room
- Statistics on samples suggest convergence, but really suggest only convergence on a given mode.



Simple Example

- $P\{X\} = [0.5, 0.5]$
- $P\{Y|X=T\} = [1-a, a]$
- $P\{Y|X=F\} = [a, 1-a]$
- Transitions:

$$P\{X_i'=T | Y=T\} = \frac{0.5(1-a)}{0.5a + 0.5(1-a)} = 1-a$$

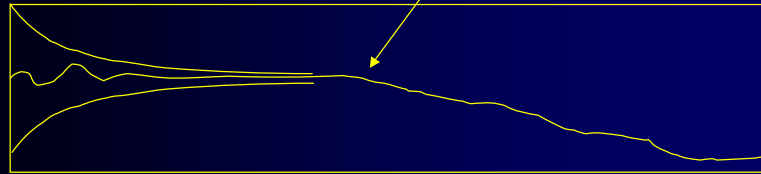
$$P\{Y_i'=T | X=T\} = 1-a$$

Simple Gibbs Example

- Suppose $(X, Y) = (T, T)$
- Sample $X \ Y \ X \ Y \ X \ Y$
- How long until we sample something other than (T, T) ?
 - $1/a$

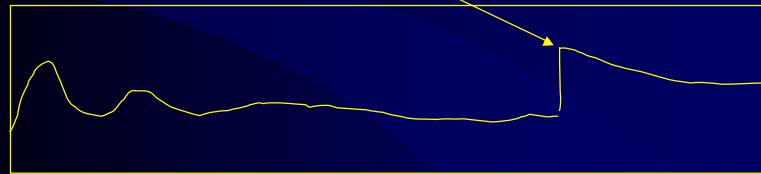
Bad Convergence

MCMC



Change to another mode

Likelihood Weighting



Found 1 sample with high likelihood

Bounded Variance Algorithm

$$upper(X_i = x_i) = \max_{x_{pa} \in \text{pa}(X_i)} P\{X_i = x_i | x_{pa}\}$$

$BV\{\epsilon, \delta, X_i, X, E\}$

$$S^* = 4(e-2) \ln\left(\frac{2}{\delta}\right) \frac{1+\epsilon}{\epsilon^2}$$

$$K_E = \prod_{i \in E} upper(X_i)$$

$$\mu_E = LW_Cum\{X, E, S^*, K_E\}$$

$$K_i = upper(X_i = x_i) K_E$$

$$\mu_i = LW_Cum\{X \setminus \{X_i\}, E \cup \{X_i\}, S^*, K_i\}$$

return (μ_i, μ_E)

LW_cumulative $\{Z, E, S^*, K\}$

$S = 0$

$N = 0$

repeat while $S < S^*$

$Z[i] = \text{Forward_Sample}(Z, E, 1)$

$\omega[i] = \text{path likelihood}(Z[i])$

$$S \leftarrow S + \frac{\omega[i]}{K}$$

$N \leftarrow N + 1$

return $\frac{KS}{N}$

AA Algorithm

Estimate means

$$(\mu_i, \mu_E) = BV \left\{ \varepsilon = \frac{1}{2}, \delta = \frac{\delta}{3}, X_i, X, E \right\}$$

Estimate variance

$$Y = 8(e-2) \ln \left(\frac{2}{\delta} \right) \frac{1}{\varepsilon^2}$$

$$N_E = \frac{Y\varepsilon}{\mu_E}, \quad N_i = \frac{Y\varepsilon}{\mu_i}$$

$$Z_E[1 \dots N] = F_S(X, E, N_E)$$

$$\sigma_E^2 = \min \{ \text{Var}(\lambda(Z_E)), \varepsilon \mu_E \}$$

$$Z_i[1 \dots N] = F_S(X \setminus X_i, E \cup X_i, N_E)$$

$$\sigma_i^2 = \min \{ \text{Var}(\lambda(Z_i)), \varepsilon \mu_i \}$$

Simulate Query

$$N_{E2} = \frac{Y\sigma_E^2}{\mu_E^2}, \quad N_{i2} = \frac{Y\sigma_i^2}{\mu_i^2}$$

$$Z_{E2}[1 \dots N] = F_S(X, E, N_{E2})$$

$$Z_{i2}[1 \dots N] = F_S(X \setminus X_i, E \cup X_i, N_{E2})$$

$$\rho(X_i | E) = \frac{\sum_{k=1}^{N_{i2}} \lambda(Z_{i2}[k])}{\sum_{k=1}^{N_{E2}} \lambda(Z_{E2}[k])}$$

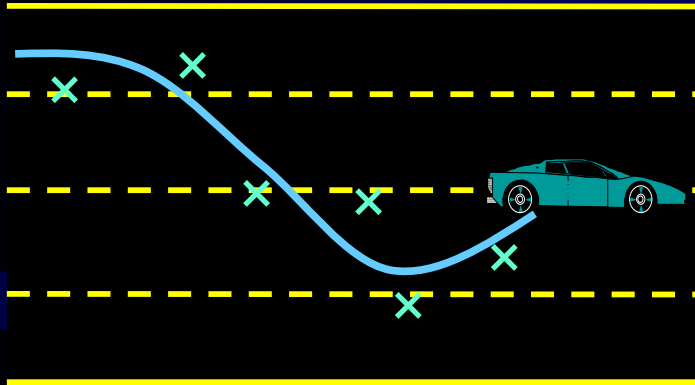
Theorem: Within a small constant of the optimal number of samples!

Tractable Inference for Complex Stochastic Processes

Xavier Boyen
Daphne Koller

Stanford University

Stochastic dynamic system



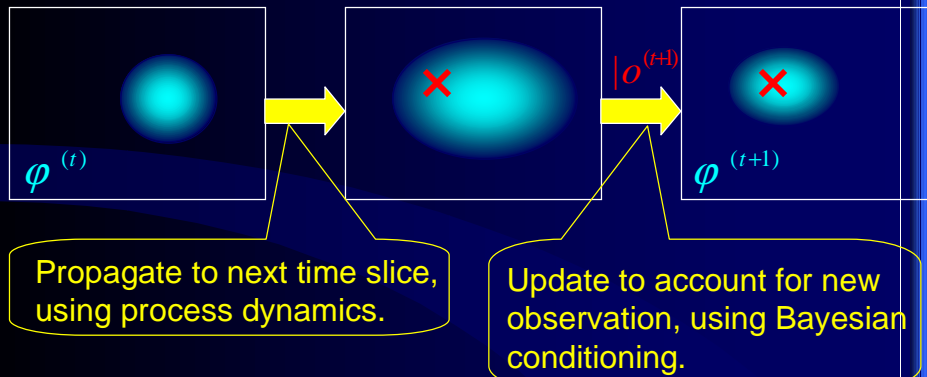
- Process evolves over time
- Dynamics are noisy or unpredictable
- Process state only partially observable

Monitoring

- Task: Online monitoring of current process state, based on observations obtained until now.
- Exact state is unknown \Rightarrow maintain distribution $\varphi^{(t)}$ over possible states at time t

Monitoring in theory

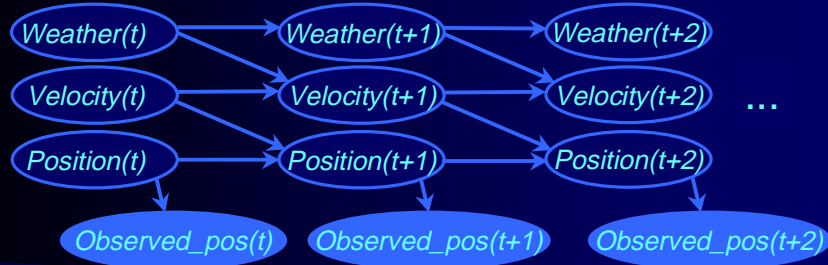
Maintaining belief state is easy (in theory):



Monitoring in practice

- Sometimes, belief state admits compact representation & manipulation
 - e.g., Kalman filters
 - ⇒ supports effective monitoring algorithm.
- Is this the case for other, more expressive representations?
 - dynamic Bayesian networks (DBNs);
 - hybrid processes (discrete + continuous).

Dynamic Bayesian networks

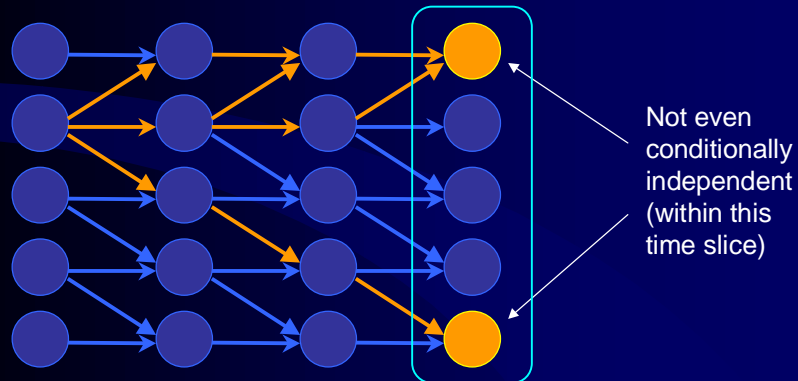


- For DBNs, belief state = distribution over all possible assignments to state variables
 \Rightarrow Explicit belief state is exponential in #variables
- But, surely we can exploit the structure, “as usual” ?
Structure \Rightarrow fast inference ... (??)

No, it's a myth !

- **Problem:** even in highly structured DBNs, belief state variables become fully correlated

DBN structure $\not\Rightarrow$ belief state structure



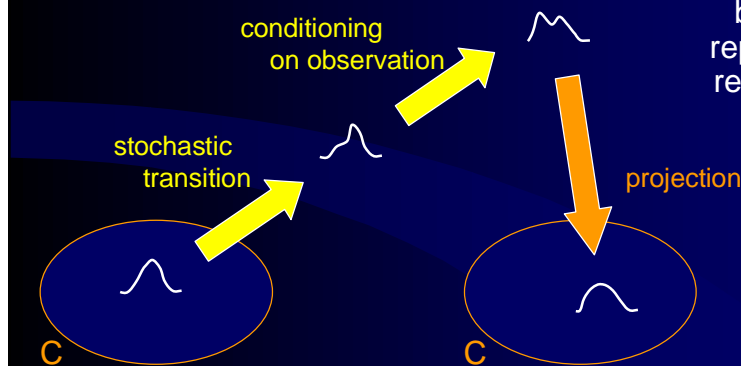
Approximate monitoring

- Exact inference in DBNs is intractable !!
⇒ Major barrier to DBN inference in realistic networks.
- Idea: Maintain approximate belief state
- Choose some predetermined subspace C of compactly representable distributions, e.g.:
 - Gaussian mixtures with few components;
 - decomposable distributions.

Approximate monitoring

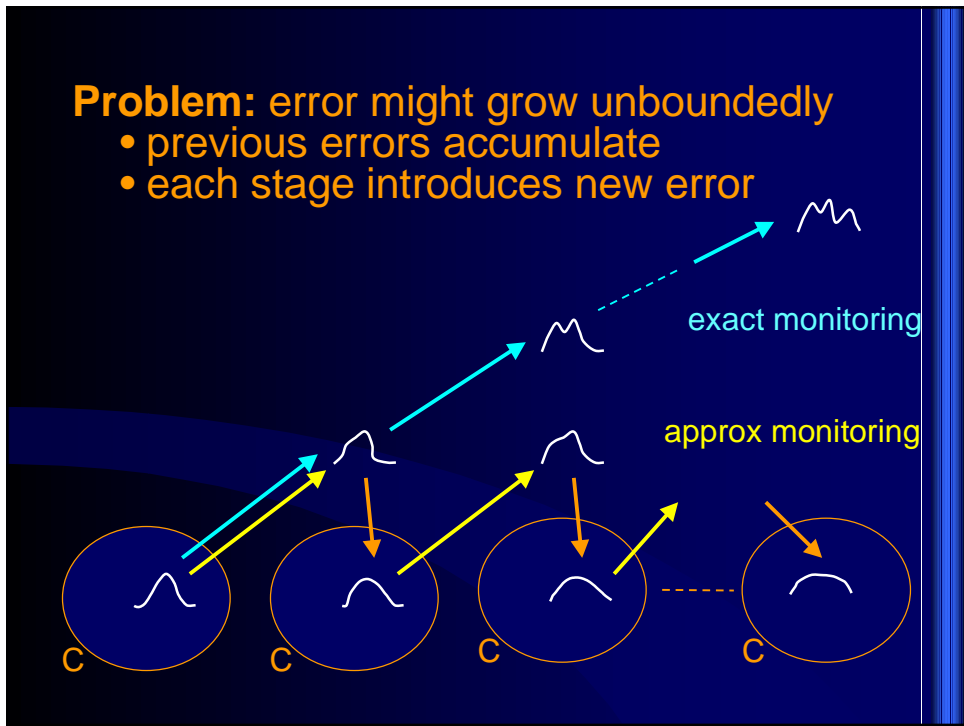
- propagate current belief → expected next belief
- condition on evidence → next belief
- project onto compact distribution in C .

Benefit:
belief state
representation
remains small

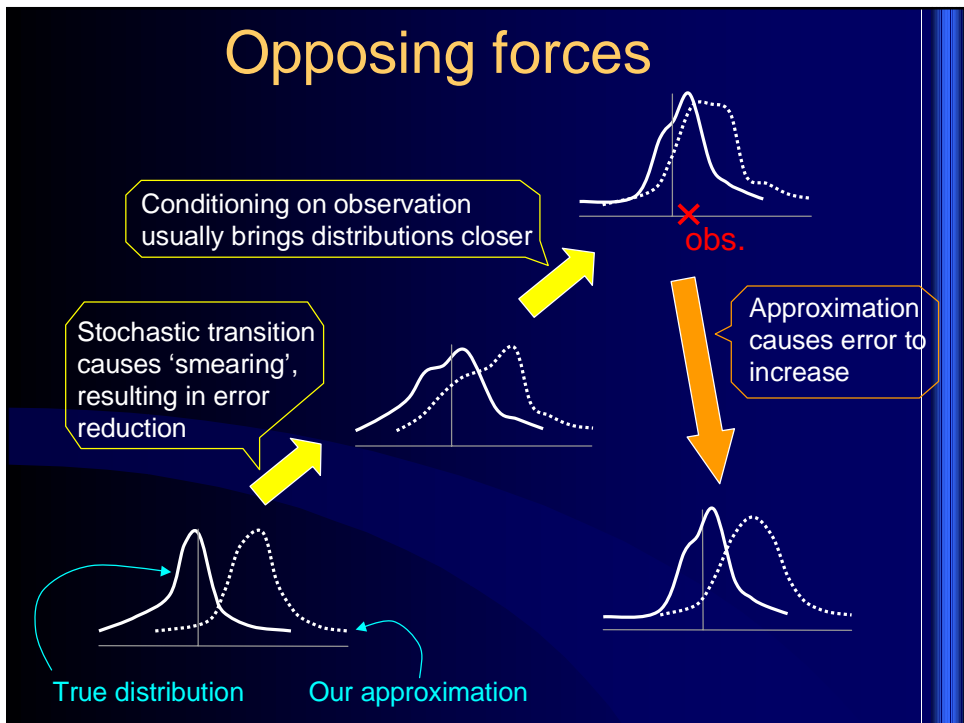


Problem: error might grow unboundedly

- previous errors accumulate
- each stage introduces new error



Opposing forces



Contraction & approximation

If :

- propagation through transition model decreases error by γ (contraction property);
- observations decrease error on expectation;
- approximation error at each stage bounded by ϵ ;

Then, total expected error is, at all times :

$$\leq \underbrace{\epsilon}_{\text{time } t \text{ error}} + \underbrace{(1-\gamma)\epsilon}_{\text{time } t-1 \text{ error}} + \underbrace{(1-\gamma)^2\epsilon}_{\text{time } t-2 \text{ error}} + \dots \leq \underline{\underline{\epsilon/\gamma}}$$

Contraction

Transition Matrix $P[X[t+1] | X[t]]$

Mixing $\gamma = \min_{x^{(i)}, x^{(j)}} \left(\sum_{x^{(k)} \in \Omega} \min [P\{x^{(k)} | x^{(i)}\} P\{x^{(k)} | x^{(j)}\}] \right)$ $\Omega = X_1 \times \dots \times X_N$

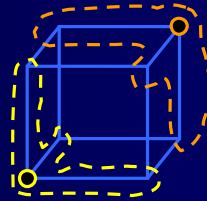
KL $D[\varphi || \psi] = \sum_{x \in \Omega} \varphi\{x\} \ln \frac{\varphi\{x\}}{\psi\{x\}}$

Contraction

$$D[\varphi' || \psi'] \leq (1-\gamma) D[\varphi || \psi]$$

Factored processes

- Problem: Contraction rate γ for large processes can be exponentially low. (even if structured !)



- Clever design of approximation scheme can exploit process structure, if:
 - process is composed of weakly interacting subprocesses
 - our approximation decomposes the belief state according to these subprocesses.

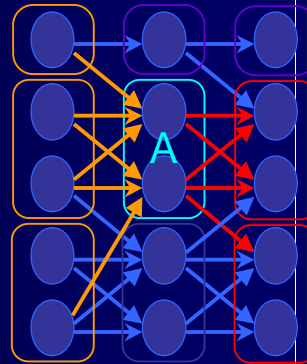
Theorem

If each subprocess

- depends on at most r others,
- influences at most q others,
- contracts at rate at least γ ,

Then,

$$\gamma_{Whole} \geq (\gamma/p)^q$$



E.g., for A,
 $r = 3, q = 2.$

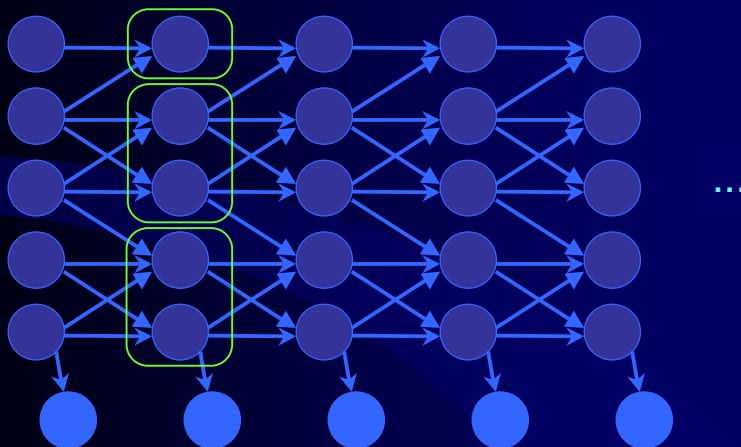
Thus, interactions between processes are costly:

- Incoming influences reduce contraction rate linearly
- Outgoing influences reduce it exponentially.

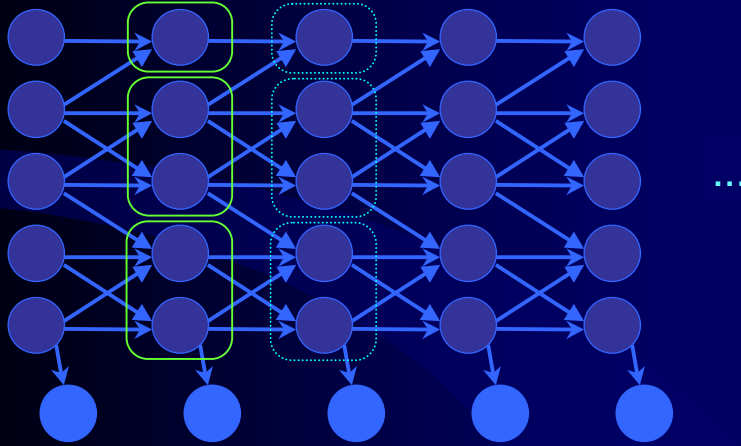
DBN Algorithm

- So, the DBN structure can be exploited for approximate inference !
- We partition state variables into “subprocesses”
 - maintain approximate belief state as product of marginally independent “sub-beliefs”;
 - update belief and project back using junction trees.

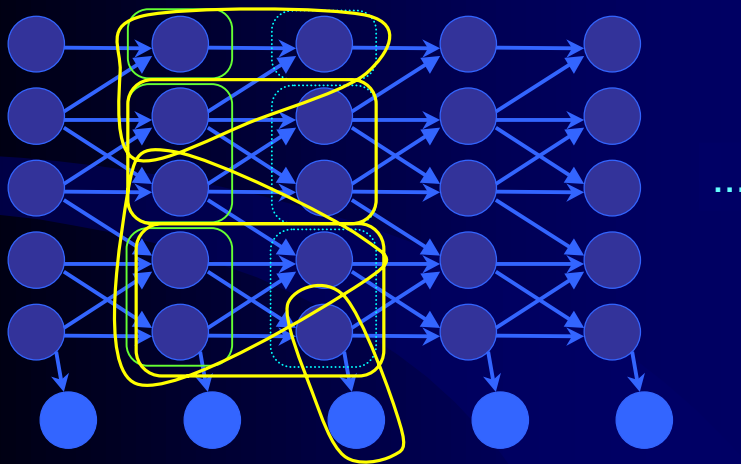
1. Start from current approximate belief.



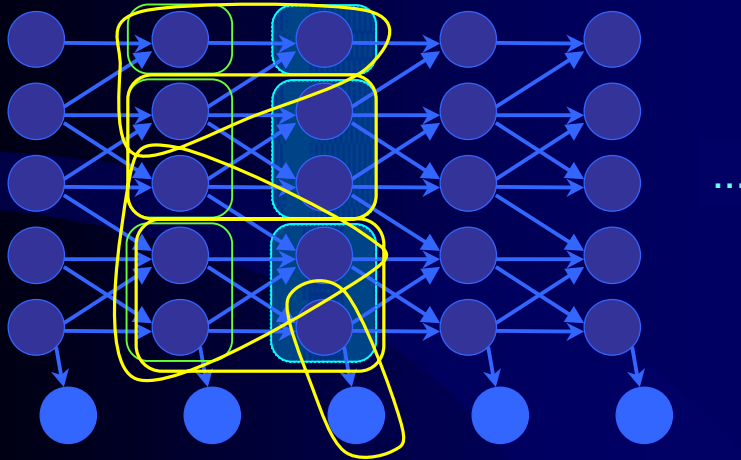
1. Start from current approximate belief.
2. Want to compute next belief.



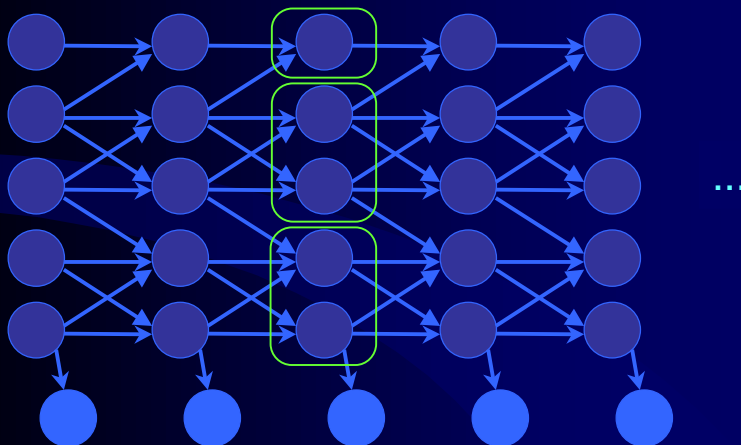
1. Start from current approximate belief.
2. Want to compute next belief.
3. Create + calibrate clique tree.



1. Start from current approximate belief.
2. Want to compute next belief.
3. Create + calibrate clique tree.
4. Extract components of next belief by marginalization.



1. Start from current approximate belief.
2. Want to compute next belief.
3. Create + calibrate clique tree.
4. Extract components of next belief by marginalization.
5. Continue with next belief ...

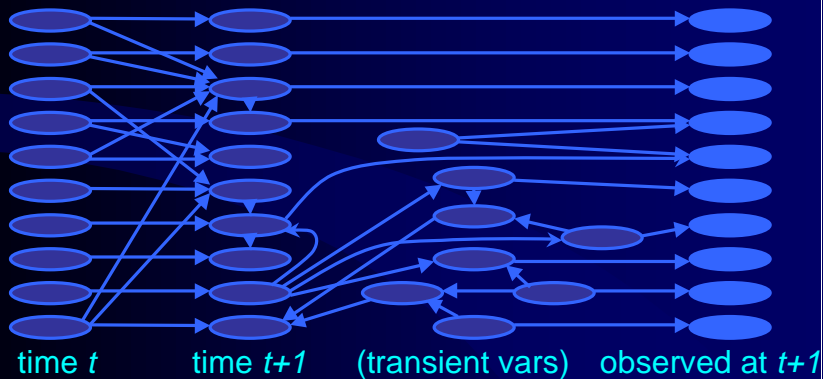


DBN Algorithm : discussion

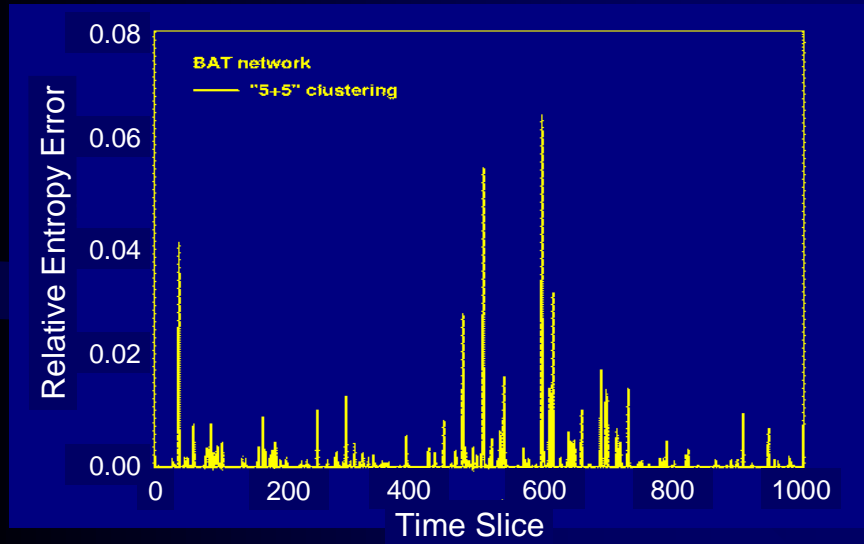
- Smaller partitions \Rightarrow
 - + faster inference
 - + better contraction
 - worse approximation
- Better contraction & approximation if partition along weak/sparse interactions.
- Very simple procedure.
- First algorithm to provably exploit structure in general DBNs.

Experimental setup

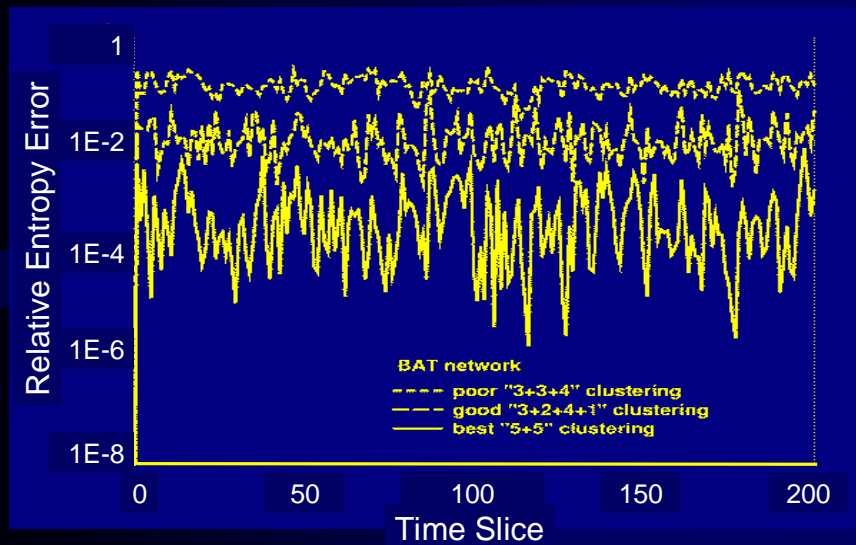
- Used BAT network for freeway traffic monitoring [Forbes et al.]
- Maintained approximate belief states given evidence sequences generated from network.



Typical evolution of error



Comparing partitions



Experimental results

Network	Max error	Avg error	Speedup
BAT	0.065	0.0007	15
	0.02	0.00013	} L_1 error for specific variables
	0.07	0.00019	
WATER	0.14	0.06	31
	0.018	0.0015	13 } conditionally independent belief state

- Error remains bounded indefinitely
- Good partitions help, as predicted
- Conditionally independent belief states are even better.

Conclusions

- Simple fast approximate inference in dynamic systems
 - Guaranteed error bounds.
 - Uses a new contraction property for relative entropy.
- Achieves orders of magnitude speedup for DBNs at minor cost in accuracy.
 - Savings dramatically larger for more complex processes.
 - First algorithm to exploit structure of general DBNs.
 - Practical inference algorithm for large DBNs!
- Major step towards reasoning about complex, real-life dynamic systems.

More Conclusions

- Doesn't matter what the approximation algorithm is...
 - If contraction holds under the algorithm, approximation has bounded error.