Infinite Dynamic Bayesian Networks

Finale Doshi-Velez
David Wingate
Joshua Tenenbaum
Nicholas Roy

ICML 2011
Learning Time-Series Models

Time series data often has complex structure we wish to understand...

What factors predict future internet traffic?

http://www.caida.org/tools/measurement/autofocus/
Learning Time-Series Models

Time series data often has complex structure we wish to understand...

Can we understand how neurons interact from spike trains?

http://www.caida.org/tools/measurement/autofocus/
Learning Time-Series Models

Time series data often has complex structure we wish to understand...

Are there simple models to explain what happens when sprites interact?
Learning Time-Series Models

Time series data often has complex structure we wish to understand...

http://www.caida.org/tools/measurement/autofocus/

What are the temporal patterns in weather data?
Example: Rainfall Data

But finding that structure isn't always obvious...

478 weather stations
4 discrete precipitation values
Data from US HCN, 1980-1989
Example: Rainfall Data

But finding that structure isn't always obvious...
Example: Rainfall Data

But finding that structure isn't always obvious...
Example: Rainfall Data

How can we go from no structural information to discovering causal patterns?
Current Approaches

There are several ways we might try to encode the structure:

- Hidden Markov Models (HMMs): one latent factor
Current Approaches

There are several ways we might try to encode the structure:

- Hidden Markov Models (HMMs): one latent factor

Infinite HMMs (Beal et al., 2002) infer number of visited states
Current Approaches

There are several ways we might try to encode the structure:

- Hidden Markov Models (HMMs): one latent factor
- Factorial HMMs: many latent factors, special structure
Current Approaches

There are several ways we might try to encode the structure:

- Hidden Markov Models (HMMs): one latent factor
- Factorial HMMs: many latent factors, special structure

\[
O(o|s) \quad T(s'|s) \quad \text{Infinite Factorial HMMs (Van Gael et al., 2002) infer number of used factors}
\]
Current Approaches

There are several ways we might try to encode the structure:

- Hidden Markov Models (HMMs): one latent factor
- Factorial HMMs: many latent factors, special structure
- Dynamic Bayes Nets: many factors, general structure
Example: Rainfall Data

Hidden factors might correspond to regions, connections represent direction of weather.
Example: Rainfall Data

Observed nodes are the stations, explained by the nearby regions.
Example: Rainfall Data

Key Question:
how many factors do we need?
Current Approaches

There are several ways we might try to encode the structure:

- Hidden Markov Models (HMMs): one latent factor
- Factorial HMMs: many latent factors, special structure
- Dynamic Bayes Nets: many factors, general structure

Our contribution:

Infinite DBNs to infer number of used factors for a general DBN structure
Using Nonparametric Bayes: Distributions over Infinite DBNs

Infinite DBN prior over DBNs with an infinite number of factors

Data

Posterior over DBNs, structure of used factors
Using Nonparametric Bayes: Distributions over Infinite DBNs

Infinite DBN prior over DBNs with an infinite number of factors

Data

Posterior over DBNs, structure of used factors

Infer:
- Number of latent factors used
- State of each factor at each time
- Causal structure for transitions and emissions

Parameters adjust bias towards fewer factors/more states or more factors/fewer states.
Why this is tricky...

A finite amount of data must be explained by a finite number of parameters – easy to run into trouble!
Why this is tricky...

A finite amount of data must be explained by a finite number of parameters – easy to run into trouble!

Suppose the apriori probability of a factor choosing being a parent of factor k was $p_k$
Why this is tricky...

A finite amount of data must be explained by a finite number of parameters – easy to run into trouble!

Then the generative process would each factor choosing child $k$ with $p_k$
Why this is tricky...

A finite amount of data must be explained by a finite number of parameters – easy to run into trouble!

Then the generative process would each factor choosing child k with $p_k$
Why this is tricky...

A finite amount of data must be explained by a finite number of parameters – easy to run into trouble!

Then the generative process would each factor choosing child $k$ with $p_k$
Why this is tricky...

A finite amount of data must be explained by a finite number of parameters – easy to run into trouble!

Then the generative process would each factor choosing child $k$ with $p_k$
Why this is tricky...

A finite amount of data must be explained by a finite number of parameters – easy to run into trouble!

Then the generative process would each factor choosing child $k$ with $p_k$
Why this is tricky...

A finite amount of data must be explained by a finite number of parameters – easy to run into trouble!

Then the generative process would each factor choosing child $k$ with $p_k$
Why this is tricky...

A finite amount of data must be explained by a finite number of parameters – easy to run into trouble!

Then the generative process would each factor choosing child $k$ with $p_k$
Why this is tricky...

A finite amount of data must be explained by a finite number of parameters – easy to run into trouble!

But there are an infinite number of factors!

Which means $k$ has an infinite number of parents!
Why this is tricky...

A finite amount of data must be explained by a finite number of parameters – easy to run into trouble!

But there are an infinite number of factors!

Which means $k$ has an infinite number of parents!

Our Approach: let children pick parents

- Will imply parents have infinite children
- But only factors affecting observed nodes matter
The iDBN Generative Process
Observed Nodes Choose Parents

Treat each parent as a dish in the Indian Buffet Process: Popular parents more likely to be chosen.
Hidden Nodes Choose Parents

Treat each parent as a dish in the Indian Buffet Process: Popular parents more likely to be chosen.
Hidden Nodes Choose Parents

Treat each parent as a dish in the Indian Buffet Process: Popular parents more likely to be chosen.
Hidden Nodes Choose Parents

Key point: we only need to instantiate parents for nodes that help predict values for the observed nodes.
Other infinite nodes are still there, we just don't need them to explain the data.
Instantiate Parameters

Sample transition
CPTs ~ HDP for all
hidden nodes

Sample observation
CPTs ~ H0 for all
observed nodes
Summary of the Prior
Summary of the Prior

$\alpha_{DBN}$

$P_{a_k}$

$P_{a_n}$

$X_t$

$X_{t+1}$

$Y_t$

$Y_{t+1}$

$T$

$\alpha_{HDP}$

$\gamma$

$\beta$

$\Omega$

$H$

$\alpha_{HDP}$ determines CPTs, expected number of states per factor
Summary of the Prior

\[ \alpha_{\text{DBN}} \] sets structure, expected number of factors... if parents chosen \(~\text{IBP}\), can guarantee finite factors to explain finite data.
Inference

General Approach: Blocked Gibbs sampling with the usual tricks (tempering, sequential initialization, etc.)

- Resample factor-factor connections: $p(P_{a_n} | P_{a_k}, X, \beta)$
  - Gibbs sampling

- Resample factor-observation connections: $p(P_{a_k} | P_{a_n}, X, \beta)$
  - Gibbs sampling

- Resample transitions: $p(T | P_{a_k}, X, \beta)$
  - Dirichlet-multinomial

- Resample observations: $p(\Omega | P_{a_n}, X, \beta, Y)$
  - Dirichlet-multinomial

- Resample state sequence: $p(X | P_{a_n}, P_{a_k}, \beta, T, \Omega, Y)$
  - Factored frontier – Loopy BP

- Add / delete factors: $p(P_{a_n} | P_{a_k}, X, \beta)$
  - Metropolis-Hastings birth/death
Inference

General Approach: Blocked Gibbs sampling with tempering, sequential initialization, etc.

| Resample factor-factor connections | \( p(P_{an} | P_{ak}, X, \beta) \) | Gibbs sampling |
|-----------------------------------|---------------------------------|----------------|
| Resample factor-observation connections | \( p(P_{ak} | P_{an}, X, \beta) \) | Gibbs sampling |
| Resample transitions | \( p(T | P_{ak}, X, \beta) \) | Dirichlet-multinomial |
| Resample observations | \( p(\Omega | P_{an}, X, \beta, Y) \) | Dirichlet-multinomial |
| Resample state sequence | \( p(X | P_{an}, P_{ak}, \beta, T, \Omega, Y) \) | Factored frontier – Loopy BP |
| Add / delete factors | \( p(P_{an} | P_{ak}, X, \beta) \) | Metropolis-Hastings birth/death |

Common to all DBN inference
Inference

General Approach: Blocked Gibbs sampling with (tempering, sequential initialization, etc.)

- Resample factor-factor connections: $p(P_{a_n} | P_{a_k}, X, \beta)$
- Resample factor-observation connections: $p(P_{a_k} | P_{a_n}, X, \beta)$
- Resample transitions: $p(T | P_{a_k}, X, \beta)$
- Resample observations: $p(\Omega | P_{a_n}, X, \beta, Y)$
- Resample state sequence: $p(X | P_{a_n}, P_{a_k}, \beta, T, \Omega, Y)$
- Add / delete factors: $p(P_{a_n} | P_{a_k}, X, \beta)$

Common to all DBN inference

Specific to iDBN

only 5% computational overhead!
Results: Toy Example

True Model

Likelihoods on Held-out Data

Factor Count

Test Likelihoods for Finite Models

iDBN Factor Count Histogram

2 Factors
Returning to Weather Example: Small Dataset

A model with just five locations quickly separates the east coast and the west coast data points.
Weather Example: Full Dataset

On the full dataset, we get regional factors with a general west-to-east pattern (the jet-stream).
Weather example: Full Dataset

Training and test performance (lower is better)
Zebra Finch Example

Given electrode readings (Smith et al., 2006), infer functional connectivity.

Results from Smith et al.

iDBN shared factors
Summary and Future Work

• Summary: We presented the iDBN prior, which allows us to infer the structure of general time-series models, showed it found interesting structure in weather, zebra-finetch data.

• What's next: improving inference/model knobs, application to patient monitoring in an eldercare setting.
Summary and Future Work

• Summary: We presented the iDBN prior, which allows us to infer the structure of general time-series models, showed it found interesting structure in weather, zebra-finch data.

• Future work: more control in the priors, improving inference, and adding control (e.g. for use in reinforcement learning).
# Results on Other Domains

<table>
<thead>
<tr>
<th>Domain</th>
<th>Negative Test Likelihood</th>
<th>Factors Discovered</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DBN</td>
<td>iFHMM</td>
</tr>
<tr>
<td>NW Star</td>
<td>174.0 ± 8.2</td>
<td>165.2 ± 3.0</td>
</tr>
<tr>
<td>NW Tree</td>
<td>255.6 ± 7.1</td>
<td>286.5 ± 2.9</td>
</tr>
<tr>
<td>NW Ring</td>
<td>181.7 ± 16.0</td>
<td>154.3 ± 1.6</td>
</tr>
<tr>
<td>Spike Train</td>
<td>142.4 ± 2.7</td>
<td>133.1 ± 2.1</td>
</tr>
<tr>
<td>Jungle</td>
<td>14.8 ± 1.4</td>
<td>13.9 ± 1.5</td>
</tr>
</tbody>
</table>
Growth of Hidden Factors and Observed Nodes