Fall 2010 Graduate Course on
Dynamic Learning

Chapter 10: Dynamic Hypernetworks

November 8, 2010

Byoung-Tak Zhang

School of Computer Science and Engineering &
Cognitive Science and Brain Science Programs
Seoul National University

http://bi.snu.ac.kr/~btzhang/

Overview

- Motivating Applications
  - Sentence completion
  - Music generation
- Generalizing HMMs
  - Higher-Order Transition Models
  - Higher-Order Observation Models
- Dynamic Hypernetworks
  - Representation
  - Learning
  - Inference
- References

Motivating Example: Sentence Completion

We ? ? a lot ? gifts.

⇨ **We don't have a lot of gifts.**

? still ? believe ? did this.

⇨ **I still can't believe you did this.**

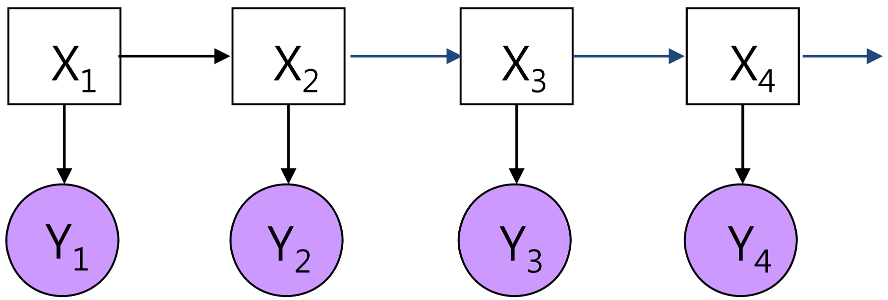
[Chan-Hoon Park et al., 2008]

Motivating Example: Music Composition

Scores generated by Evolutionary Hypernetworks that learned
American (A), Scottish (B), Korean Singer Kim (C), and Korean Singer Shin (D)
with the cue (left side of the bar in the middle)
from “Swanee River”, the famous American folk song

[Hyunwoo Kim et al., 2009]

Review: Hidden Markov Model (HMM)

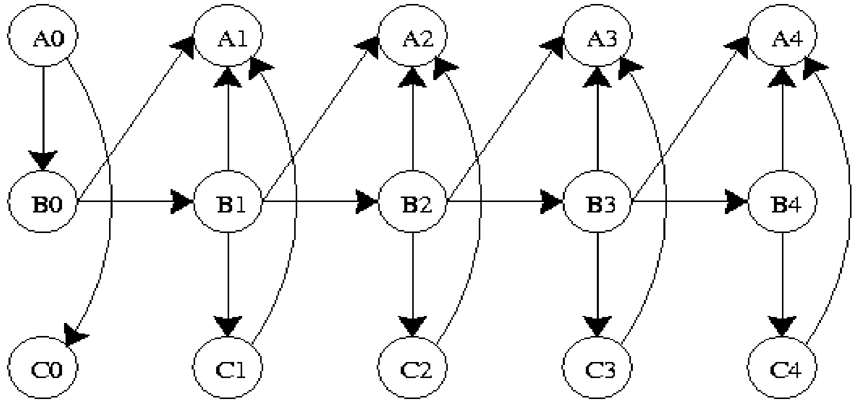


- One discrete hidden node and one discrete or continuous observed node per time slice. X: hidden variables, Y: observations
- Structures and parameters remain the same over time
- Three parameters in an HMM:
  - The initial state distribution $P(X_1)$
  - The transition model $P(X_t \mid X_{t-1}) = \{a_{ij}\}$
  - The observation model $P(Y_t \mid X_t) = \{b_{jyk}\}$
- HMM is the simplest DBN
  - A discrete state variable with arbitrary dynamics and arbitrary measurements

$$Y = (y_1 \dots y_T),\ \mu = (\{a_{ij}\}, \{b_{jy_k}\}, \pi)$$

$$P(Y \mid \mu) = \sum_{\{x_1 \dots x_T\}} \pi_{x_1} b_{x_1 y_1} \prod_{t=1}^{T-1} a_{x_t x_{t+1}} b_{x_{t+1} y_{t+1}}$$

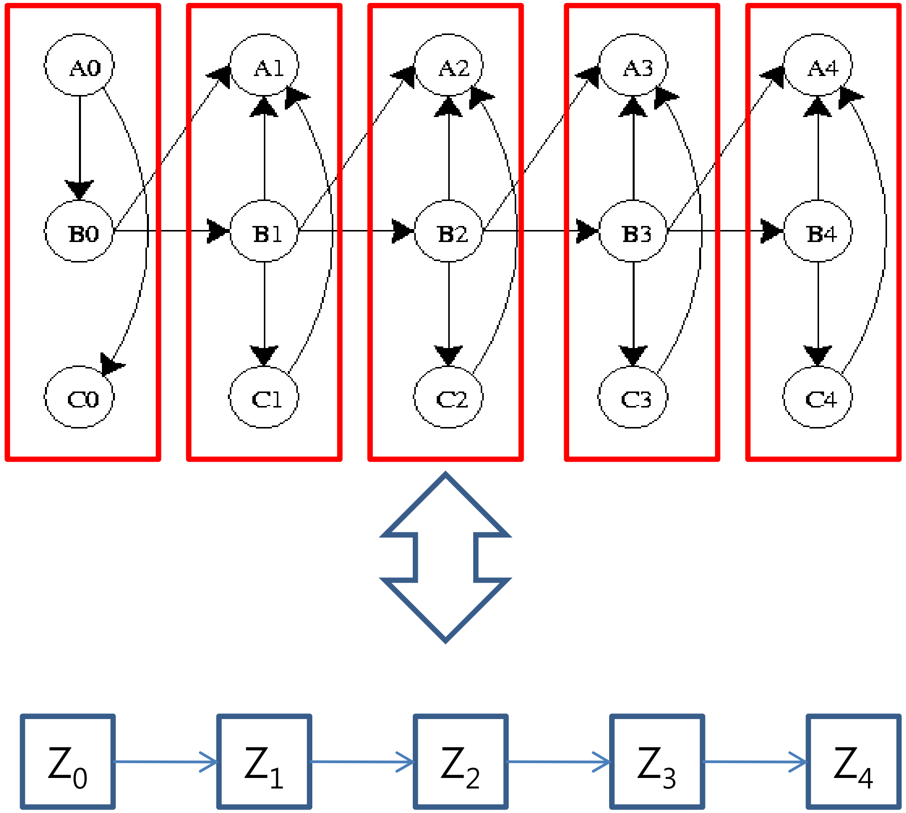
Review: Dynamic Bayesian Network (DBN)



- First-order Markov assumption: the parents of a node can only be in the same time slice or the previous time slice, X_{t-1} i.e., arcs do not across slices
- Inter-slice arcs are all from left to right, reflecting the arrow of time
- Intra-slice arcs can be arbitrary as long as the overall DBN is a DAG
- Time-invariant assumption: the parameters of the CPDs don't change over time
- The semantics of DBN can be defined by "unrolling" the 2TBN to T time slices
- The resulting joint probability distribution is then defined by

$$P(Z_{1:T}) = \prod_{t=1}^{T} \prod_{i=1}^{N} P(Z_t^i \mid \pi(Z_t^i))$$

DBN



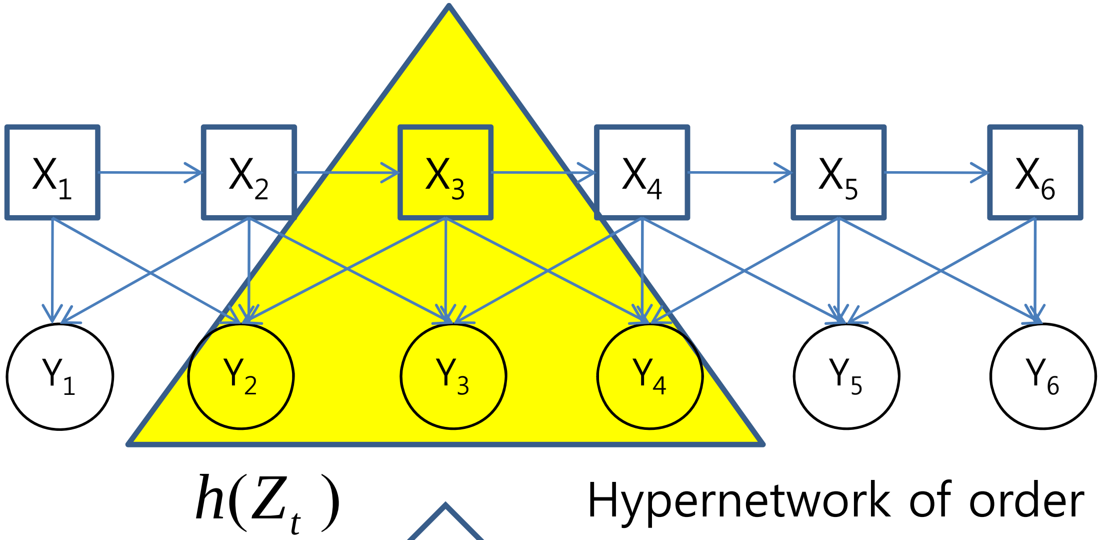
X = (A,B)
Y = C

Z = (X,Y)
= (A,B,C)

Topology of Z:
DAG structure

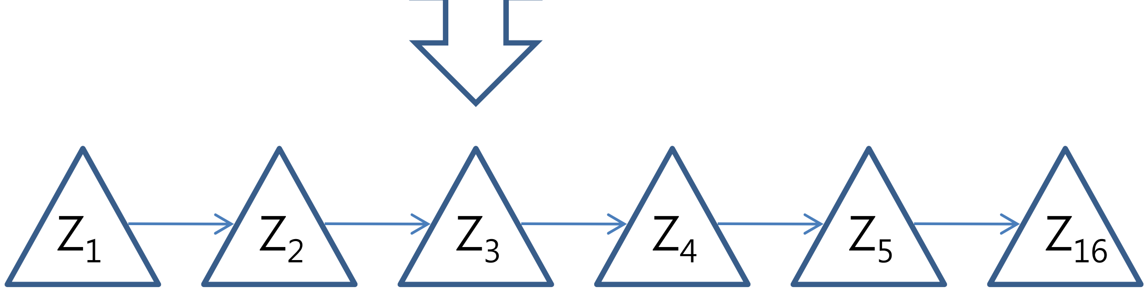
$$P(Z_{1:T}) = \prod_{t=1}^{T}\prod_{i=1}^{N} P(Z_t^i \mid \pi(Z_t^i))$$

Dynamic Hypernetwork (DHN)



General
Higher-Order
Observation Model

Hypernetwork of order k=3 hyperedges



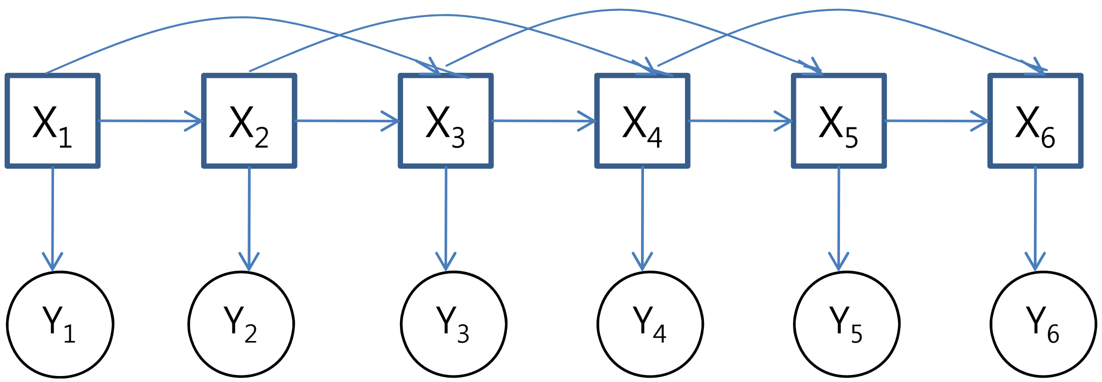
$Z_t = (X_t, Y_t^{(k)})$

Topology of Z:
hypergraph structure

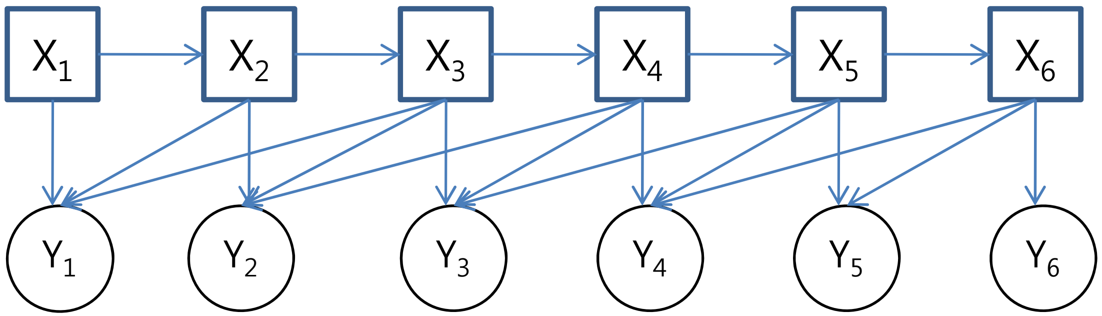
$$P(Z_{1:T}) = \prod_{t=1}^{T} \prod_{i=1}^{N} P(Z_t^i \mid h(Z_t^i))$$

Generalizing HMMs

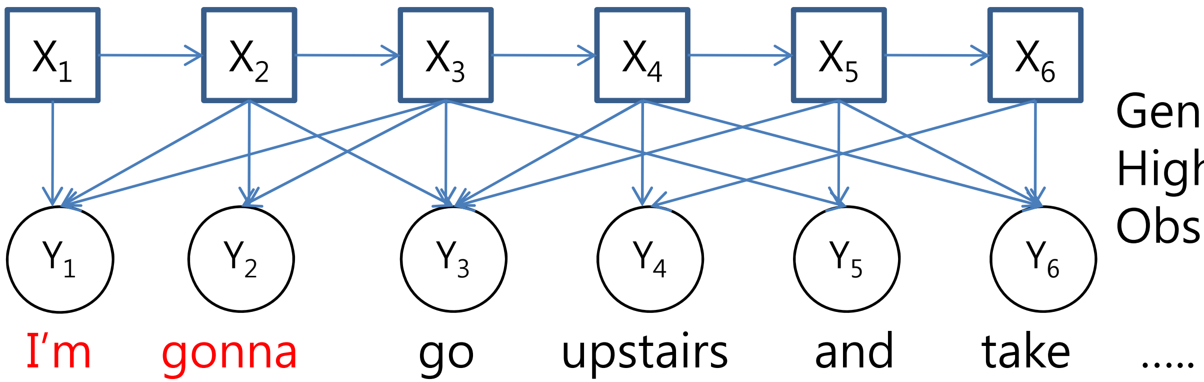
Exploring the Unexplored



Higher-Order State Model

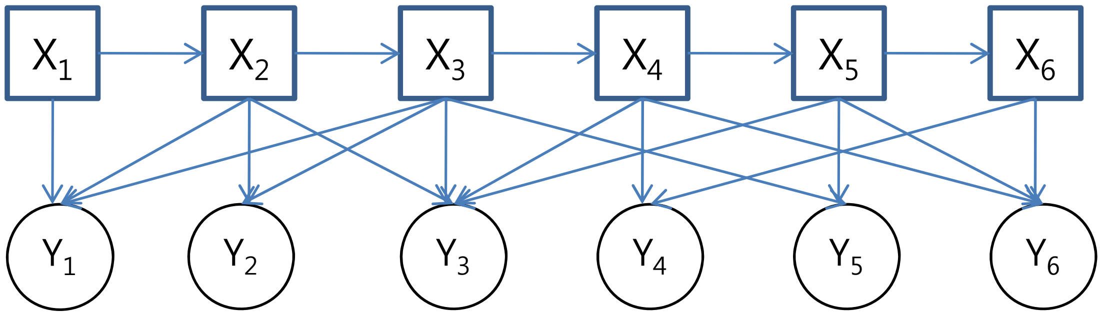


Higher-Order Observation Model

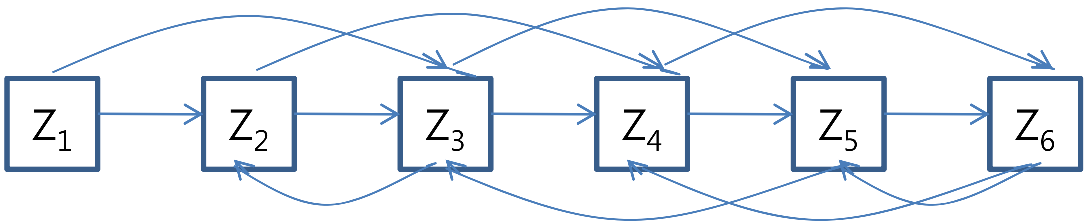


General Higher-Order Observation Model

General Higher-Order Model



General Higher-Order Observation Model

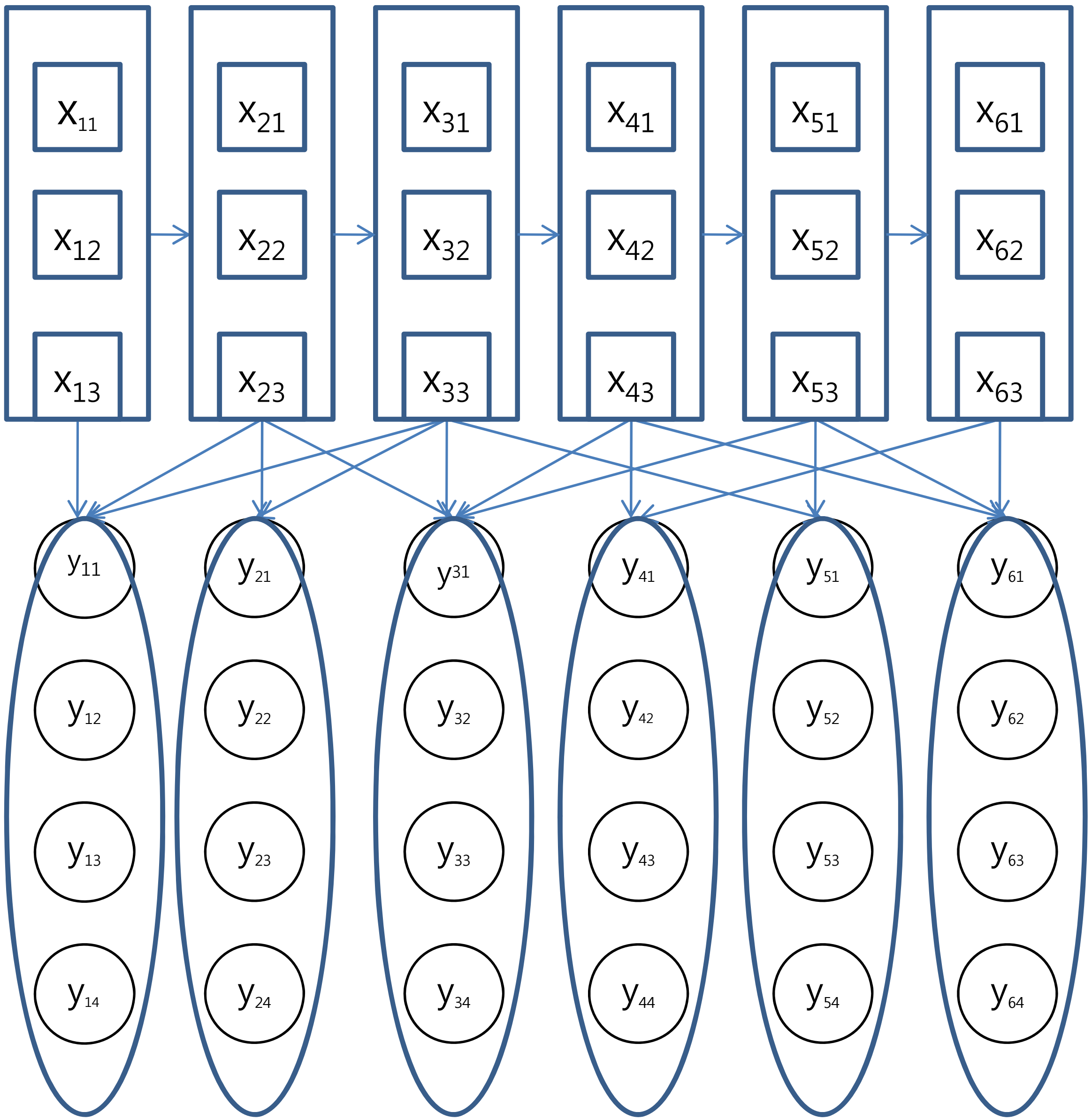


$Z = (X_t, Y_t^{(k)})$

Topology of Z: hypergraph structure

$$P(Z_{1:T}) = \prod_{t=1}^{T}\prod_{i=1}^{N} P(Z_t^i \mid \pi^{\rightarrow}(Z_t^i), \pi^{\leftarrow}(Z_t^i))$$

In Terms of CPTs

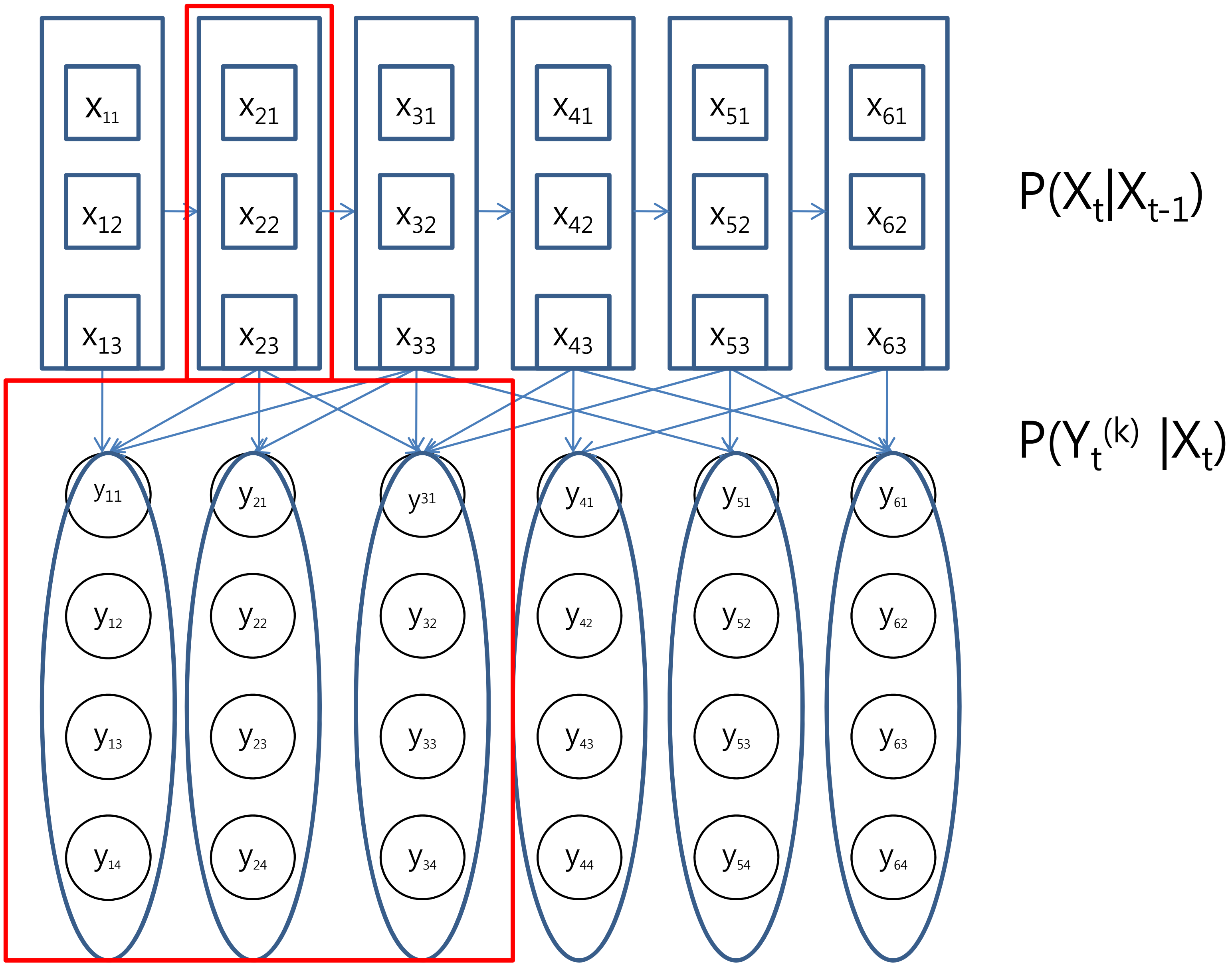


$P(X_t|X_{t-1})$

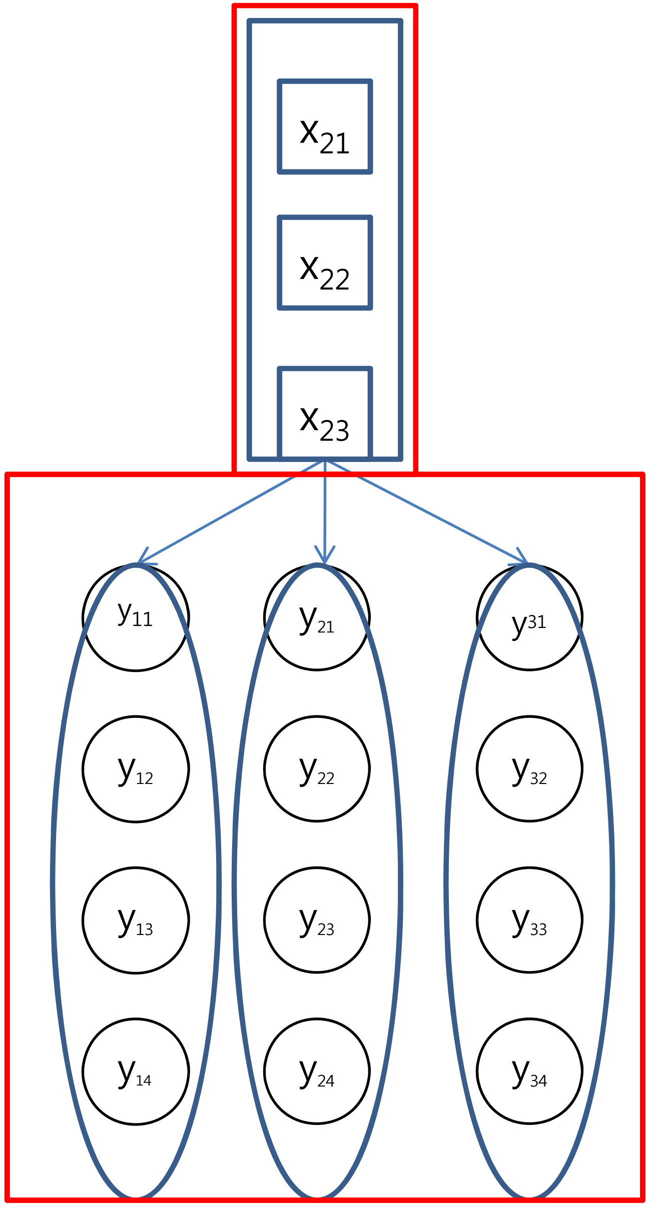
$P(Y_t^{(k)} |X_t)$

Dynamic Hypernetworks

One Slice of Time Series

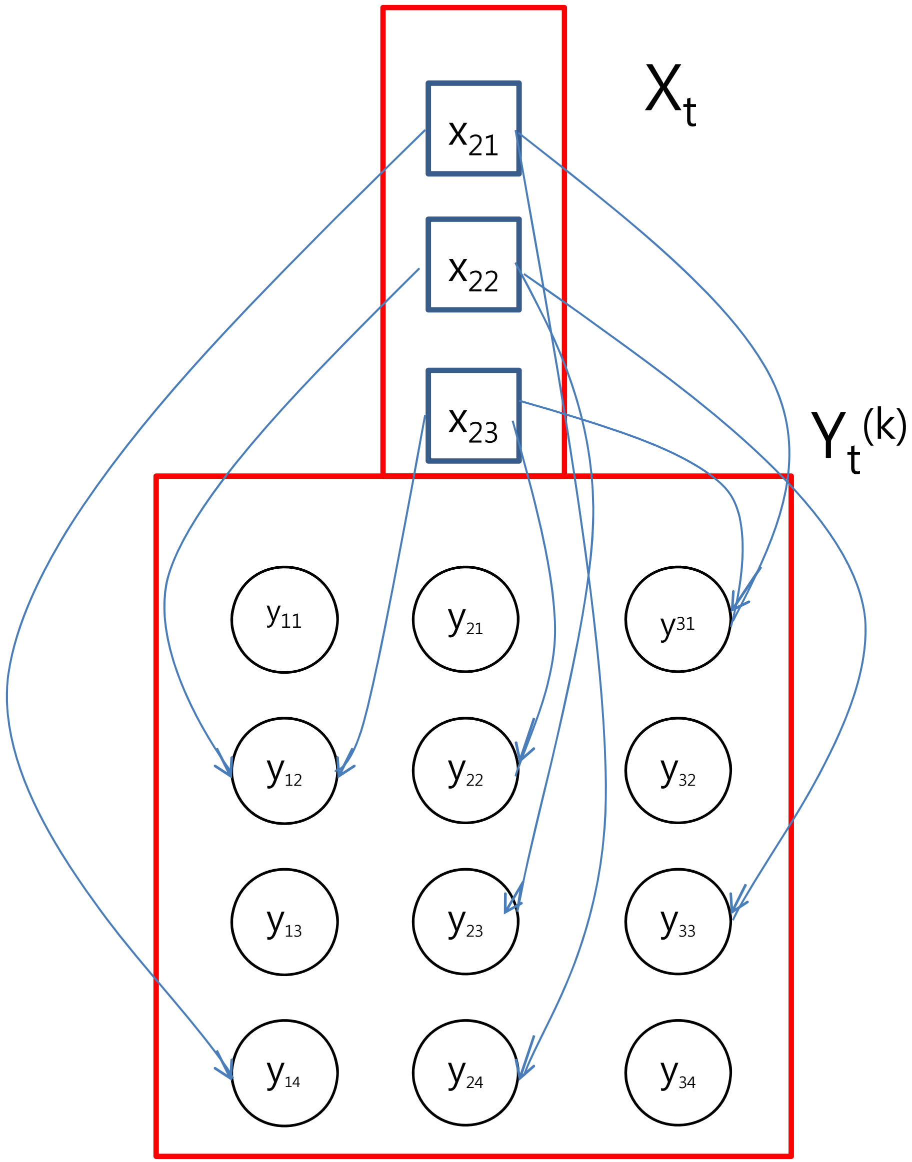


One Slice of Time Series



$P(Y_t^{(k)} | X_t)$

Hypernetwork Representation



$P(Y_t^{(k)} | X_t)$ represented as a "sparse" weighted hypergraph

Representing 3 states out of a total of 4 x 4 x 4 = 64 states

How to find these states?

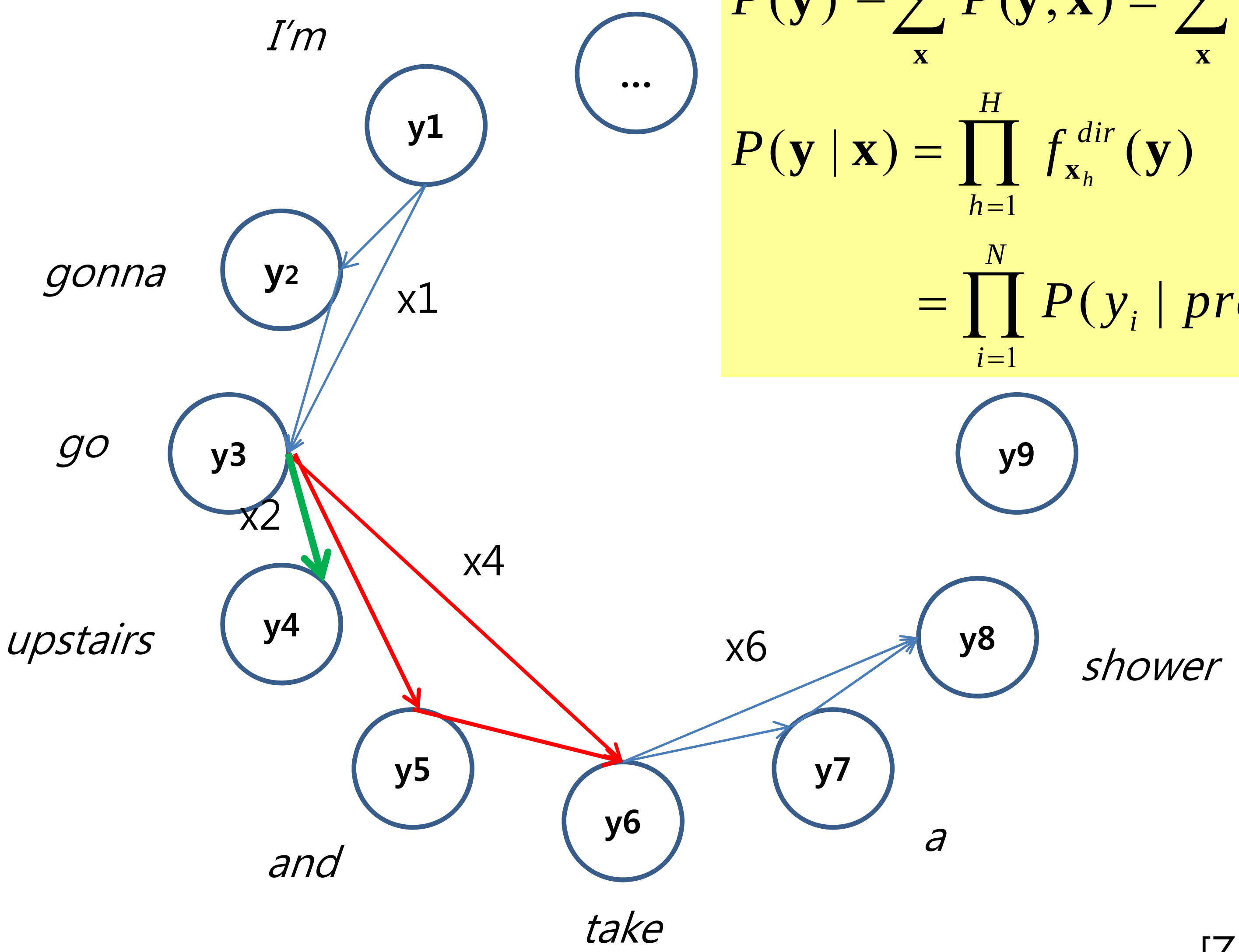
➔ Structure learning problem

Directed Hypernetwork for a Sequence

$$P(\mathbf{y}) = \sum_{\mathbf{x}} P(\mathbf{y},\mathbf{x}) = \sum_{\mathbf{x}} P(\mathbf{y}\mid\mathbf{x})P(\mathbf{x})$$

$$P(\mathbf{y}\mid\mathbf{x}) = \prod_{h=1}^{H} f_{\mathbf{x}_h}^{dir}(\mathbf{y})$$

$$= \prod_{i=1}^{N} P(y_i \mid pred_{\mathbf{x}}(y_i))$$



[Zhang, 2010, in prep

Learning and Inference in Dynamic Hypernetworks

Sequential Bayesian Learning of DHN: Basic Steps

Given $p(x_{t-1} \mid y_{1:t-1})$ ... prior (filtering) distribution

1. Prediction

$$p(x_t \mid y_{1:t-1}) = \int_{x_{t-1}} p(x_t \mid x_{t-1}) p(x_{t-1} \mid y_{1:t-1}) dx_{t-1} \qquad \text{(Eqn. 1)}$$

1.5 Generate

$$p(y_t \mid y_{1:t-1}) = \int_{x_{t-1}} p(y_t \mid x_{t-1}) p(x_{t-1} \mid y_{1:t-1}) dx_{t-1} \qquad \text{(Eqn. *)}$$

2. Update ... posterior distribution (after generating y_t)

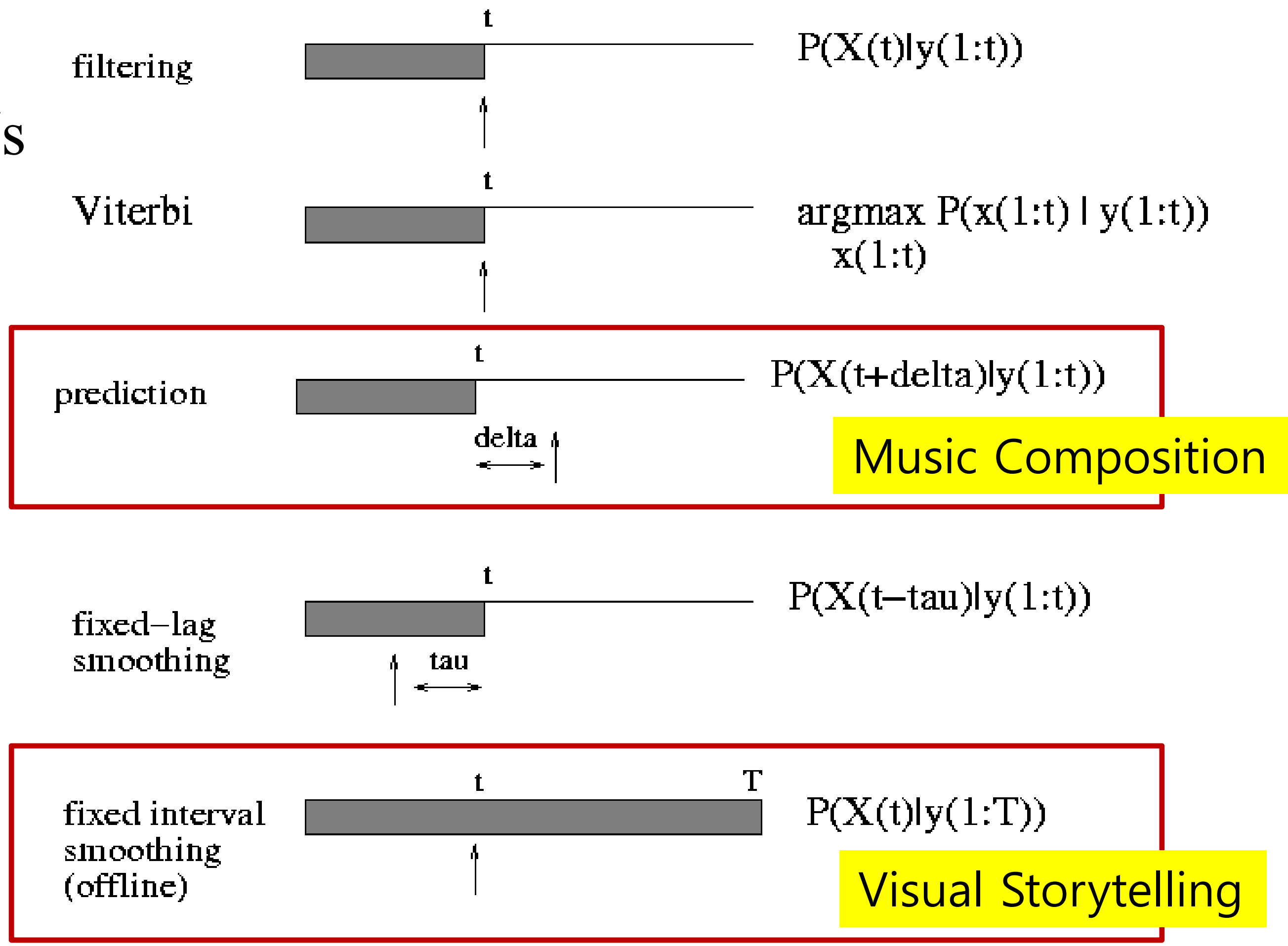
$$p(x_t \mid y_{1:t}) = \frac{p(y_t \mid x_t) p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})} \qquad \text{(Eqn. 2)}$$

where $p(y_t \mid y_{1:t-1}) = \int_{x_t} p(y_t \mid x_t) p(x_t \mid y_{1:t-1}) dx_t$

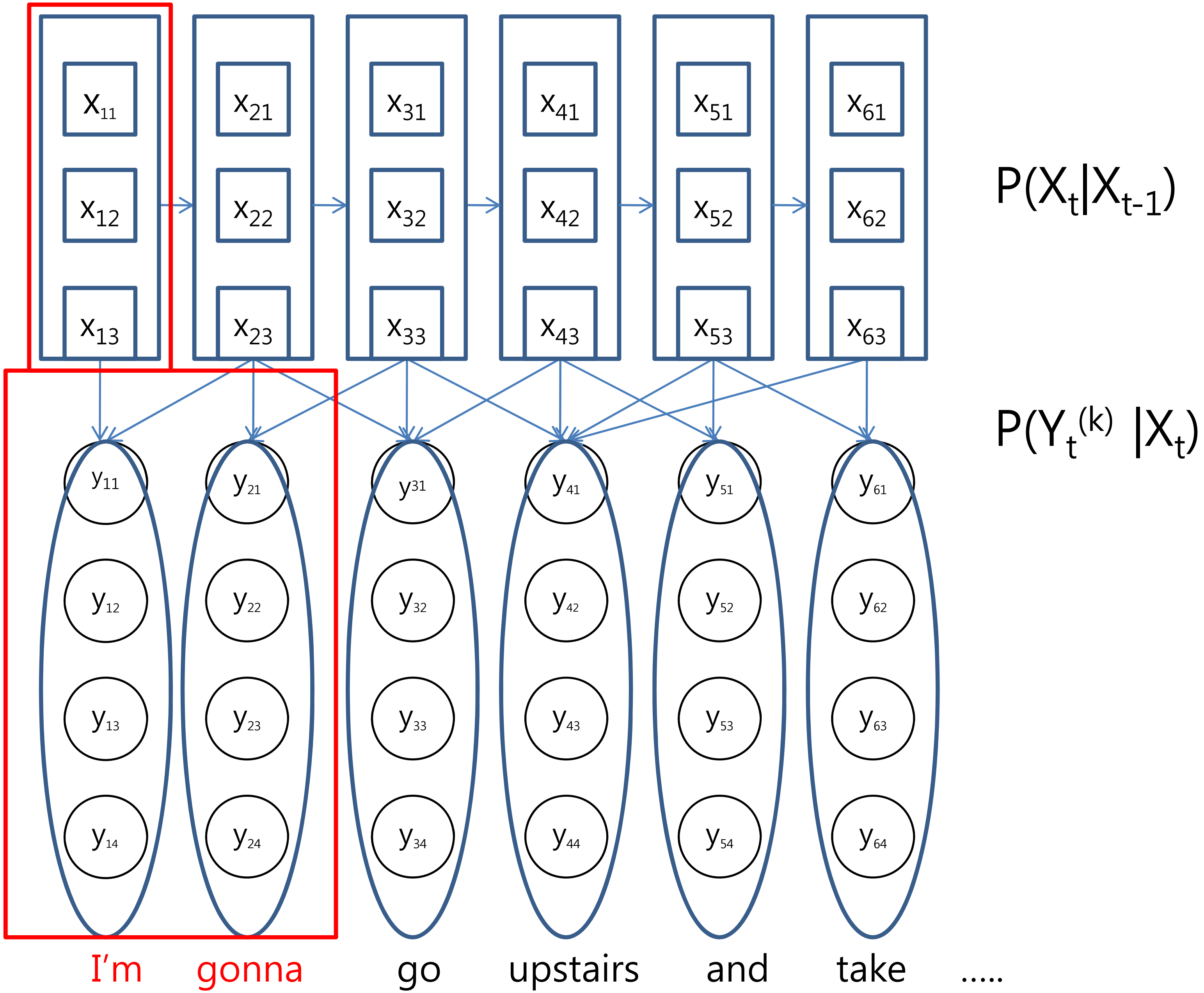
DHN: Inference

$$P(X_t^i \mid y_{1:r})$$

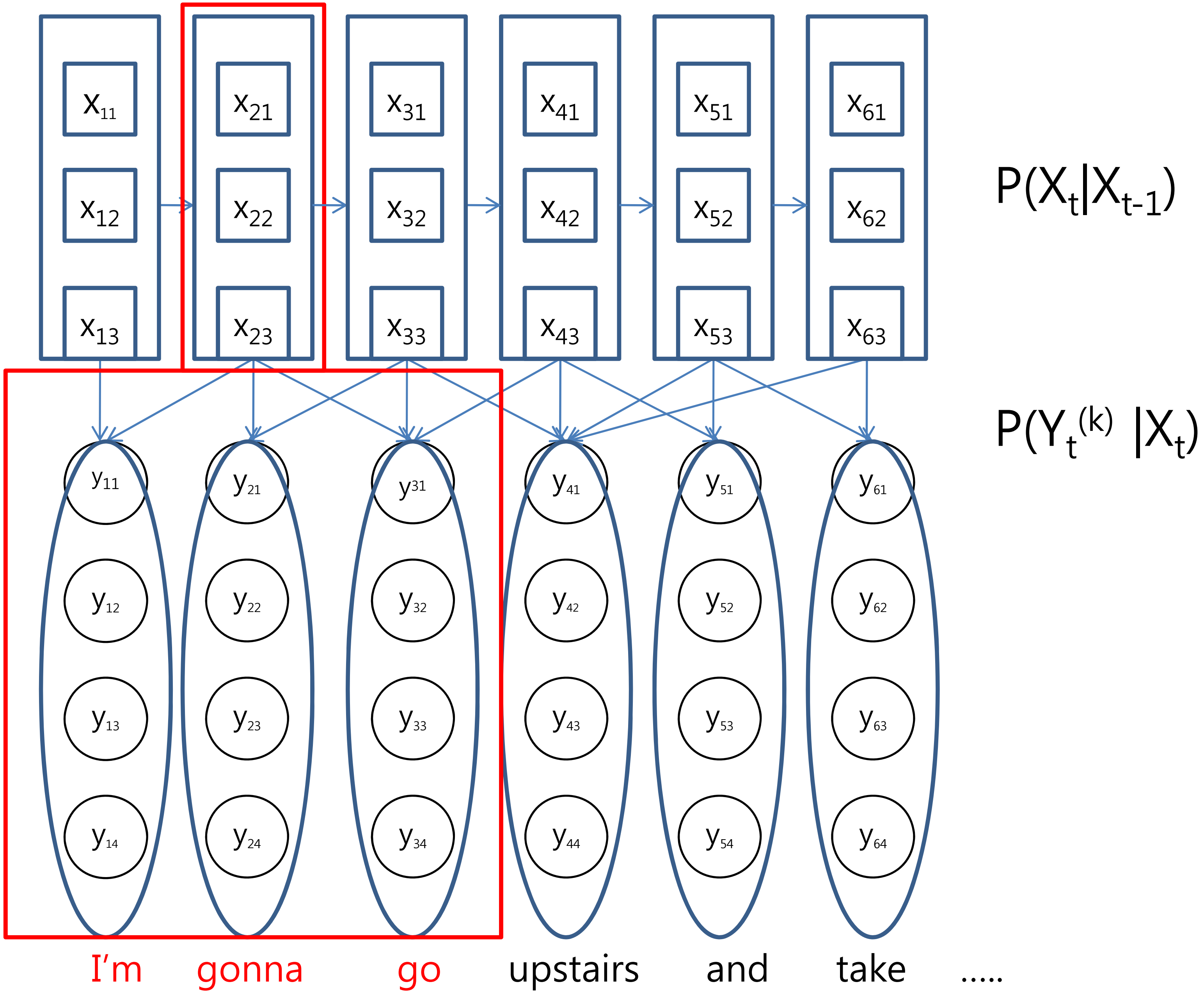
- The goal of inference in DHNs is to compute
  - Filtering: $r = t$
  - Smoothing: $r > t$
  - Prediction: $r < t$
  - Viterbi: MPE



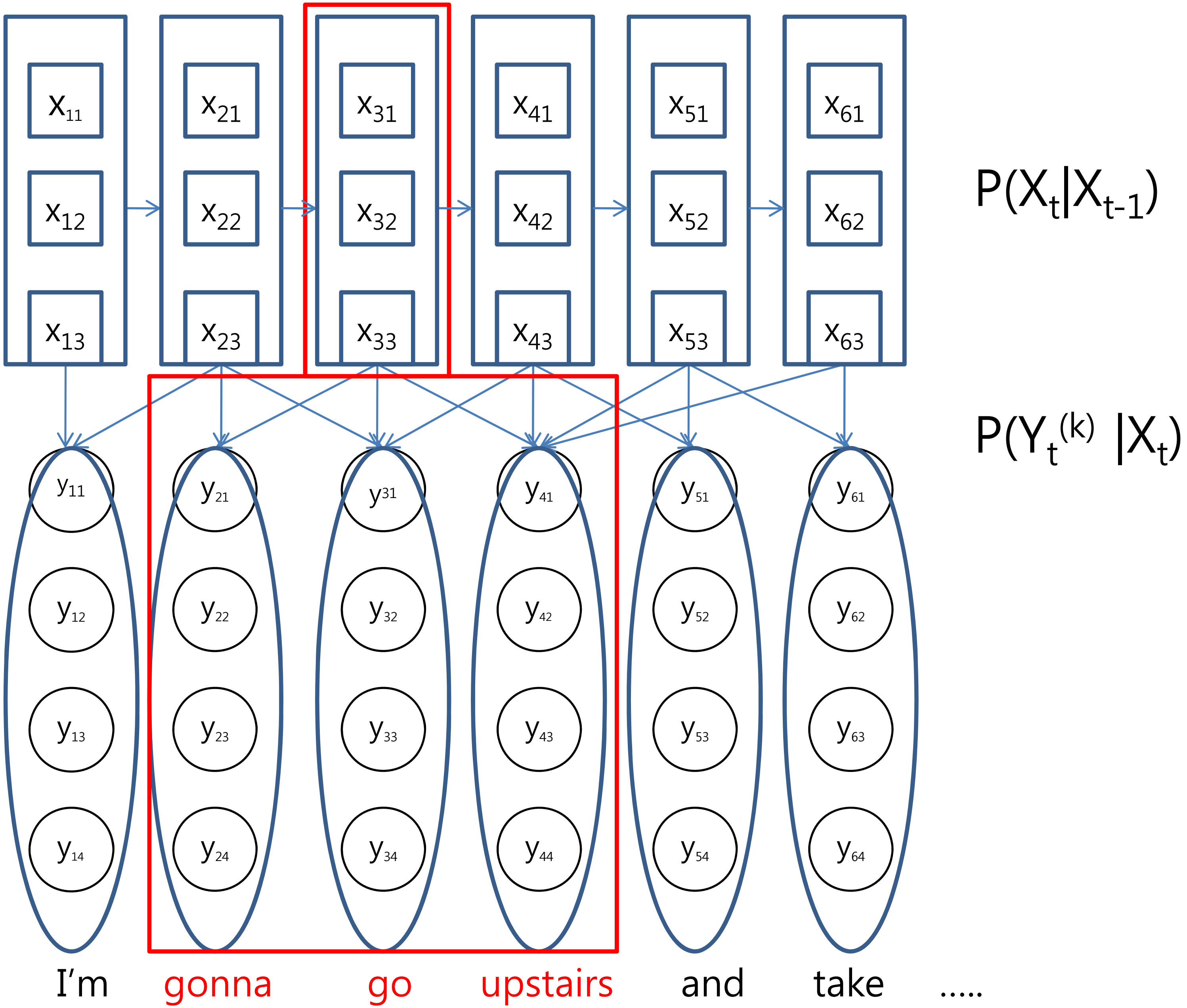
Learning the Observation Model



Learning the Observation Model



Learning the Observation Model



1. Offline Learning of a Dynamic Hypernetwork

$X = \{\}$

For each y in $D = Y = \{\text{sentences } y\}$

Let $y = (y_1, y_2, y_3, \ldots, y_N)$

For $t = 1$ to N

$x_t \leftarrow k$-gram fragments of y_t

// simplest case of n-gram model

$X = X + \{x_t\}$

For each y in $D = Y = \{\text{sentences } y\}$

$x^* = \text{argmax}_x P(y|x)$ // $x^* = (x_1^*, \ldots, x_N^*)$

For $t = 1$ to N

$w(x_t^*) \leftarrow (1 + \alpha)\, w(x_t^*)$

Learn Structure of $P(Y|X)$

Learn Parameters of $P(Y|X)$: copies of x_t

Learn Transition Model $P(X_t|X_{t-1})$

2. Online Learning of a Dynamic Hypernetwork

Let X = {hyperedges x_t}

For each y in $D = Y$ = {sentences y}

  Let $y = (y_1, y_2, y_3, \ldots, y_N)$

  For $t = 1$ to N

    $\{x_{t-1}\} = matchset(y_{t-1}, y_t \,\|\, x_{t-1}, x_t)$

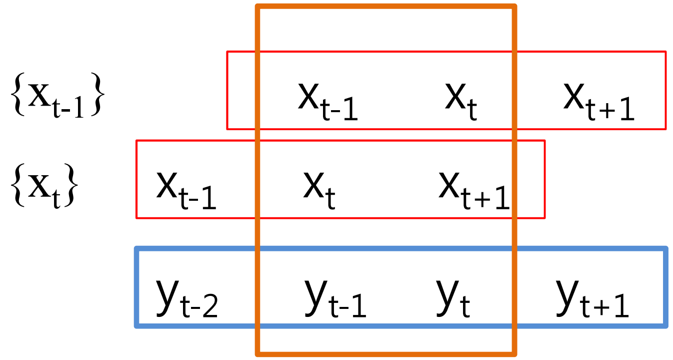
    $\{x_t\} = matchset(y_{t-1}, y_t \,\|\, x_t, x_{t+1})$

    $\{x_m\} = \{x_{t-1}\} + \{x_t\}$

    $x^* = \text{argmax}_{x \text{ in } \{xm\}} P(y_{t+1} | x)$

    $w(x_m) \leftarrow (1 + \alpha) w(x_m)$ for $x_m \in \{x_m\}$

    $w(x^*) \leftarrow (1 + \alpha)\, w(x^*)$



$$\alpha_i(t) = P(o_1 \ldots o_t, x_t = i \mid \mu)$$

$$\alpha_j(t+1) = \sum_{i=1 \ldots N} \alpha_i(t) a_{ij} b_{jo_{t+1}}$$

Learn $P(X_t | X_{t-1})$

Learn $P(Y|X)$ (Refine)

3. Inference: Sequential Prediction (Storytelling)

Let X = {hyperedges x_t}
Let y = $(y_1, y_2, \ldots, y_q, \ldots, y_N)$ // $q < N$
For t = 1 to q-1
  {x_{t-1}} = *matchset*($y_{t-1}, y_t \,\|\, x_{t-1}, x_t$)
  {x_t} = *matchset*($y_{t-1}, y_t \,\|\, x_t, x_{t+1}$)
  {x_m} = {x_{t-1}} + {x_t}
y_q = argmax $_y$ P(y|{x_m})
y = (y, y_{q+1})
For t = q+1 to N
  {x_{t-1}} = *matchset*($y_{t-1}, y_t \,\|\, x_{t-1}, x_t$)
  {x_t} = *matchset*($y_{t-1}, y_t \,\|\, x_t, x_{t+1}$)
  {x_m} = {x_{t-1}} + {x_t}
  y_{q+1} = argmax $_y$ P(y|{x_m})
  y = (y, y_{q+1})
Return(y)

$$\delta_j(t) = \max_{x_1 \ldots x_{t-1}} P(x_1 \ldots x_{t-1}, y_1 \ldots y_{t-1}, x_t = j, y_t)$$

$$\delta_j(t+1) = \max_i \delta_i(t) a_{ij} b_{j y_{t+1}}$$

4. Inference: Smoothing (Completion)

Let X = {hyperedges x_t}

Let $y = (y_1, y_2, y_3, \ldots, y_q)$ // $q < N$

For t = 1 to q

$\{x_{t-1}\} = matchset(y_{t-1}, y_t \| x_{t-1}, x_t)$

$\{x_t\} = matchset(y_{t-1}, y_t \| x_t, x_{t+1})$

$\{x_m\} = \{x_{t-1}\} + \{x_t\}$

$y_{q+1} = \text{argmax}_y\, P(y|\{x_m\})$

$y = (y, y_{q+1})$

For $t = q+1$ to N

$\{x_{t-1}\} = matchset(y_{t-1}, y_t \| x_{t-1}, x_t)$

$\{x_t\} = matchset(y_{t-1}, y_t \| x_t, x_{t+1})$

$\{x_m\} = \{x_{t-1}\} + \{x_t\}$

$y_{t+1} = \text{argmax}_y\, P(y|\{x_m\})$

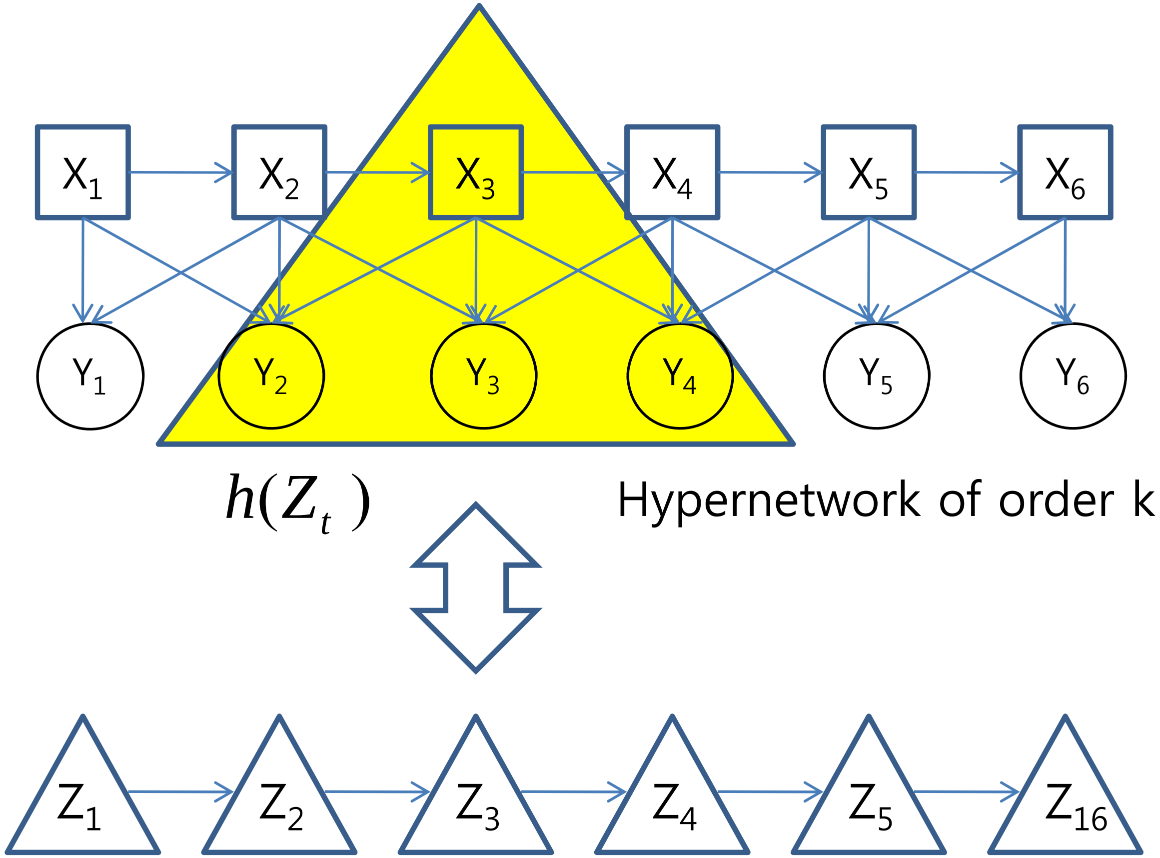
$y = (y, y_{t+1})$

Return(y)

$$\delta_j(t) = \max_{x_1 \ldots x_{t-1}} P(x_1 \ldots x_{t-1}, y_1 \ldots y_{t-1}, x_t = j, y_t)$$

$$\delta_j(t+1) = \max_i \delta_i(t) a_{ij} b_{j y_{t+1}}$$

Summary: Dynamic Hypernetwork (DHN)



- First-Order Markovian Transition Model
- General Higher-Order Observation Model

$Z_t = (X_t, Y_t^{(k)})$

Topology of Z: hypergraph structure

$$P(Z_{1:T}) = \prod_{t=1}^{T} \prod_{i=1}^{N} P(Z_t^i \mid h(Z_t^i))$$

Summary: DHN vs. DBN & HMM

- DHNs are higher-order Markov models.
  - Like DBNs and HMMs, DHNs employ a first-order Markovian state-transition model.
  - Unlike DBNs and HMMs, DHNs employ a higher-order Markovian observation model.
- DHNs learn only from left to right and does not learn right to left.
  - Sequence length T is variable and there is no goal state.
- DHNs learn sparse tables of the transition and observation probabilities.
  - DBN and HMM maintain the full matrix of conditional probability tables.
- DHNs build a complex state model which is learned.
  - DBNs also employ a complex state model, but the model is typically designed
- DHNs use hypergraph structure for a time-slice model.
  - DBNs use DAG structure for a time-slice model.
- DHNs employ a varying number of state variables.
  - HMM uses a single state variable with discrete values and DBNs use a fixed number of state variables with DAG dependency structure.
- DHNs can be learned offline and online. Online sequential learning DHNs are useful for forecasting and smoothing.
  - Forecasting → Storytelling, Smoothing → Completion

References

- **Hypernetworks: A molecular evolutionary architecture for cognitive learning and memory**, B.-T. Zhang, *IEEE Computational Intelligence Magazine*, **3**(3):49-63, 2008.
- **Layered hypernetwork models for cross-modal associative text and image keyword generation in multimodal information retrieval**, J.-W. Ha, B.-H. Kim, B. Lee, and B.-T. Zhang, *Proceedings of the Eleventh Pacific Rim International Conference on AI (PRICAI2010)*, *Lecture Notes in Artificial Intelligence*, **6230**:76-87, 2010.
- **Visual query expansion via incremental hypernetwork models of image and text**, M.-O. Heo, M.-G. Kang, and B.-T. Zhang, *Proceedings of the Eleventh Pacific Rim International Conference on AI (PRICAI2010)*, *Lecture Notes in Artificial Intelligence*, **6230**:88-99, 2010.
- **Evolutionary layered hypernetworks for identifying microRNA-mRNA regulatory modules**, S.-J. Kim, J.-W. Ha, B. Lee, and B.-T. Zhang, *IEEE World Congress Computational Intelligence (WCCI-CEC 2010)*, pp. 2299-2306, 2010.
- **Evolving hypernetwork models of binary time series for forecasting price movements on stock markets**, E. Bautu, S. Kim, A. Bautu, H. Luchian, and B.-T. Zhang, *IEEE Congress on Evolutionary Computation (CEC 2009)*, pp.166-173, 2009.
- **An in vitro DNA hypernetwork for digit pattern recognition**, K.-A. Yang, I.-H. Lee, and B.-T. Zhang, *Preliminary Proceedings of the 15th International Meeting on DNA Computing (DNA15)*, 2009
- **Evolutionary hypernetworks for learning to generate music from examples**, H.-W. Kim, B.-H. Kim, and B.-T. Zhang, *IEEE International Confernece on Fuzzy Systems (Fuzz IEEE 2009)*, pp. 47-52, 2009.
- **Text-to-image cross-modal retrieval of magazine articles based on higher-order pattern recall by hypernetworks**, J.-W. Ha, B.-H. Kim, H.-W. Kim, W.C. Yoon, J.-H. Eom, and B.-T. Zhang, *The 10th International Symposium on Advanced Intelligent Systems (ISIS 2009)*, pp. 274-277, 2009.
- **Self-assemblying hypernetworks for cognitive learning of linguistic memory**, B.-T. Zhang and C.-H. Park, *CESSE2008* (WASET), Vol. 27, pp.134-138, 2008.
- **Random hypergraph models of learning and memory in biomolecular networks: shorter-term adaptability vs. longer-term persistency**, B.-T. Zhang, *The First IEEE Symp. on Foundations of Computational Intelligence (FOCI '07)*, pp. 344-349, 2007.
- **DNA hypernetworks for information storage and retrieval**, B.-T. Zhang and J.-K. Kim, *Preliminary Proceedings of the Twelfth International Meeting on DNA Computing (DNA 12)*, pp. 283-292, 2006.