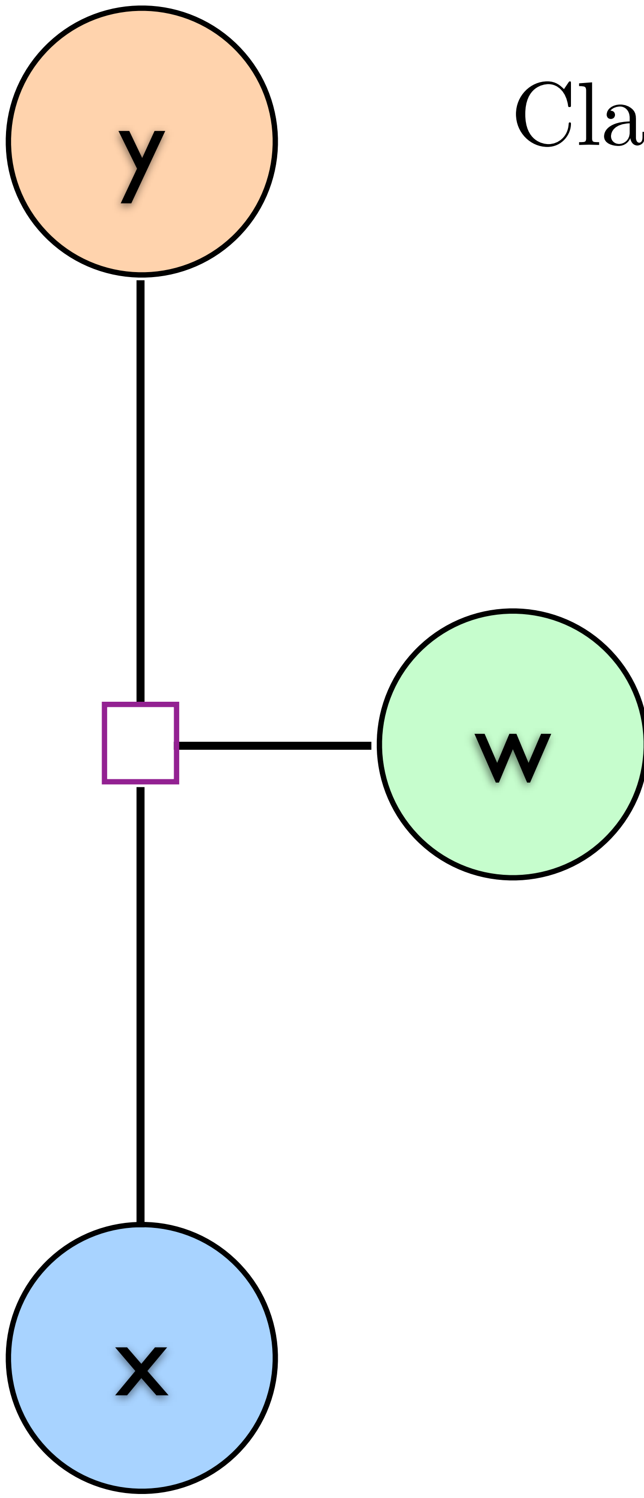
Conditional Random Fields with Latent Variables

Mark Schmidt, June 2014
(e-mail me for references)

Outline

- Overview of General Conditional Random Fields
- Conditional Random Fields with Latent Variables

Binary Logistic Regression

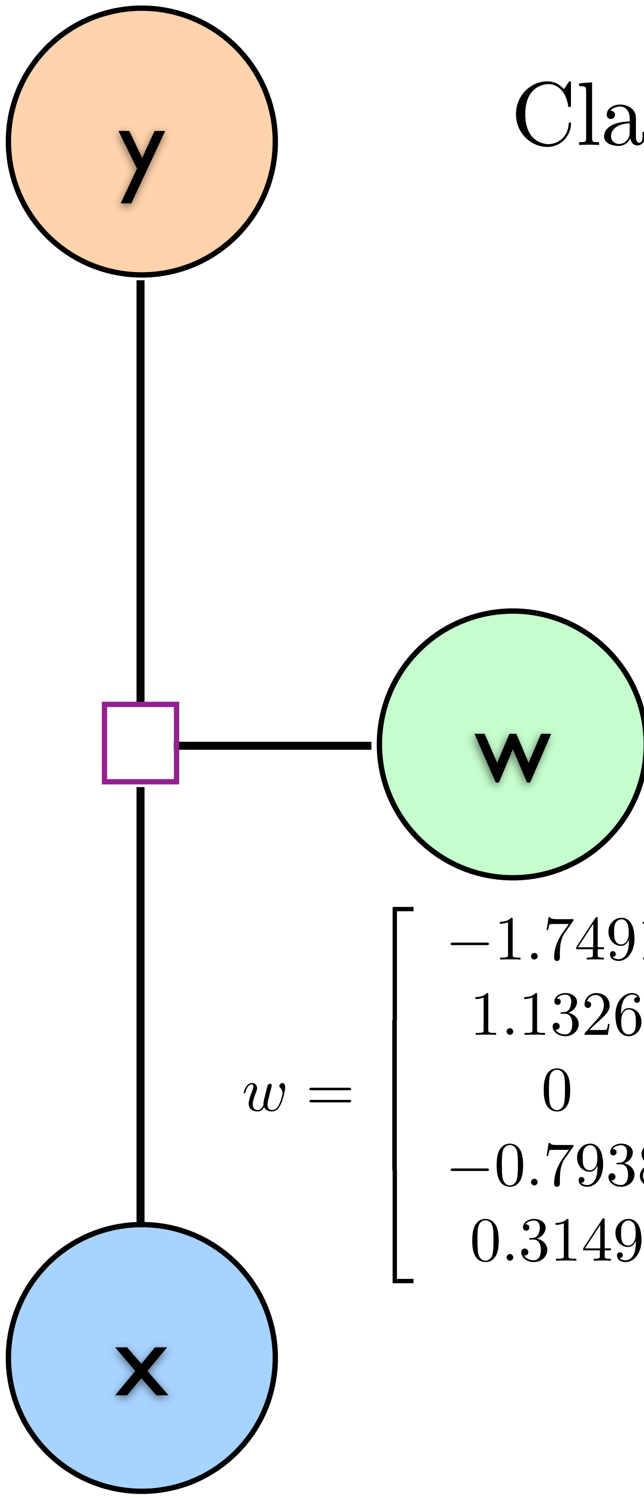


Classify using $y = \text{sign}(w^T x)$.

Binary Logistic Regression

Classify using $y = \text{sign}(w^T x)$.

$y = [-1]$



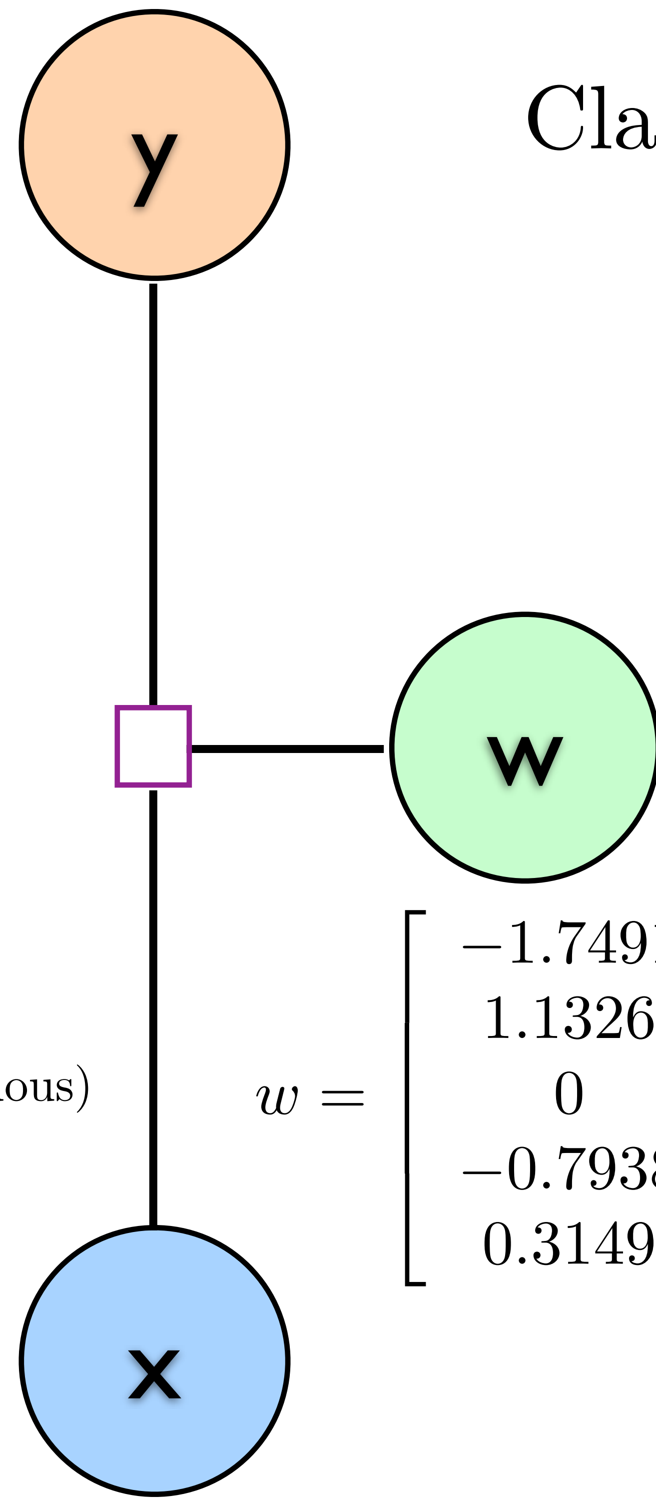
$$w = \begin{bmatrix} -1.7491 \\ 1.1326 \\ 0 \\ -0.7938 \\ 0.3149 \end{bmatrix}$$

$$x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Binary Logistic Regression

Classify using $y = \text{sign}(w^T x)$.

$y = [-1]$



$$w = \begin{bmatrix} -1.7491 \\ 1.1326 \\ 0 \\ -0.7938 \\ 0.3149 \end{bmatrix}$$

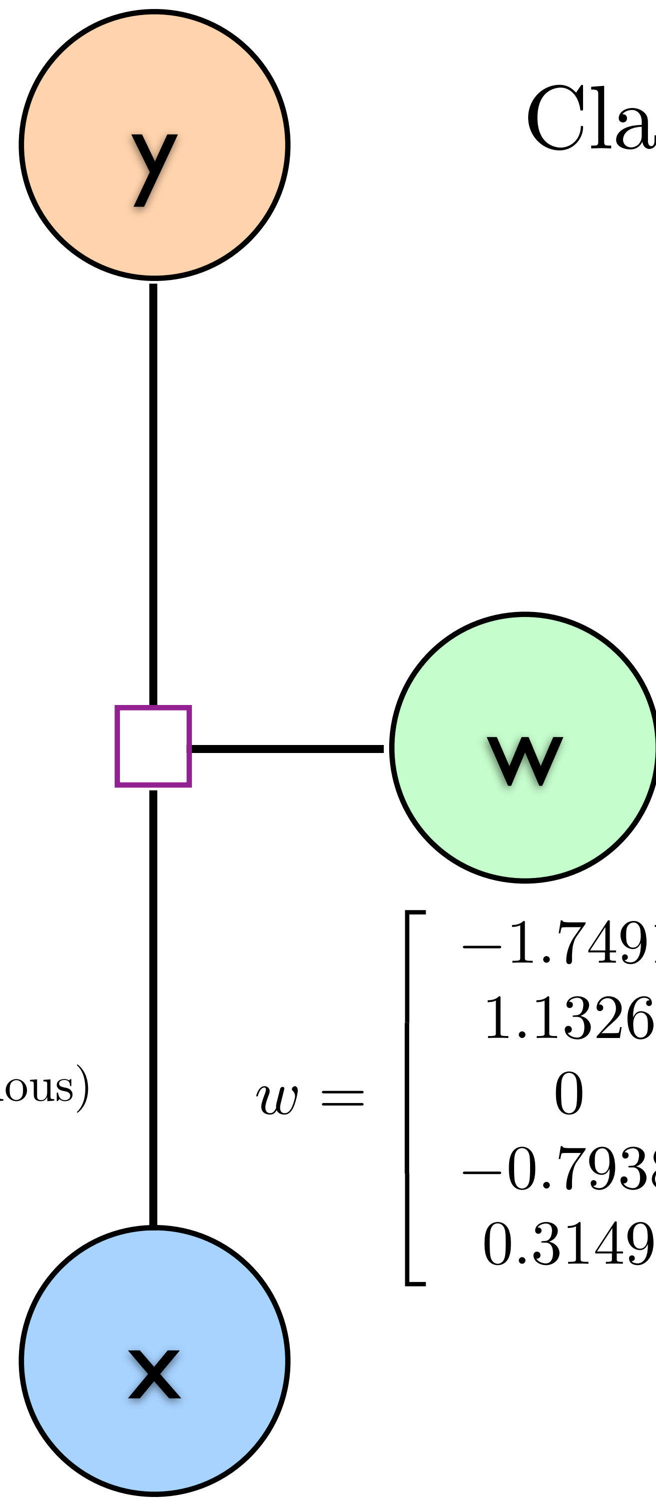
(x_i can be continuous)

$$x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Binary Logistic Regression

Classify using $y = \text{sign}(w^T x)$.

$y = [-1]$



$$p(y = 1 | x, w) = \frac{\exp(y w^T x)}{\exp(w^T x) + \exp(-w^T x)}$$
$$= \frac{\exp(y w^T x)}{\sum_{y'} \exp(y w^T x)}$$

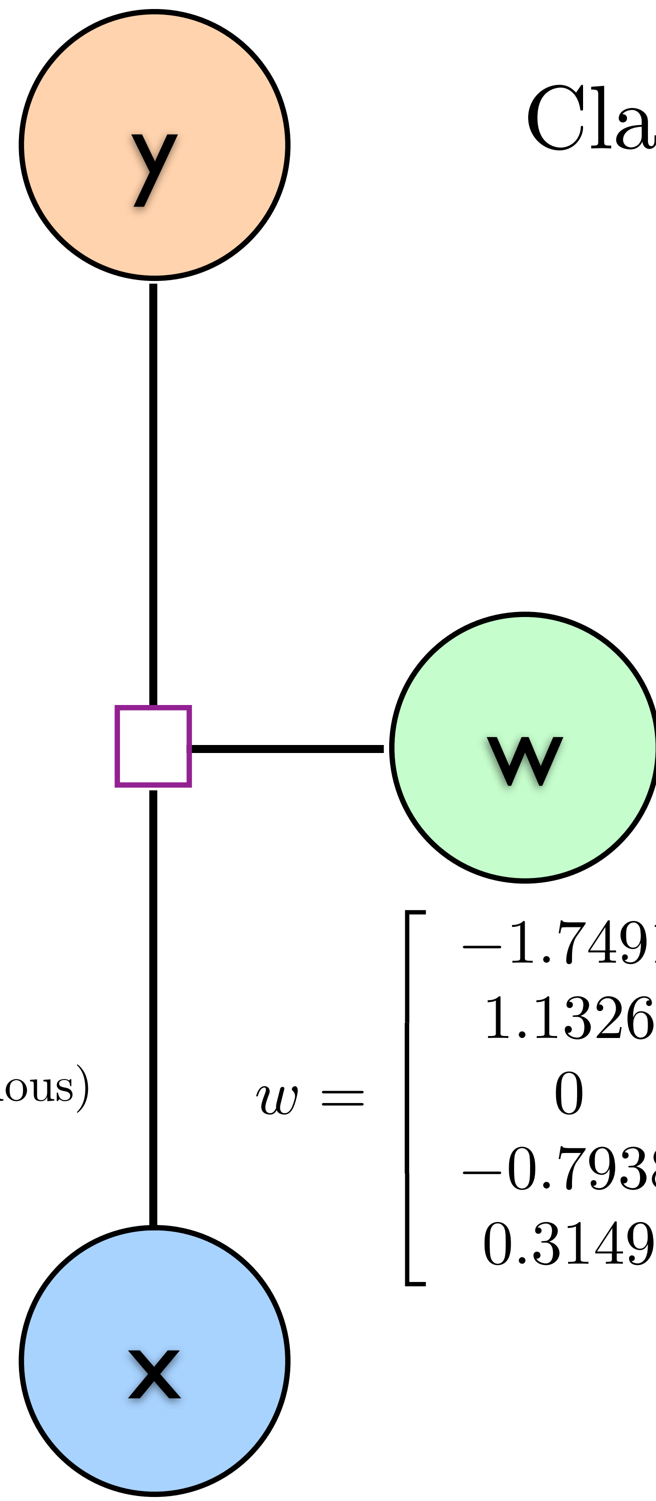
$$w = \begin{bmatrix} -1.7491 \\ 1.1326 \\ 0 \\ -0.7938 \\ 0.3149 \end{bmatrix}$$

(x_i can be continuous)

$$x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Binary Logistic Regression

$y = [-1]$



Classify using $y = \mathrm{sign}(w^T x)$.

$$p(y = 1 | x, w) = \frac{\exp(y w^T x)}{\exp(w^T x) + \exp(-w^T x)}$$
$$= \frac{\exp(y w^T x)}{\sum_{y'} \exp(y w^T x)}$$

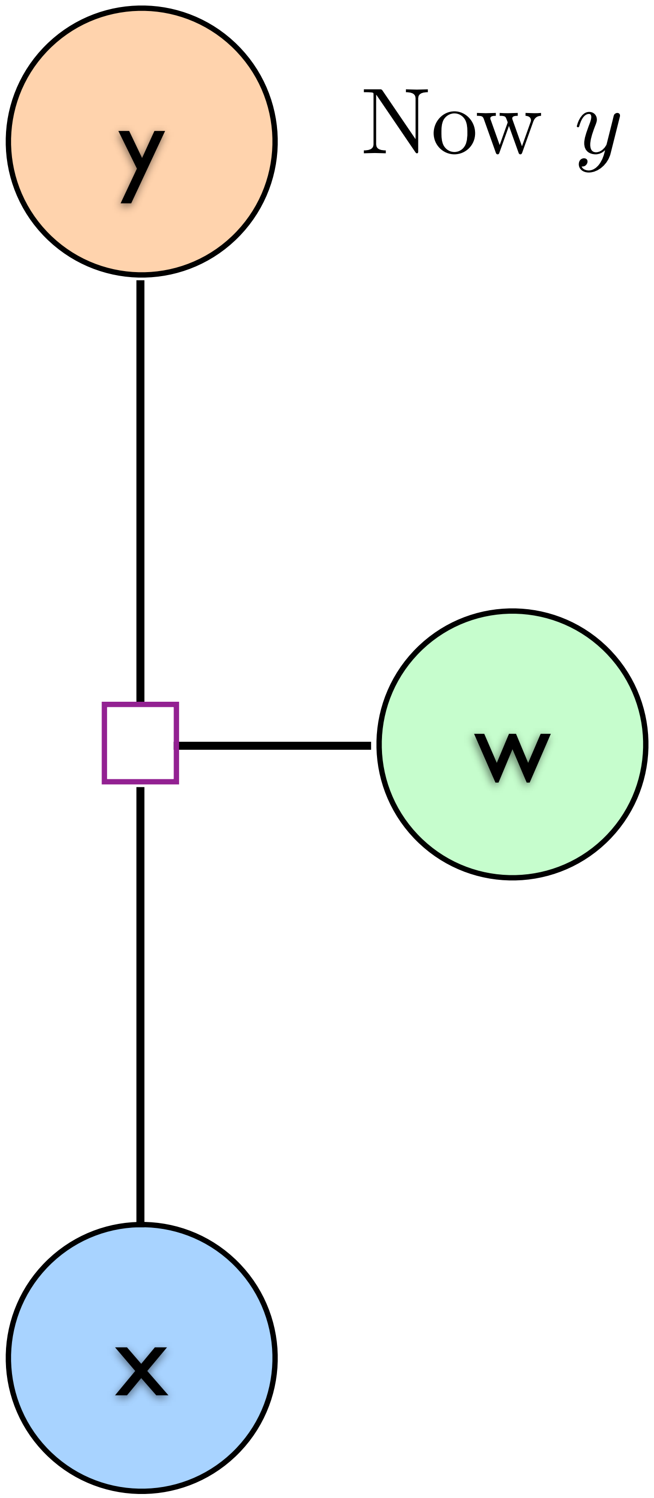
$$w = \begin{bmatrix} -1.7491 \\ 1.1326 \\ 0 \\ -0.7938 \\ 0.3149 \end{bmatrix}$$

(x_i can be continuous)

$$x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$$\boxed{p(y = s | x, w) \propto \exp(s w^T x)}$$

Multi-Class Logistic Regression

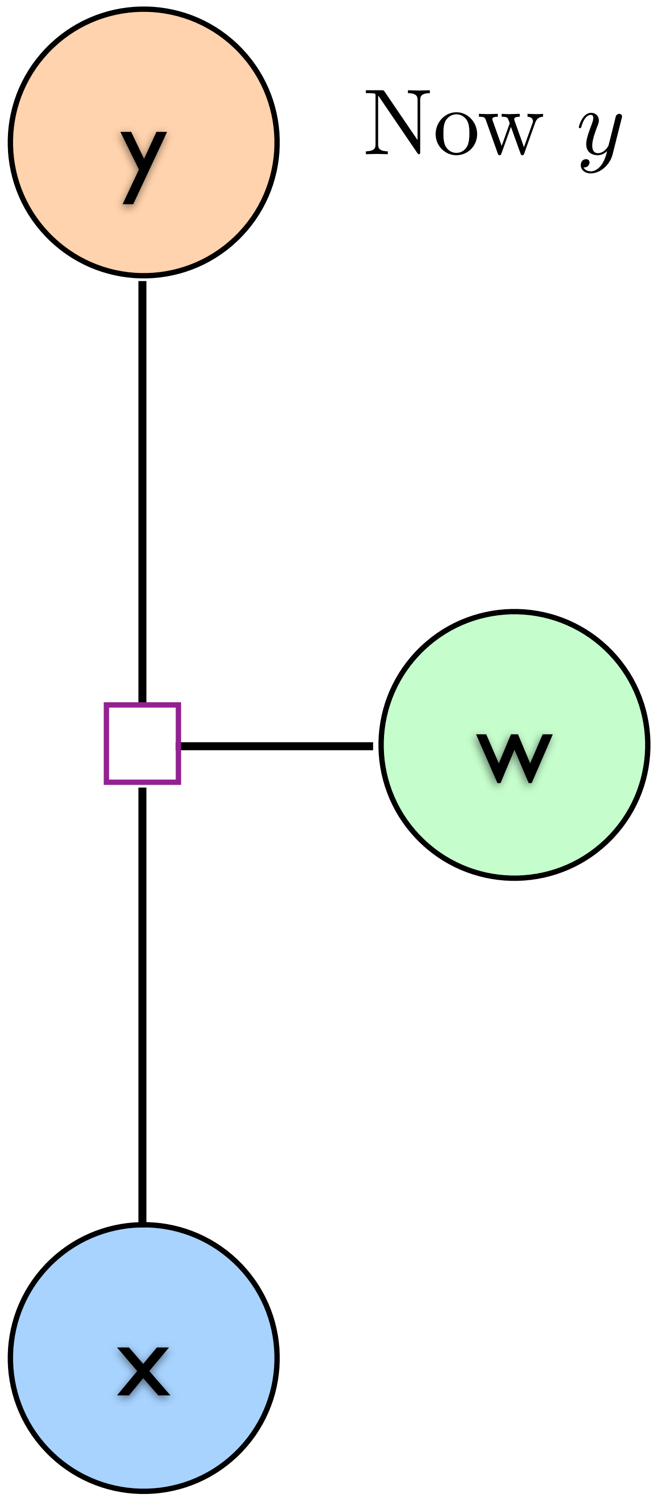


Now $y \in \{1, 2, 3, \ldots, S\}$.

Multi-Class Logistic Regression

$y = [3]$

Now $y \in \{1, 2, 3, \ldots, S\}$.



$$w = \begin{bmatrix} -1.7491 & 1.7411 & 0.8106 \\ 1.1326 & 0.4868 & 0.6985 \\ 0 & 1.0488 & -0.4016 \\ -0.7938 & 1.4886 & 1.2688 \\ 0.3149 & 1.2705 & -0.7836 \end{bmatrix}$$

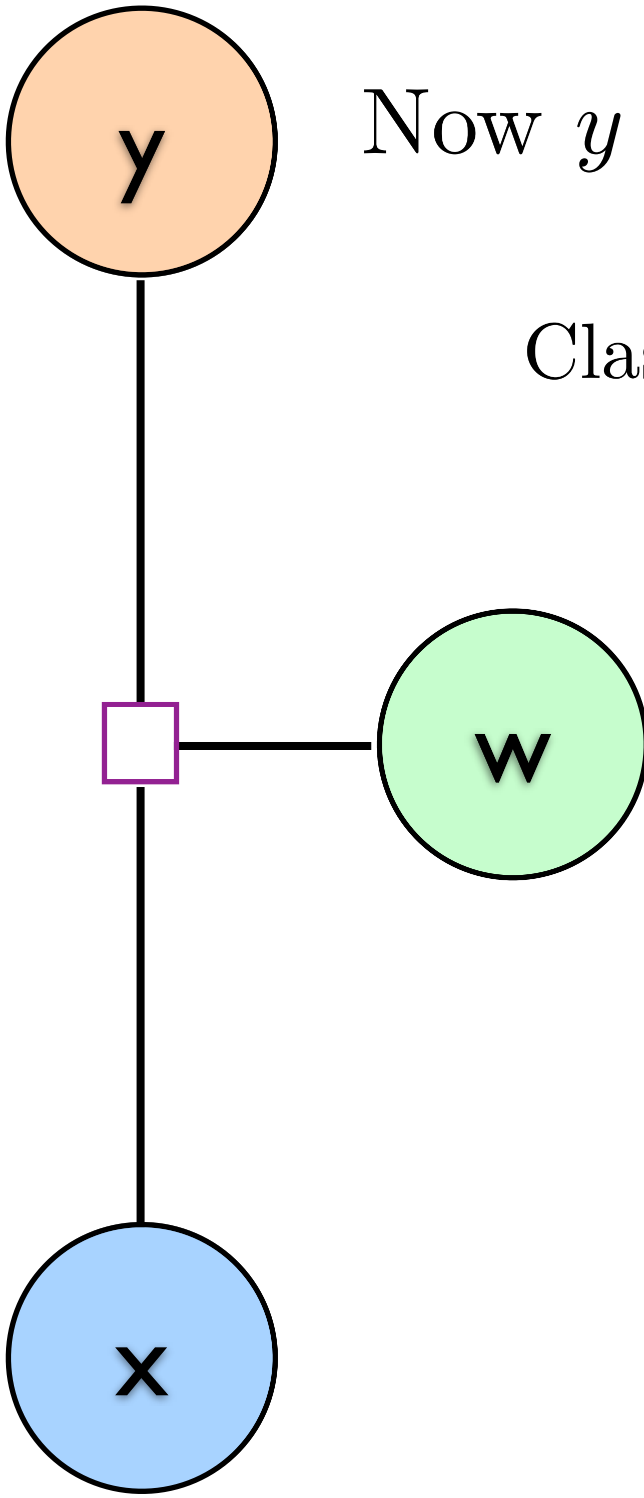
$$x = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Multi-Class Logistic Regression

$y = [3]$

Now $y \in \{1, 2, 3, \ldots, S\}$.

Classify by maximizing $w_s^T x_s$ over s



$$w = \begin{bmatrix} -1.7491 & 1.7411 & 0.8106 \\ 1.1326 & 0.4868 & 0.6985 \\ 0 & 1.0488 & -0.4016 \\ -0.7938 & 1.4886 & 1.2688 \\ 0.3149 & 1.2705 & -0.7836 \end{bmatrix}$$

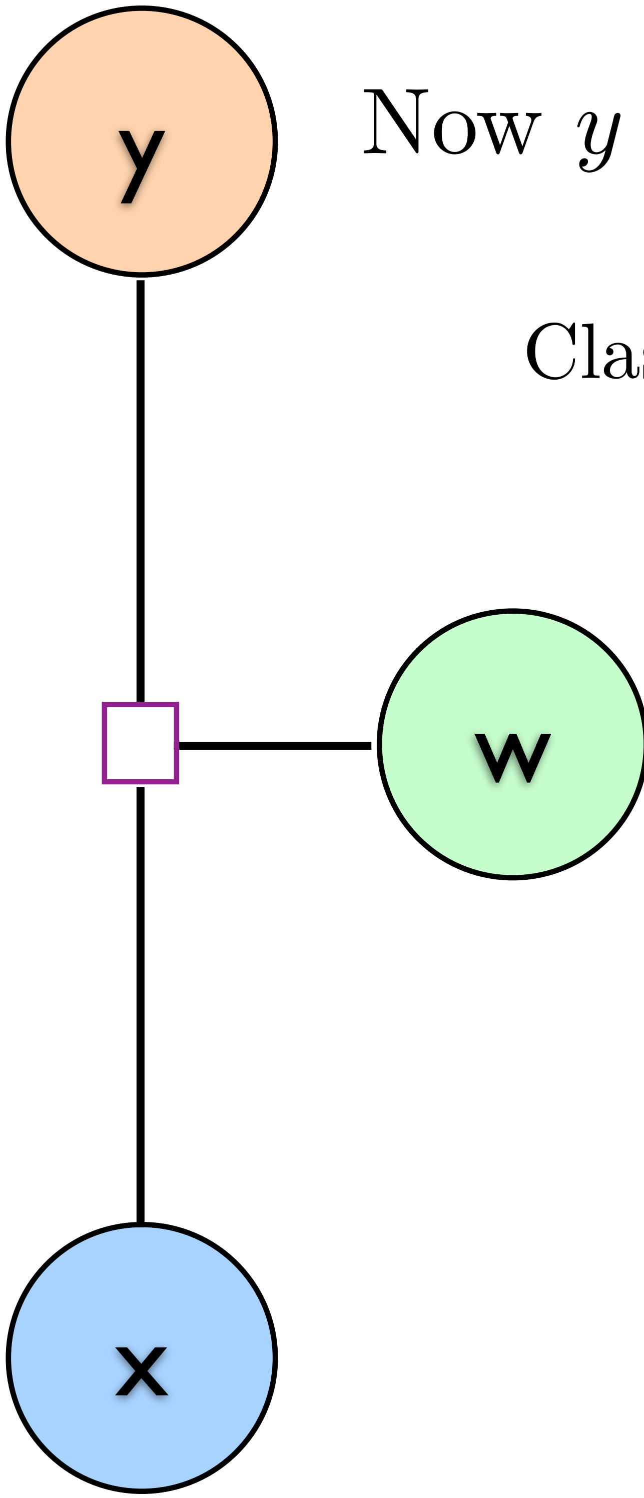
$$x = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Multi-Class Logistic Regression

$y = [3]$

Now $y \in \{1, 2, 3, \ldots, S\}$.

Classify by maximizing $w_s^T x_s$ over s



$$w = \begin{bmatrix} -1.7491 & 1.7411 & 0.8106 \\ 1.1326 & 0.4868 & 0.6985 \\ 0 & 1.0488 & -0.4016 \\ -0.7938 & 1.4886 & 1.2688 \\ 0.3149 & 1.2705 & -0.7836 \end{bmatrix}$$

This is w_3

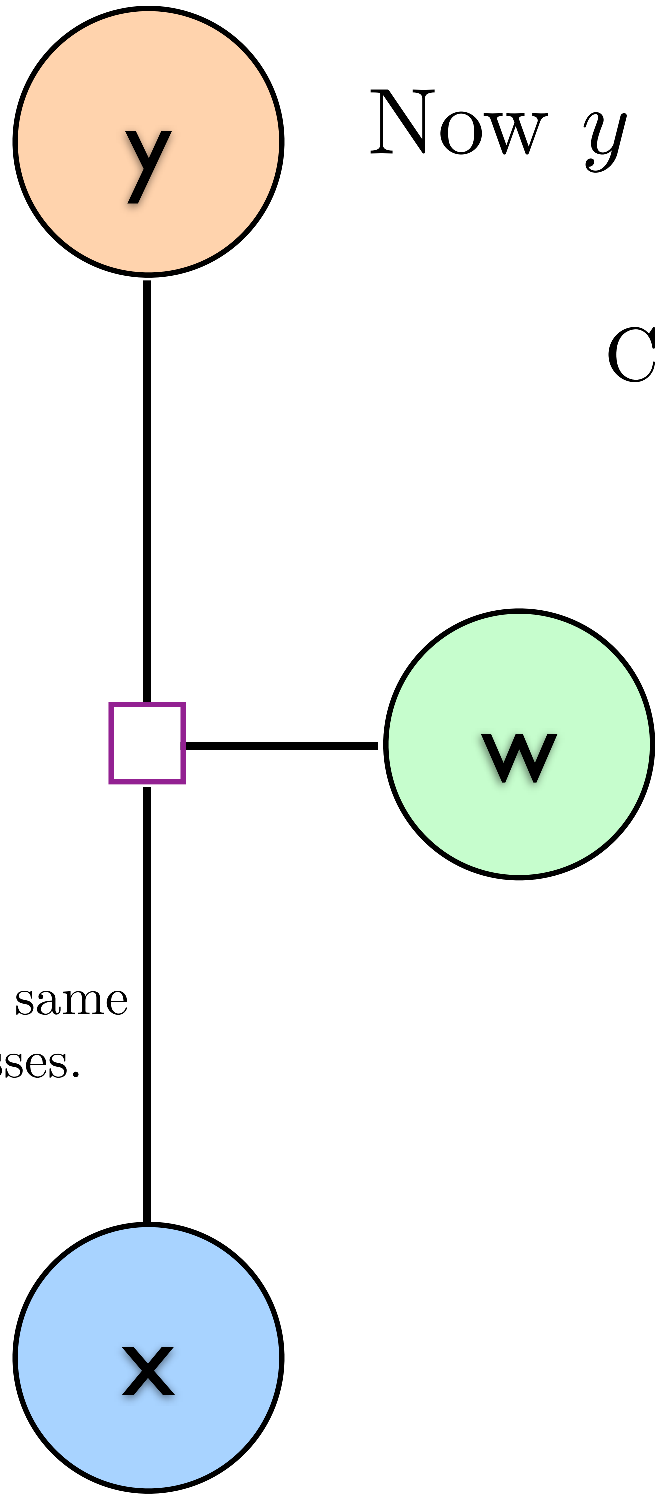
$$x = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Multi-Class Logistic Regression

$y = [3]$

Now $y \in \{1, 2, 3, \ldots, S\}$.

Classify by maximizing $w_s^T x$ over s



$$w = \begin{bmatrix} -1.7491 & 1.7411 & 0.8106 \\ 1.1326 & 0.4868 & 0.6985 \\ 0 & 1.0488 & -0.4016 \\ -0.7938 & 1.4886 & 1.2688 \\ 0.3149 & 1.2705 & -0.7836 \end{bmatrix}$$

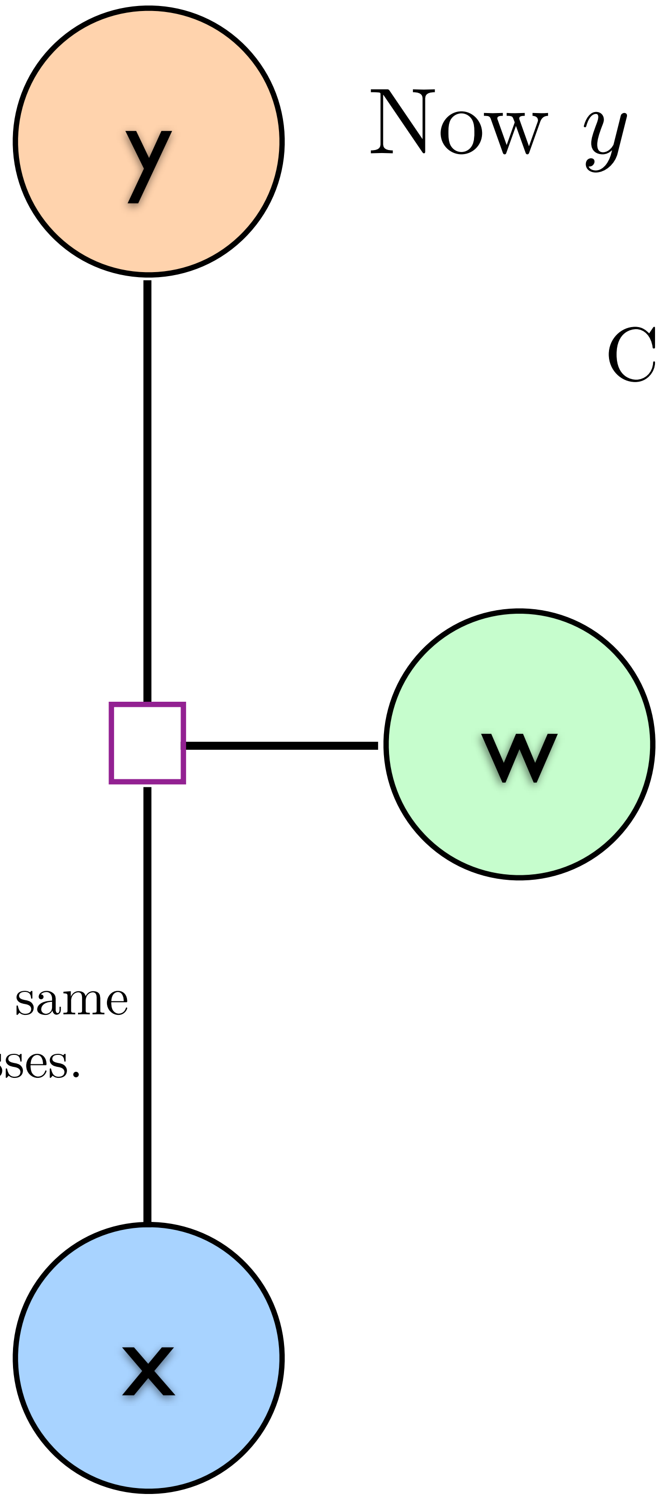
This is w_3

Usually we use the same features across classes.

$$x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Multi-Class Logistic Regression

$y = [3]$



Now $y \in \{1, 2, 3, \ldots, S\}$.

Classify by maximizing $w_s^T x$ over s

$$w = \begin{bmatrix} -1.7491 & 1.7411 & 0.8106 \\ 1.1326 & 0.4868 & 0.6985 \\ 0 & 1.0488 & -0.4016 \\ -0.7938 & 1.4886 & 1.2688 \\ 0.3149 & 1.2705 & -0.7836 \end{bmatrix}$$

This is w_3

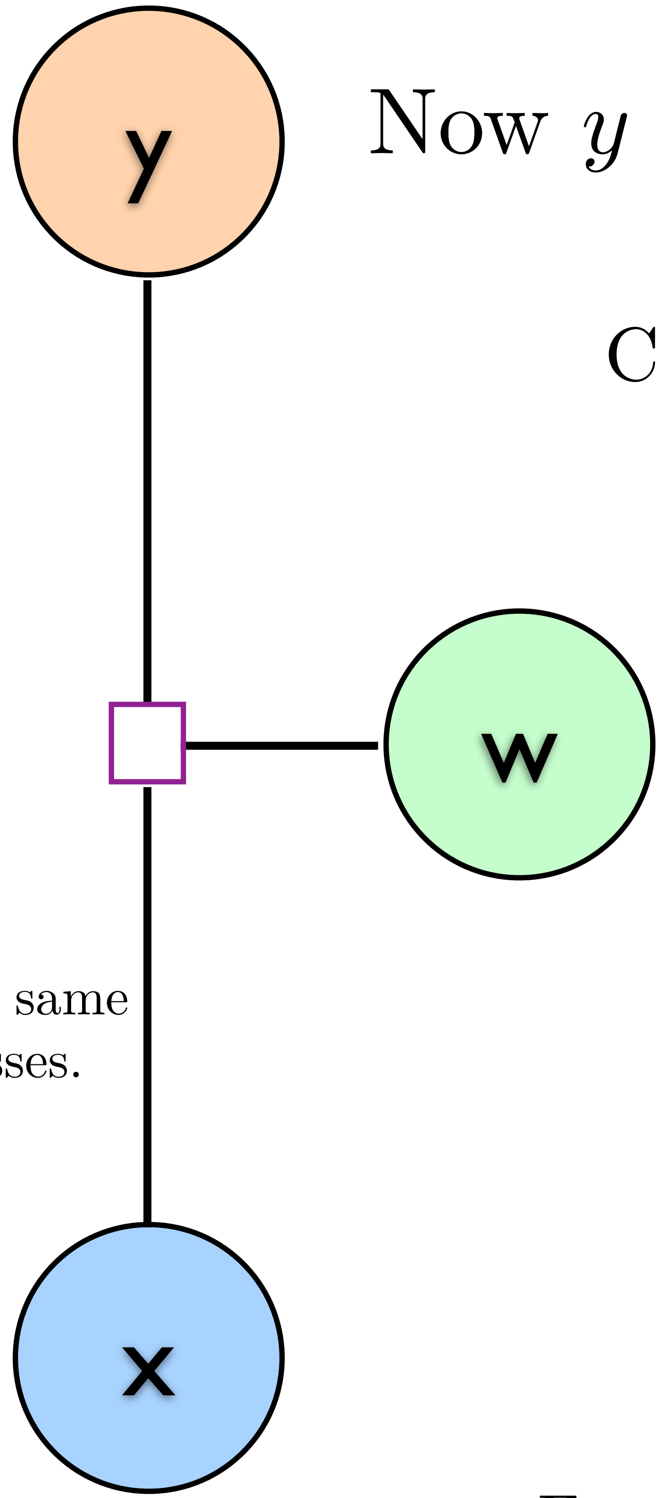
Usually we use the same features across classes.

$$x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$$\boxed{p(y = s | x, w) \propto \exp(w_s^T x)}$$

Multi-Class Logistic Regression

$y = [3]$



Now $y \in \{1, 2, 3, \dots, S\}$.

Classify by maximizing $w_s^T x$ over s

$$w = \begin{bmatrix} -1.7491 & 1.7411 & 0.8106 \\ 1.1326 & 0.4868 & 0.6985 \\ 0 & 1.0488 & -0.4016 \\ -0.7938 & 1.4886 & 1.2688 \\ 0.3149 & 1.2705 & -0.7836 \end{bmatrix}$$

This is w_3

Usually we use the same features across classes.

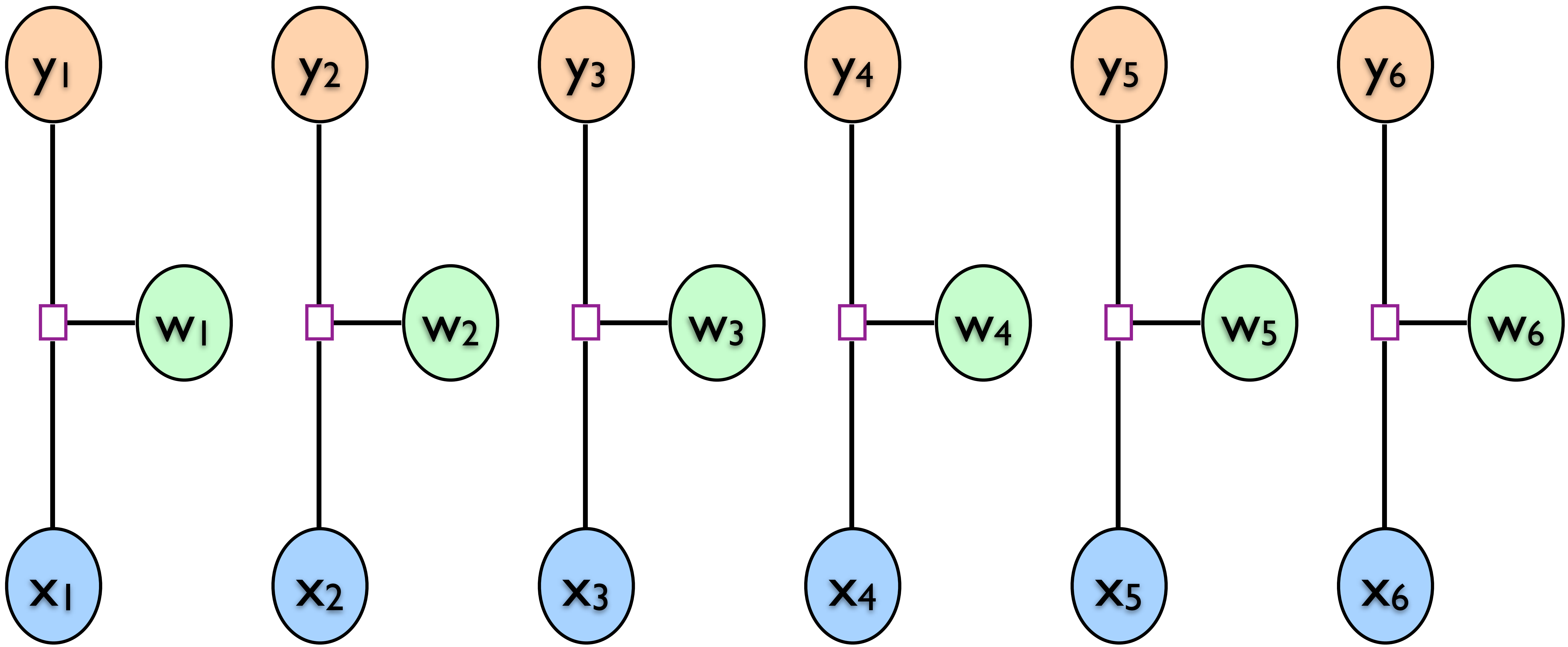
$$x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$$\boxed{p(y = s | x, w) \propto \exp(w_s^T x)}$$

For *ordered* classes, use *ordinal* logistic regression.

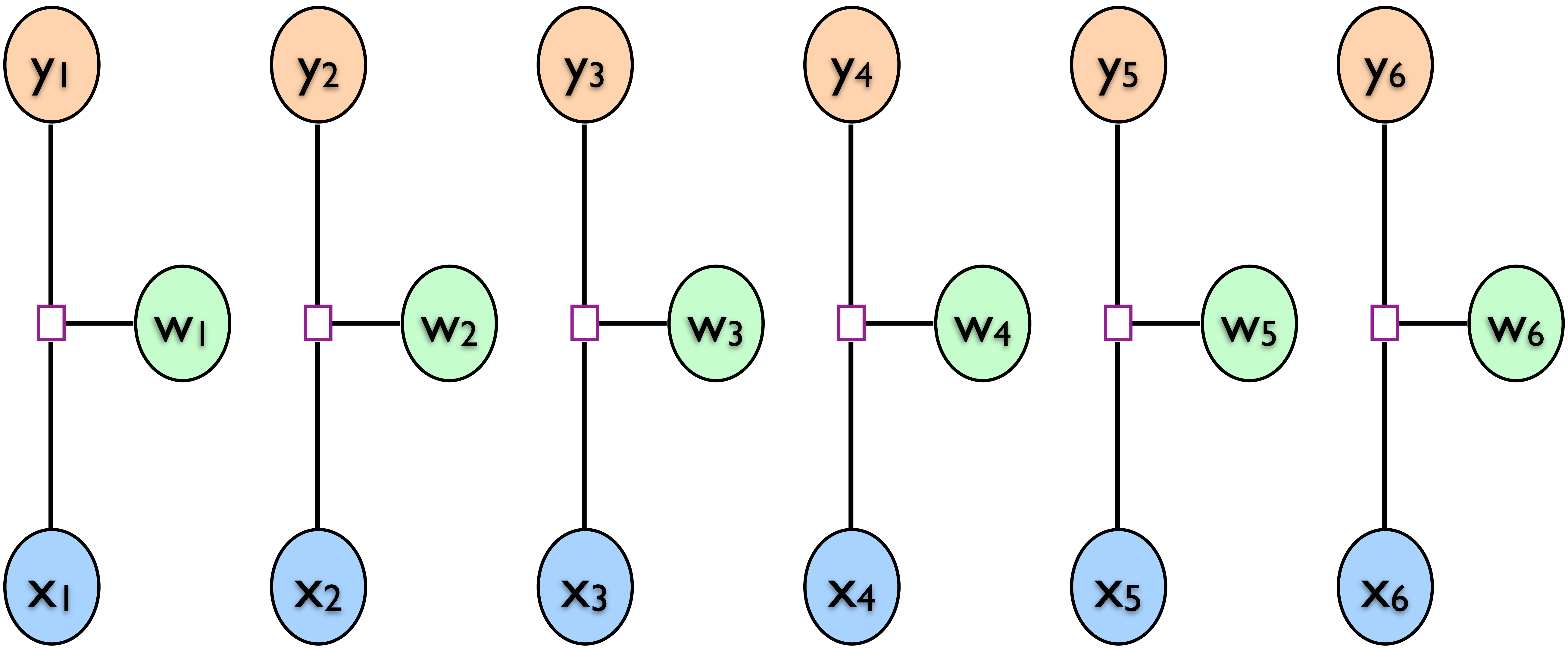
Multi-Label Logistic Regression

We now have multiple labels y_n



Multi-Label Logistic Regression

We now have multiple labels y_n



$$p(y = s|x, w) \propto \prod_{n=1}^{N} \exp(w_{n,s_n}^T x_n)$$

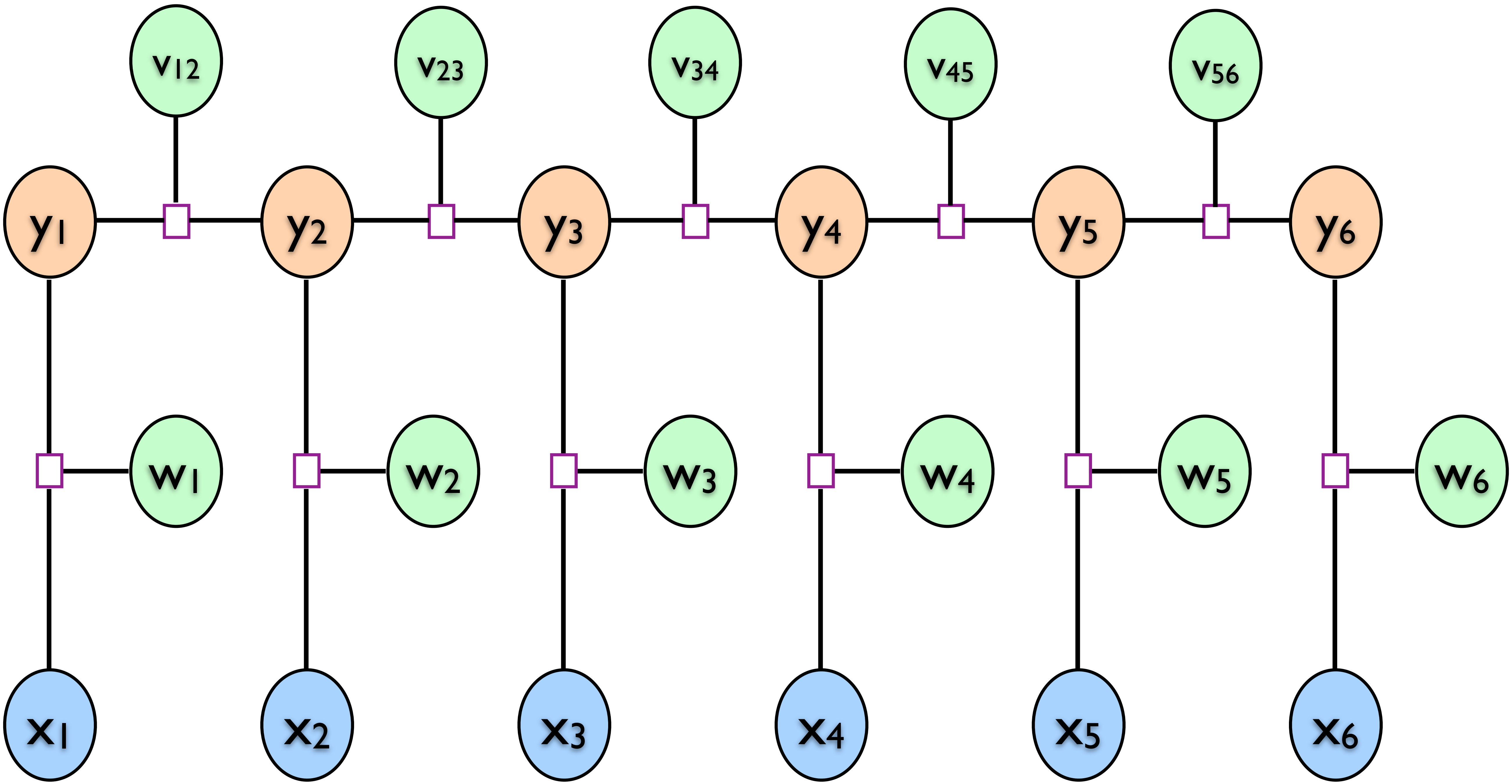
Challenges:

share information across the w_n

model in correlations in the y_n

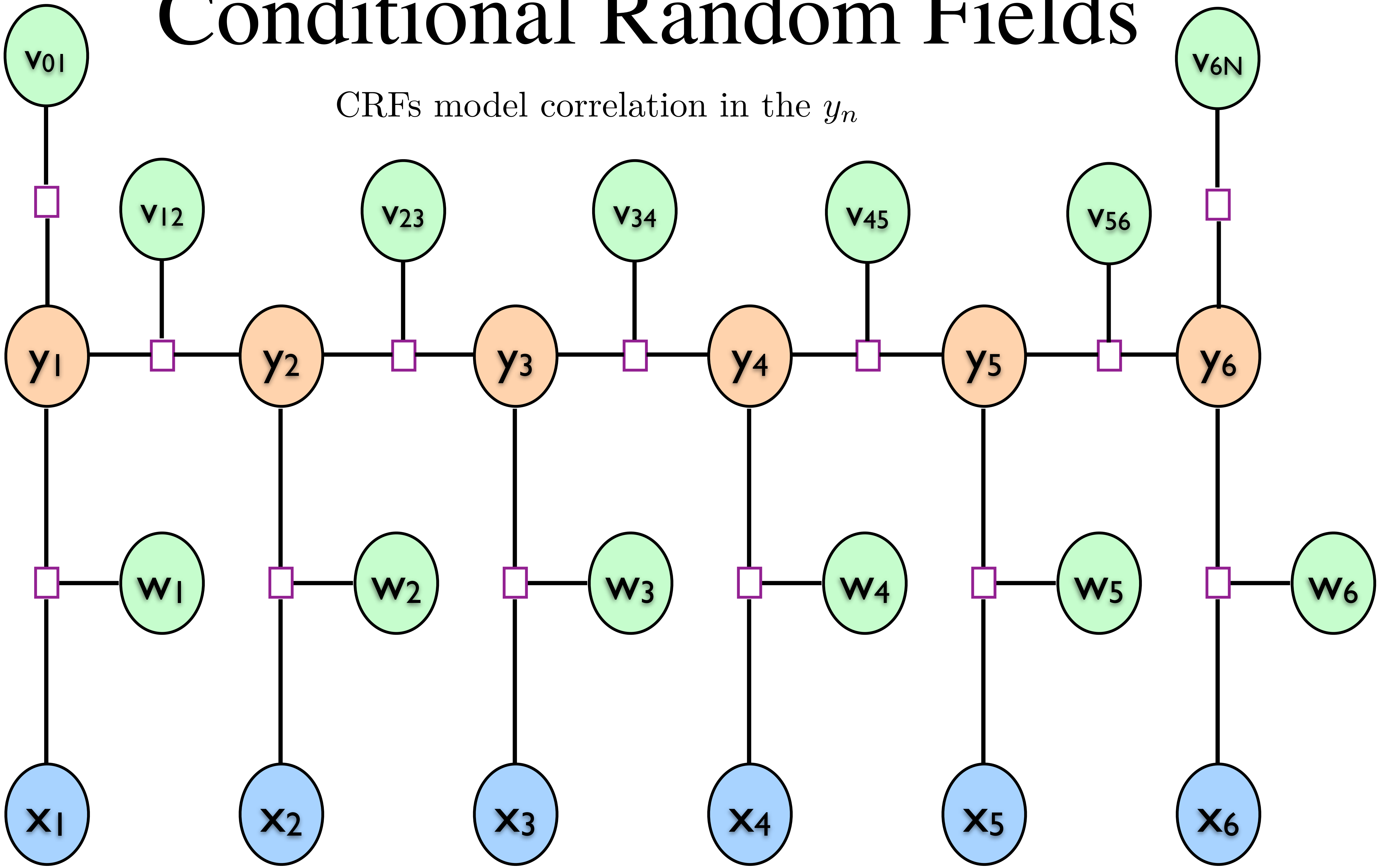
Conditional Random Fields

CRFs model correlation in the y_n



Conditional Random Fields

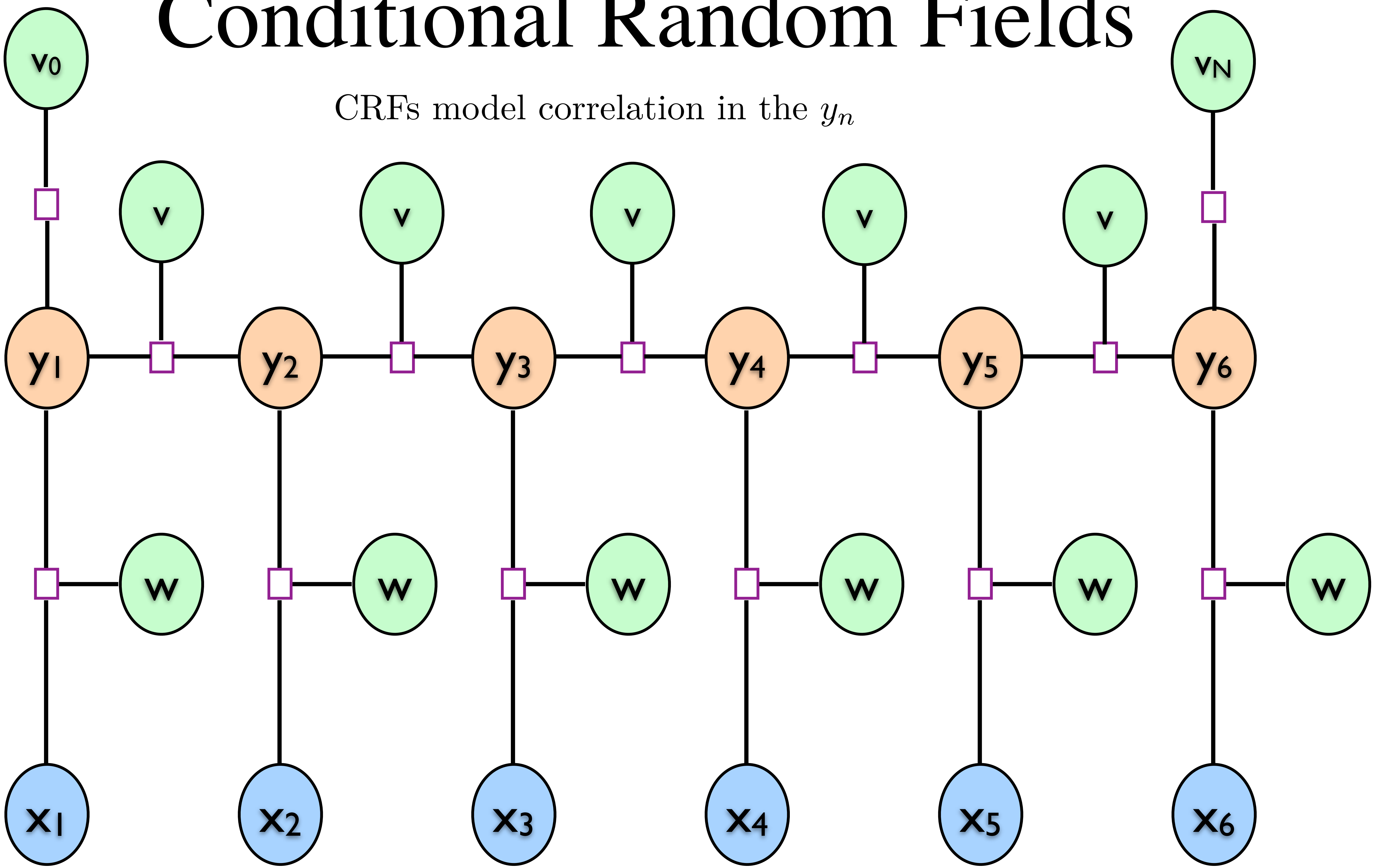
CRFs model correlation in the y_n



Can have special potentials on start/end

Conditional Random Fields

CRFs model correlation in the y_n

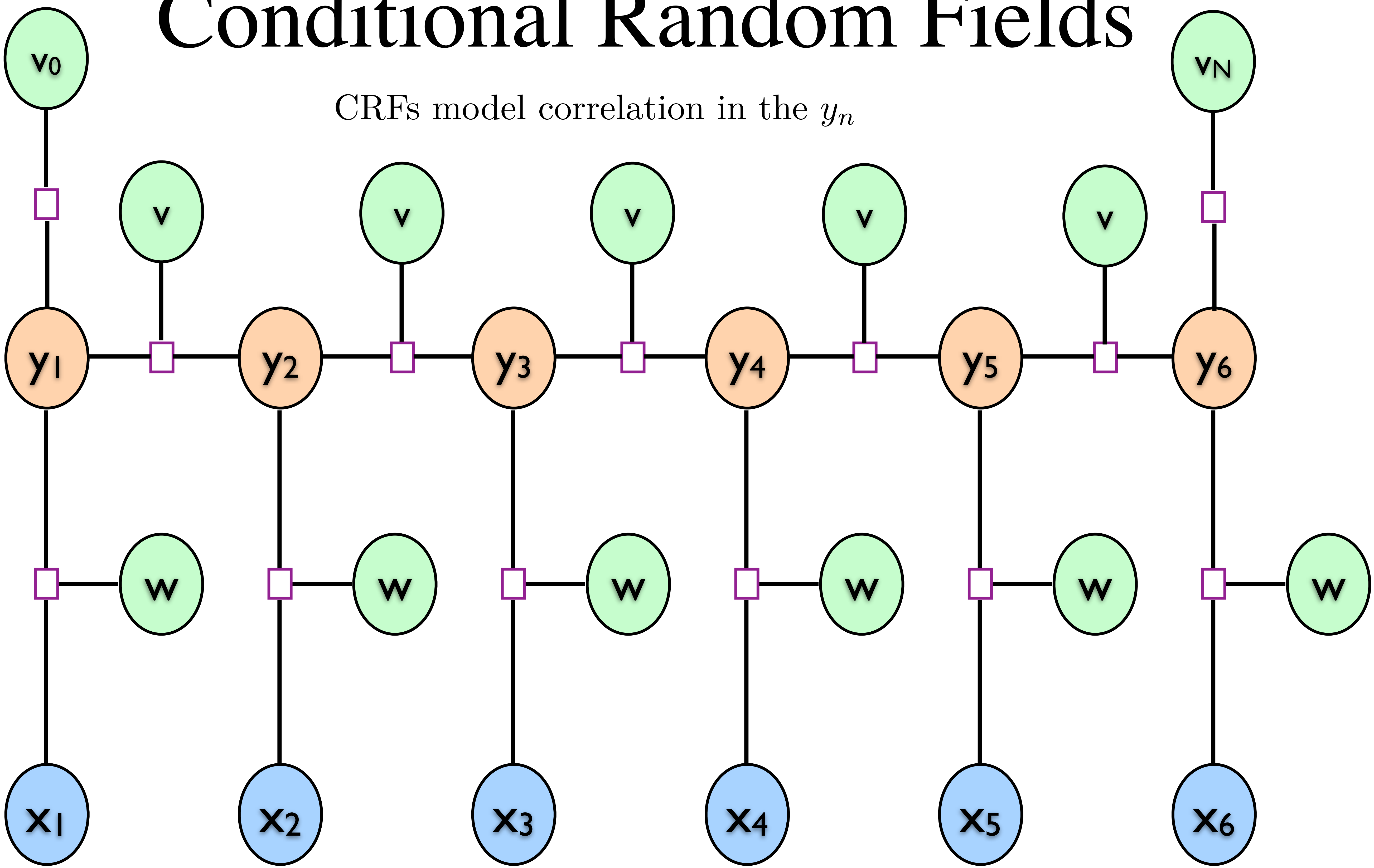


Can have special potentials on start/end

We often tie parameters (but can have node/edge types)

Conditional Random Fields

CRFs model correlation in the y_n



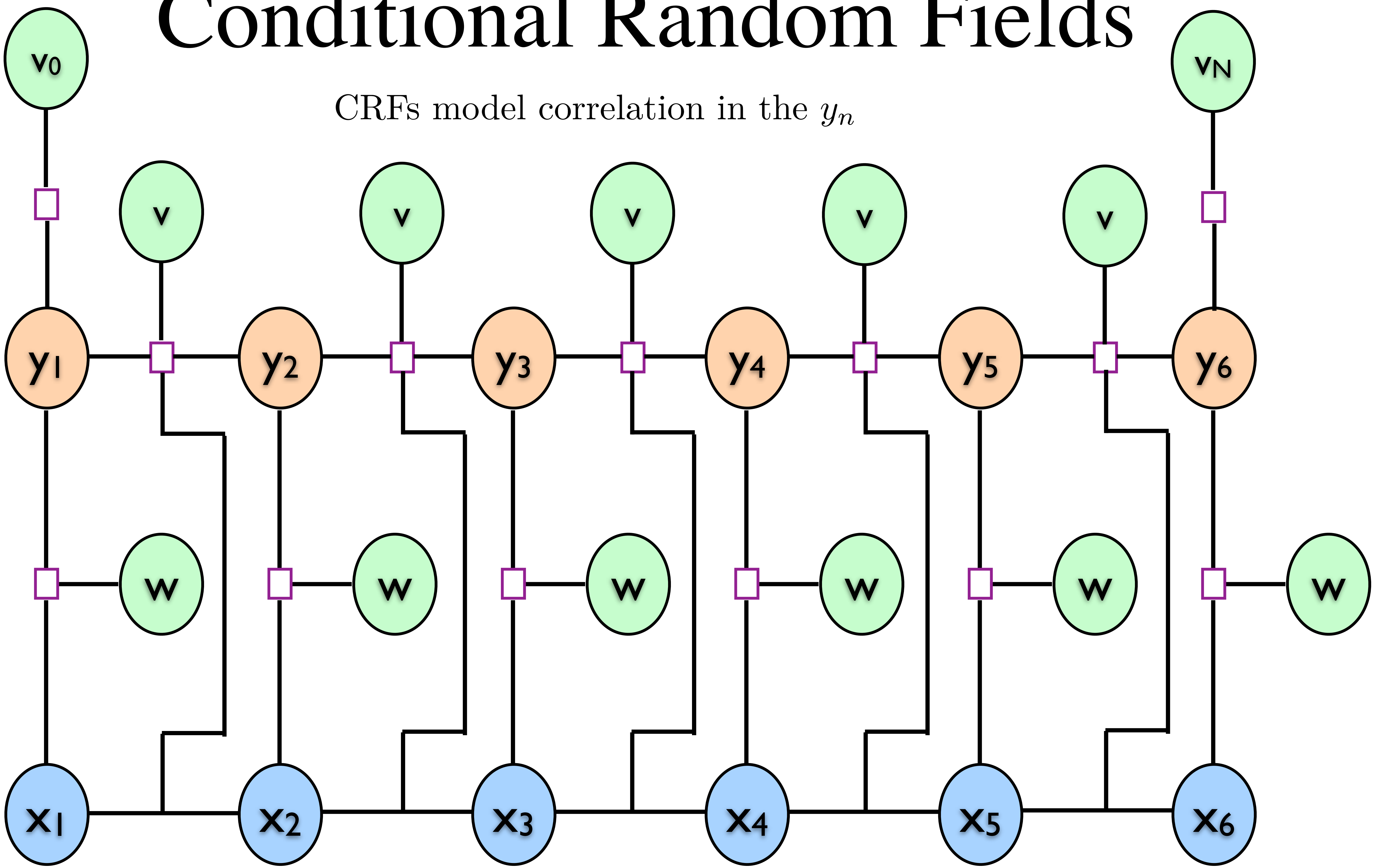
Can have special potentials on start/end

We often tie parameters (but can have node/edge types)

Could also share information through regularization

Conditional Random Fields

CRFs model correlation in the y_n



Can have special potentials on start/end

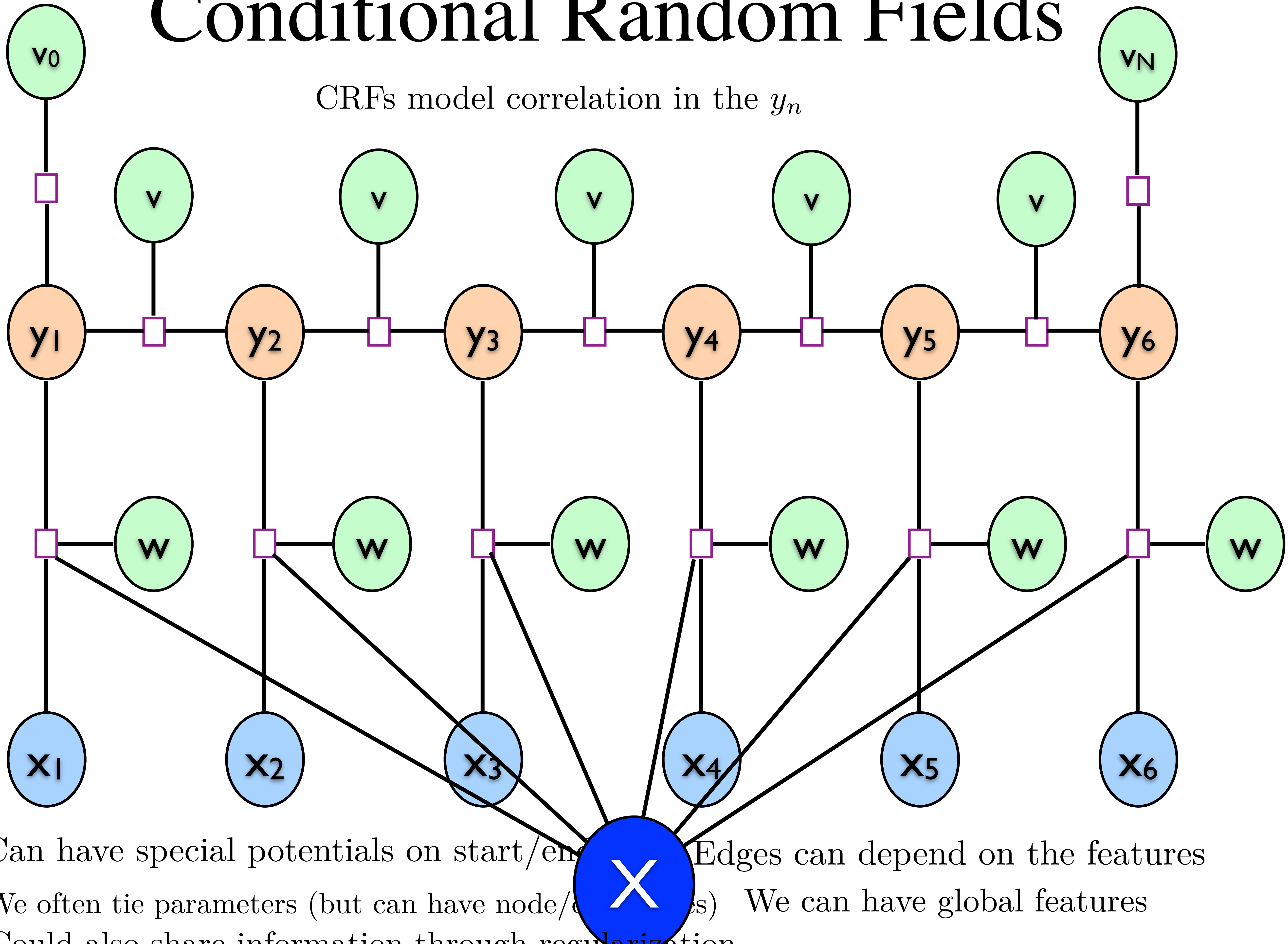
Edges can depend on the features

We often tie parameters (but can have node/edge types)

Could also share information through regularization

Conditional Random Fields

CRFs model correlation in the y_n



Can have special potentials on start/en

Edges can depend on the features

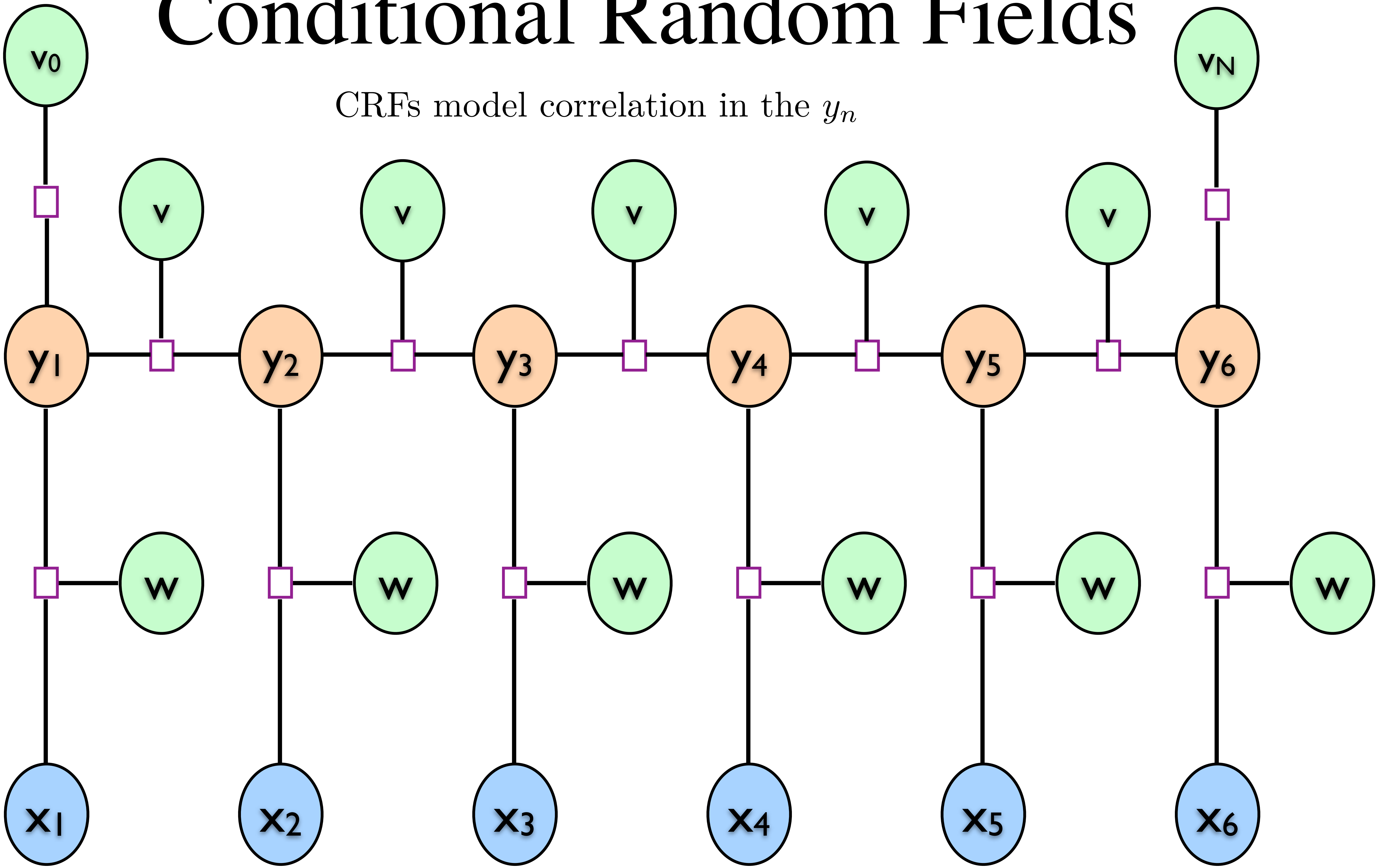
We often tie parameters (but can have node/e es)

We can have global features

Could also share information through regularization

Conditional Random Fields

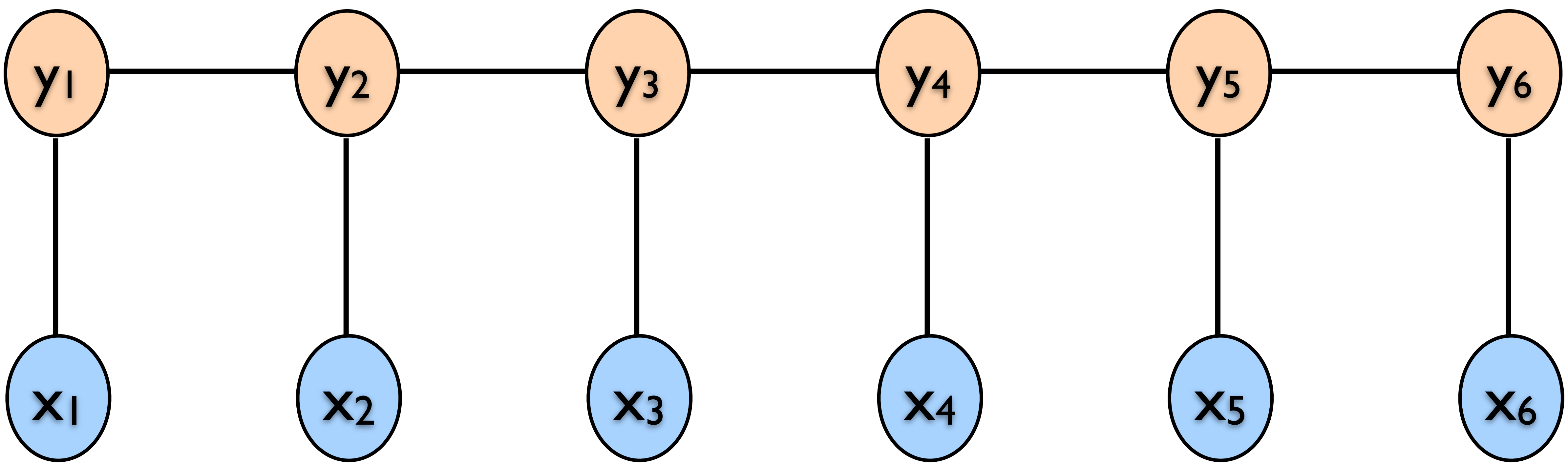
CRFs model correlation in the y_n



$$p(y = s|x, w) \propto \prod_{n=1}^{N} \exp(w_{s_n}^T x_n) \prod_{n=0}^{N} \exp(v_{s_n, s_{n+1}})$$

General Conditional Random Fields

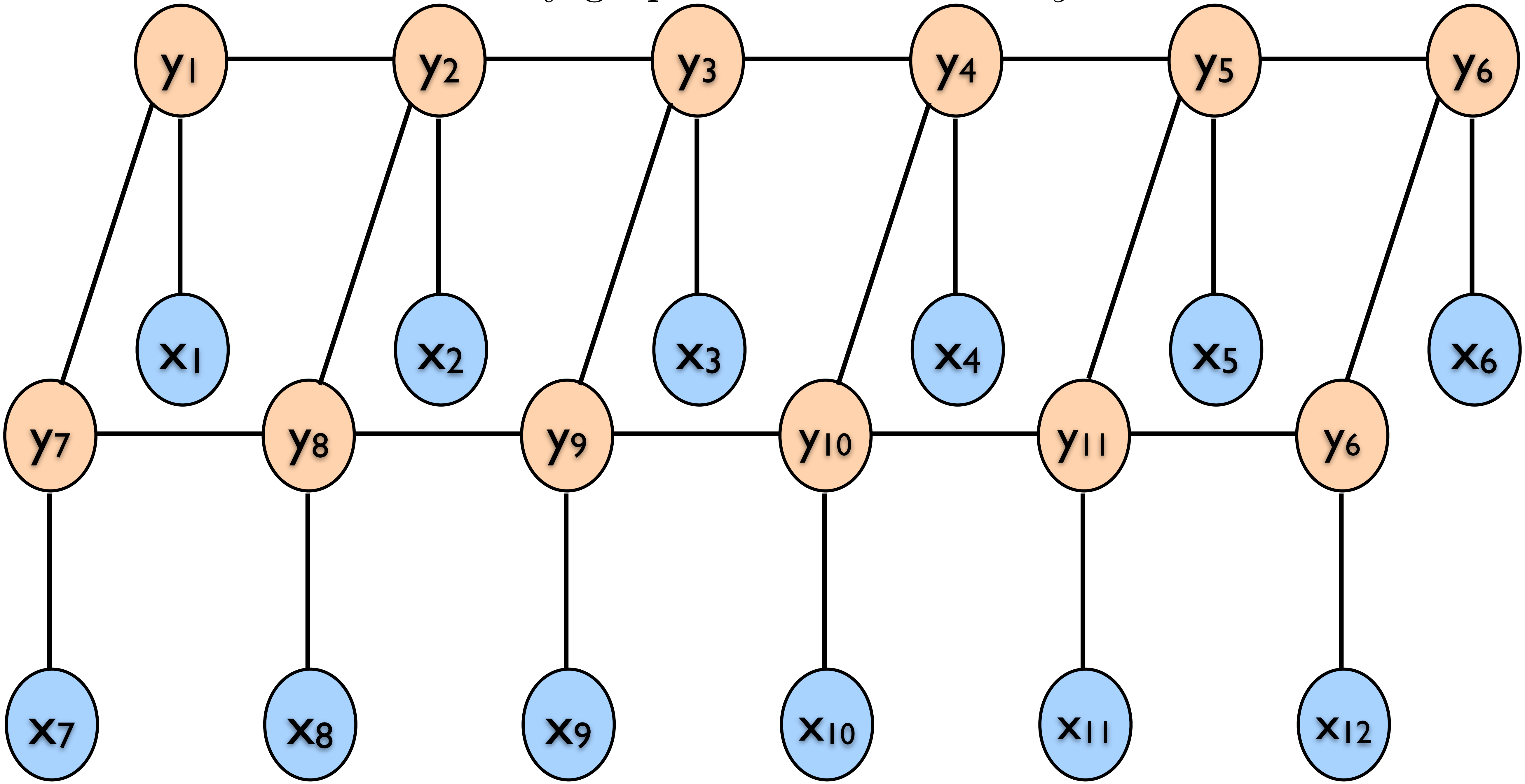
We can have any graph structure on the y_n



$$p(y = s|x, w) \propto \prod_{i \in N} \exp(w_{s_i}^T x_i) \prod_{i,j \in E}^{N} \exp(v_{s_i, s_j})$$

General Conditional Random Fields

We can have any graph structure on the y_n



$$p(y = s|x, w) \propto \prod_{i \in N} \exp(w_{s_i}^T x_i) \prod_{i,j \in E}^{N} \exp(v_{s_i, s_j})$$

General Conditional Random Fields

Tasks involving states s:

- Decoding: $\arg\max_s p(y = s|x, w)$
- Inference: $\sum_s p(y = s|x, w)$ and $\sum_{s|s_i=c} p(y = s|x, w)$
- Sampling: generate $s \sim p(y = s|x, w)$

For chain structured data:

- Decode using Viterbi
- Inference using Forward-Backward
- Sampling using Forward-Filter, Backward-Sample

General Conditional Random Fields

Exact methods:

- Cutset conditioning
- Super nodes
- Junction tree
- Graph cuts (for decoding of binary associative)

General Conditional Random Fields

Exact methods:

- Cutset conditioning
- Super nodes
- Junction tree
- Graph cuts (for decoding of binary associative)

Exponential in tree-width

General Conditional Random Fields

Exact methods:

- Cutset conditioning
- Super nodes
- Junction tree

Exponential in tree-width

- Graph cuts (for decoding of binary associative)

Approximate methods:

- Decode using local search
- Inference using variational
- Sample using MCMC

General Conditional Random Fields

Exact methods:

- Cutset conditioning
- Super nodes
- Junction tree
- Graph cuts (for decoding of binary associative)

Exponential in tree-width

Approximate methods:

- Decode using local search
- Inference using variational
- Sample using MCMC

Use one task to perform the other:

- Inference wtih sampling: counting
- Inference with decoding: Viterbi approximation
- Decoding with inference: max-product
- Decoding with sampling: simulated annealing
- Sampling with inference: variational MCMC
- Sampling with decoding: herding

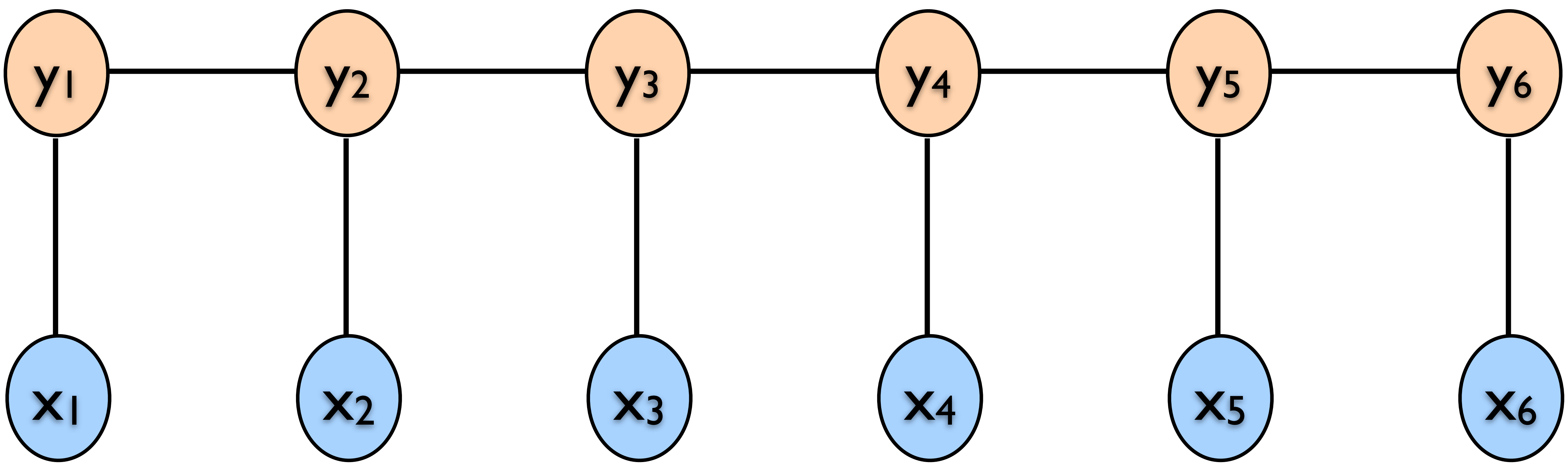
General Conditional Random Fields

Estimation methods to find w:

- Inference: maximum likelihood and regularized maximum likelihood
- Decoding: perceptron and max-margin Markov networks
- Sampling: contrastive divergence and stochastic maximum likelihood
- None: pseudo-likelihood and composite likelihoods

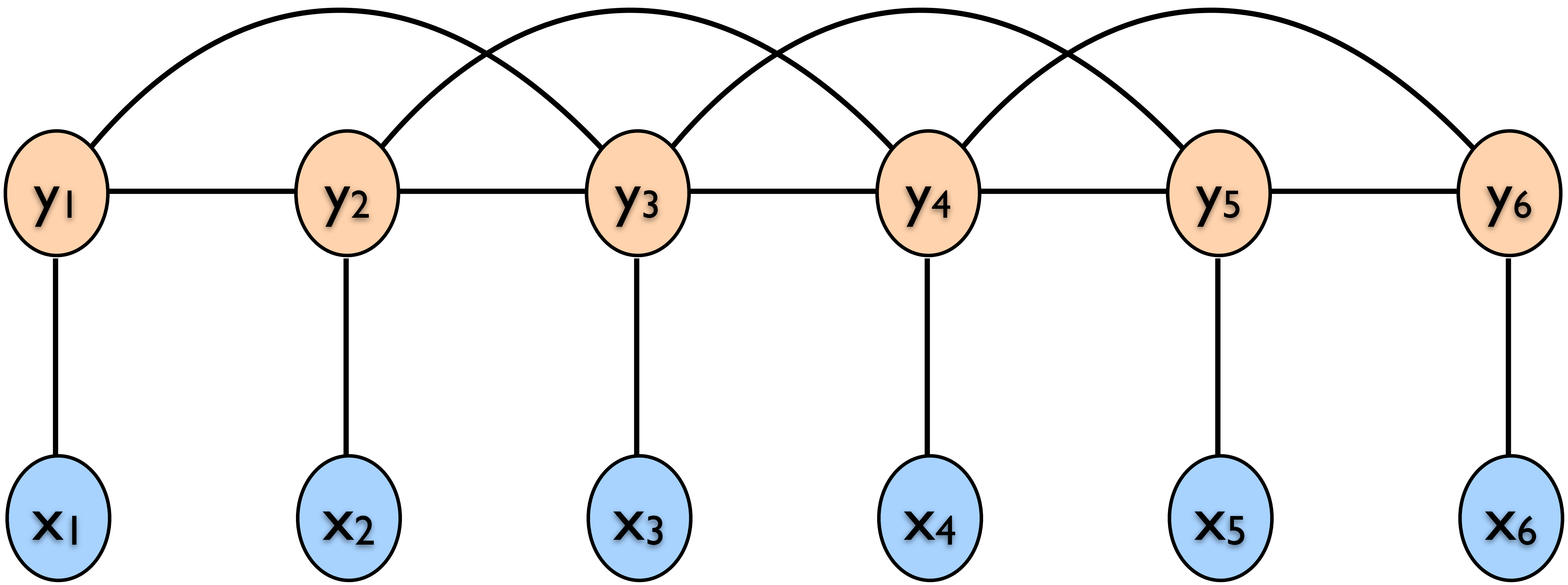
Higher-Order CRF

Add 2nd- or higher-order dependencies



Higher-Order CRF

Add 2nd- or higher-order dependencies

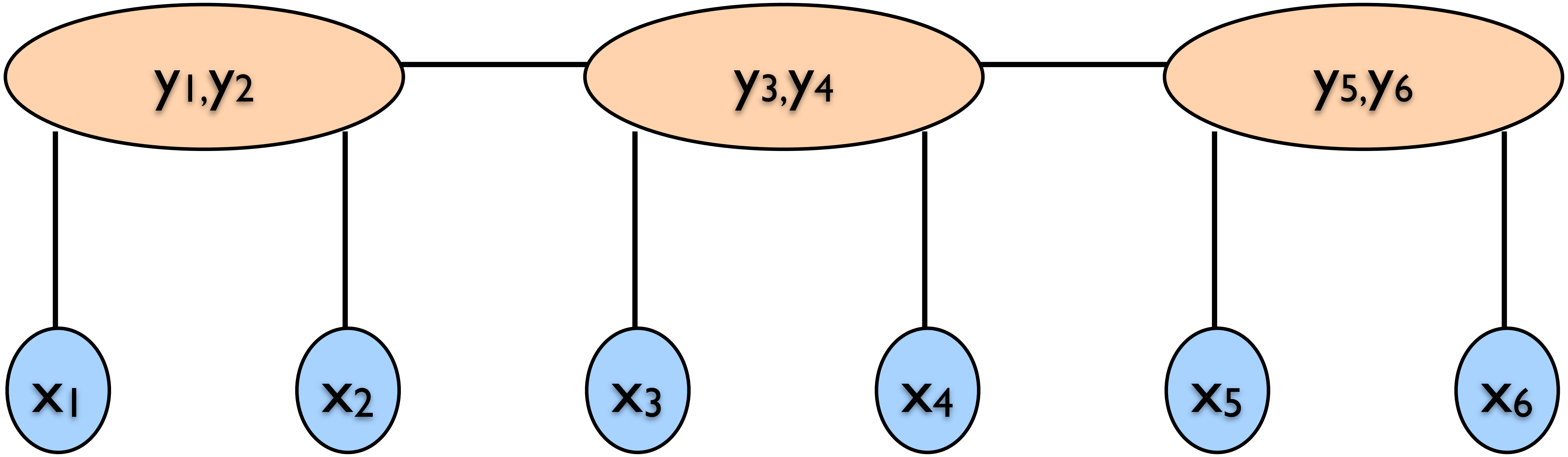


Higher-Order CRF

Add 2nd- or higher-order dependencies

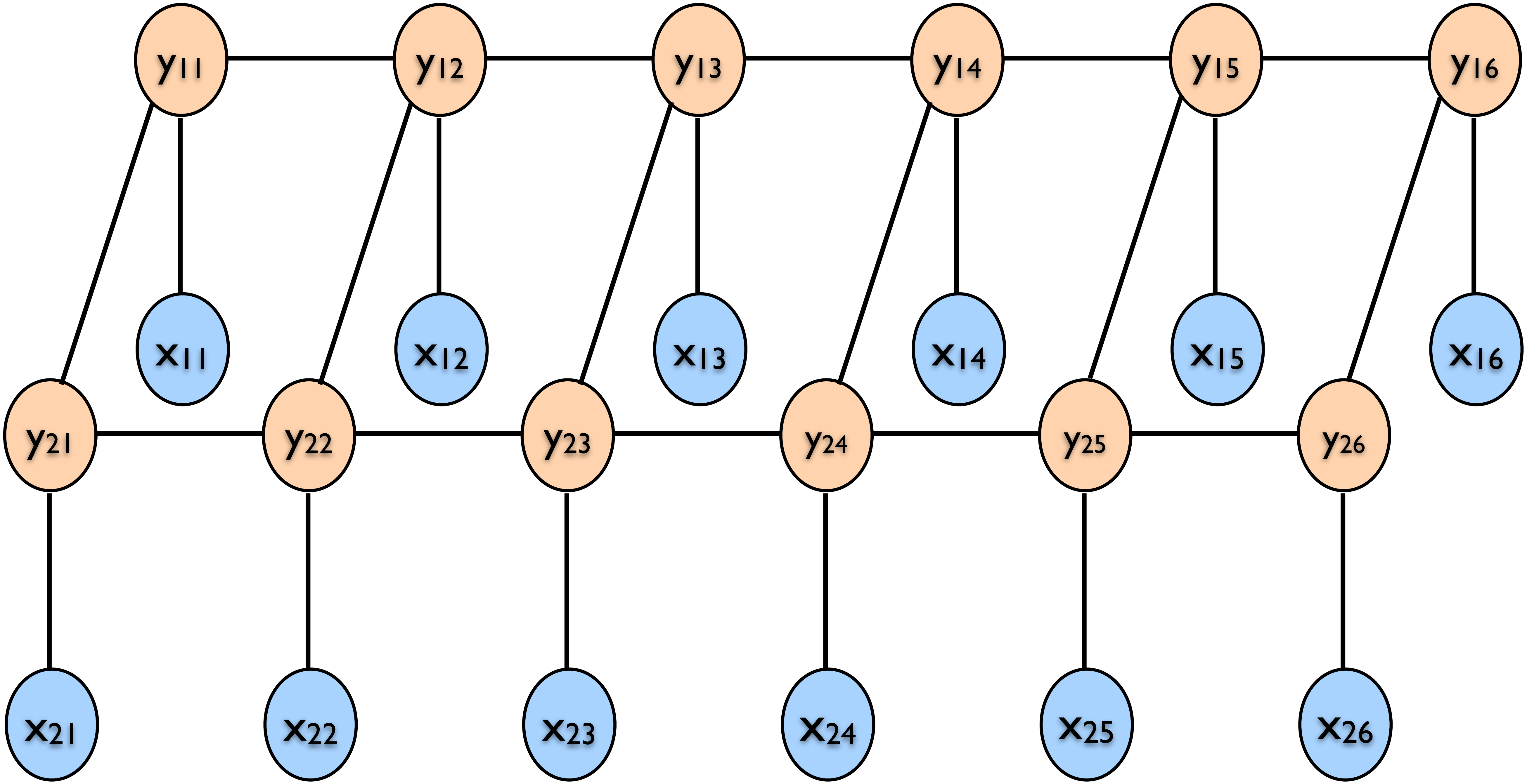
Inference with super nodes:

- 1st-order: $O(NS^2)$
- 2nd-order: $O(\frac{N}{2}S^4)$
- ith-order: $O(\frac{N}{i}S^{2i})$



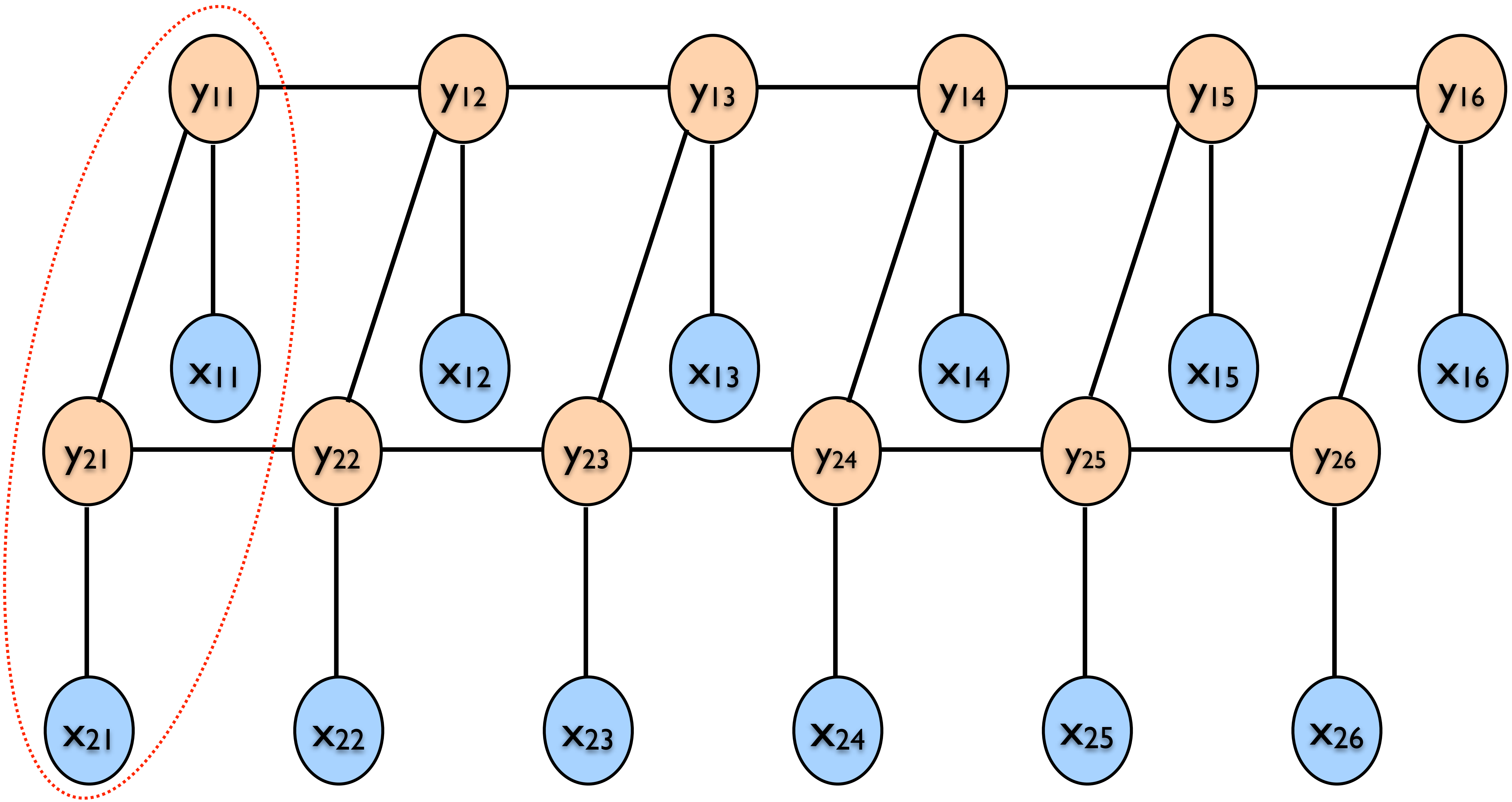
Dynamic CRFs

Track multiple variables with repeated structure



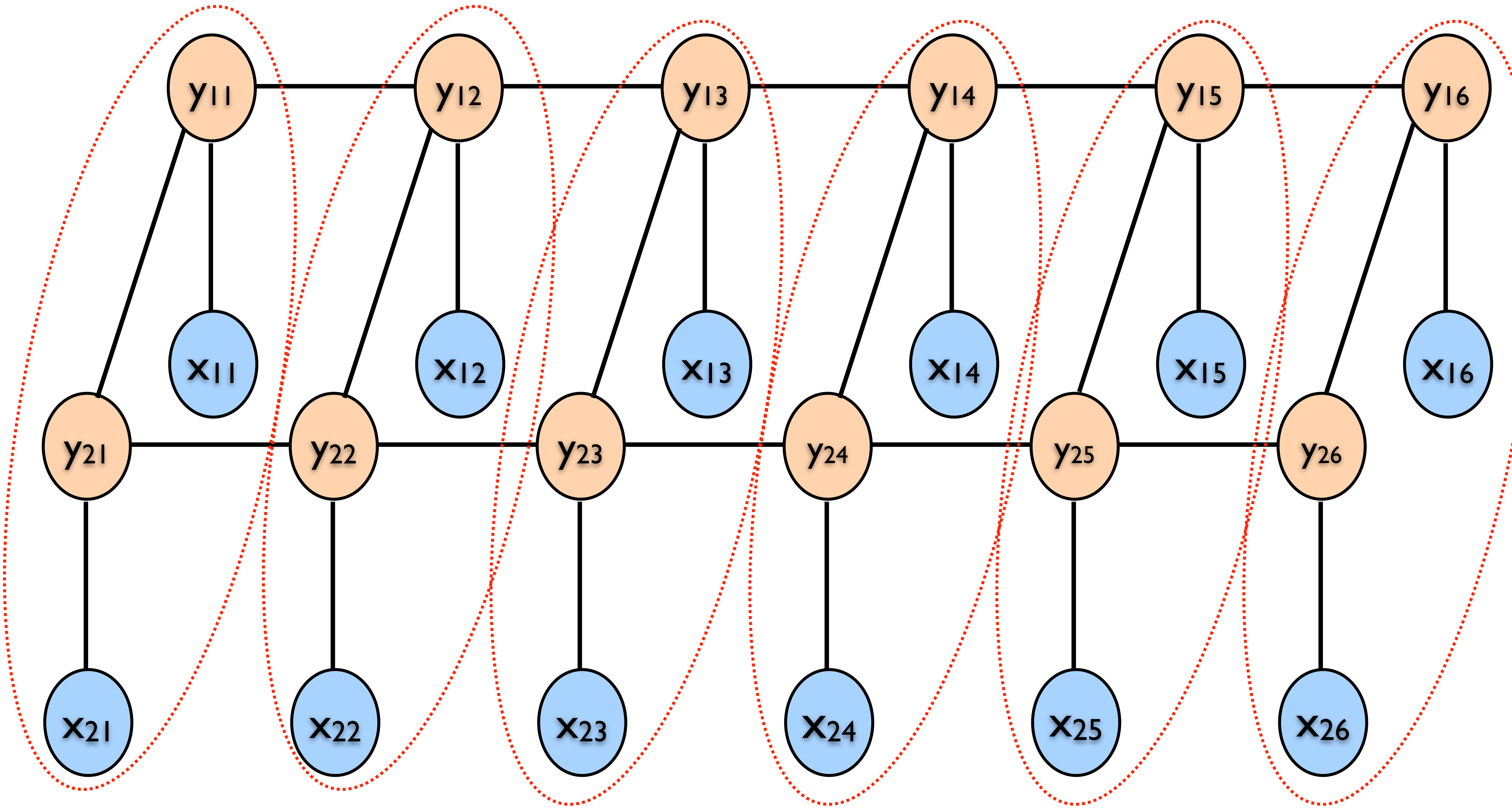
Dynamic CRFs

Track multiple variables with repeated structure



Dynamic CRFs

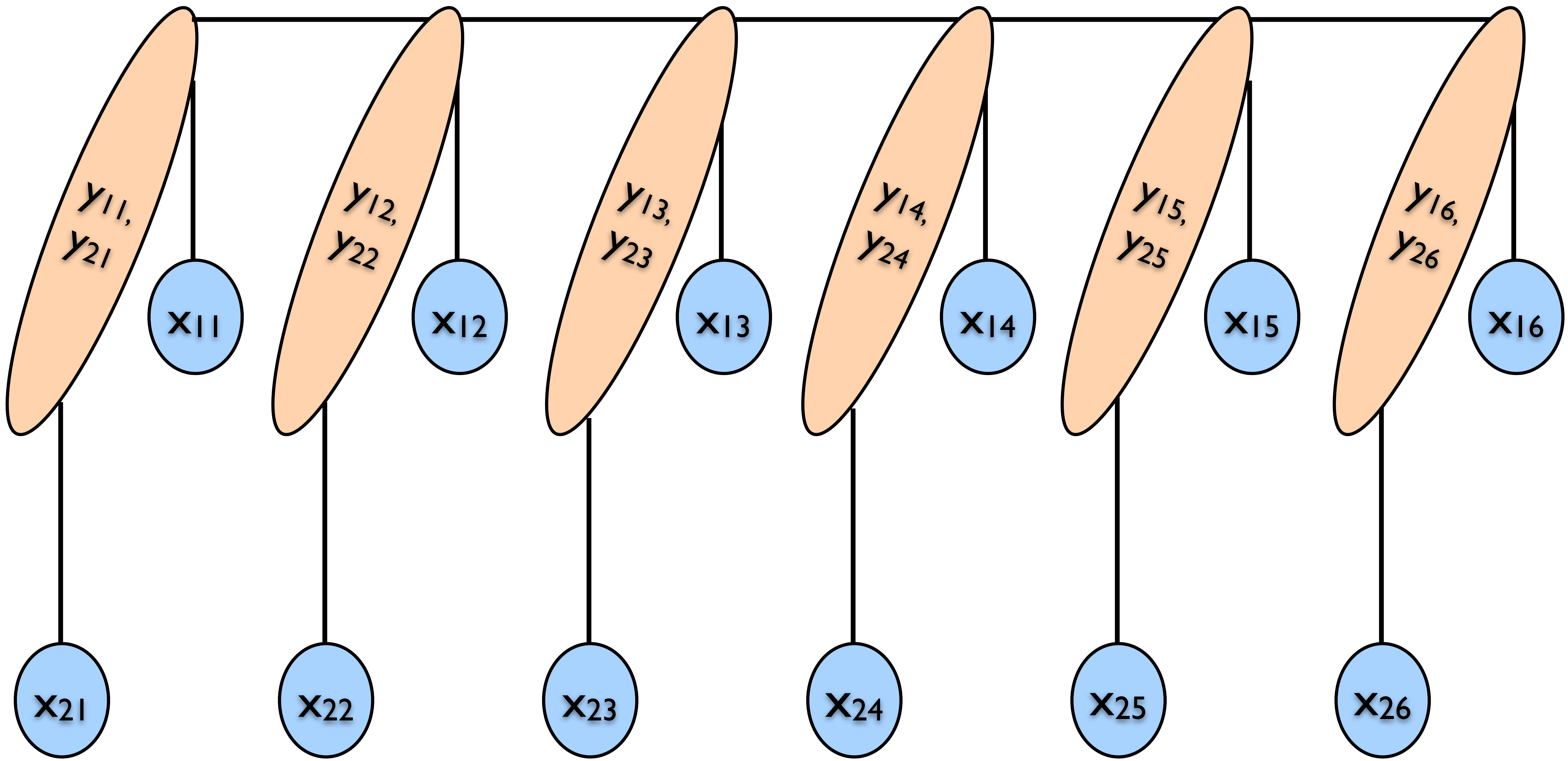
Track multiple variables with repeated structure



Dynamic CRFs

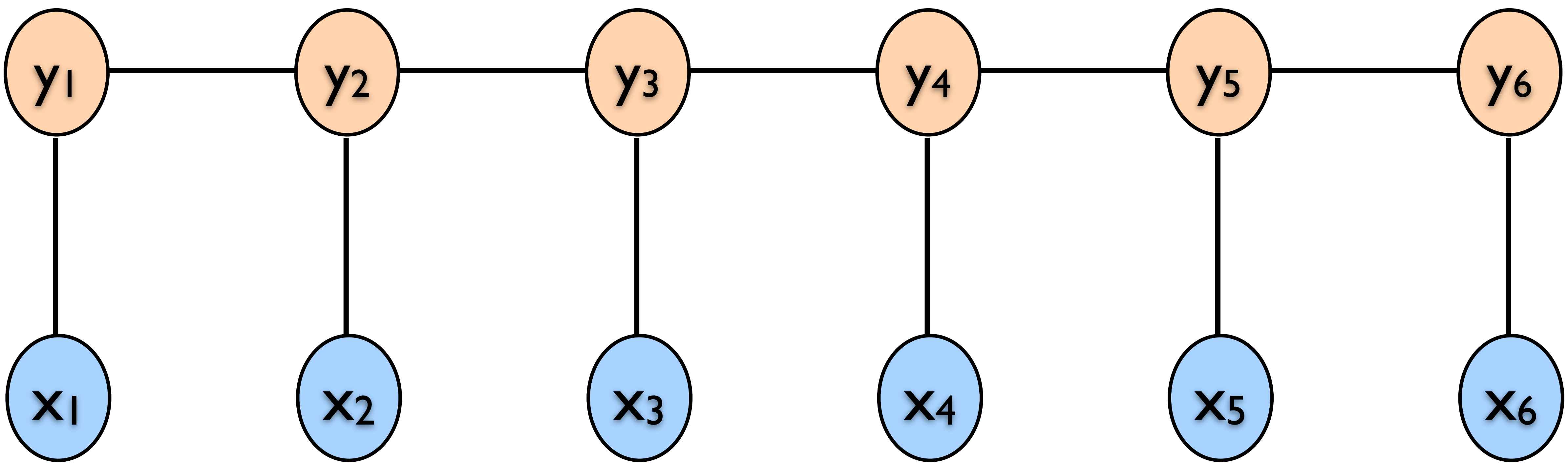
Track multiple variables with repeated structure

Inference with super-nodes



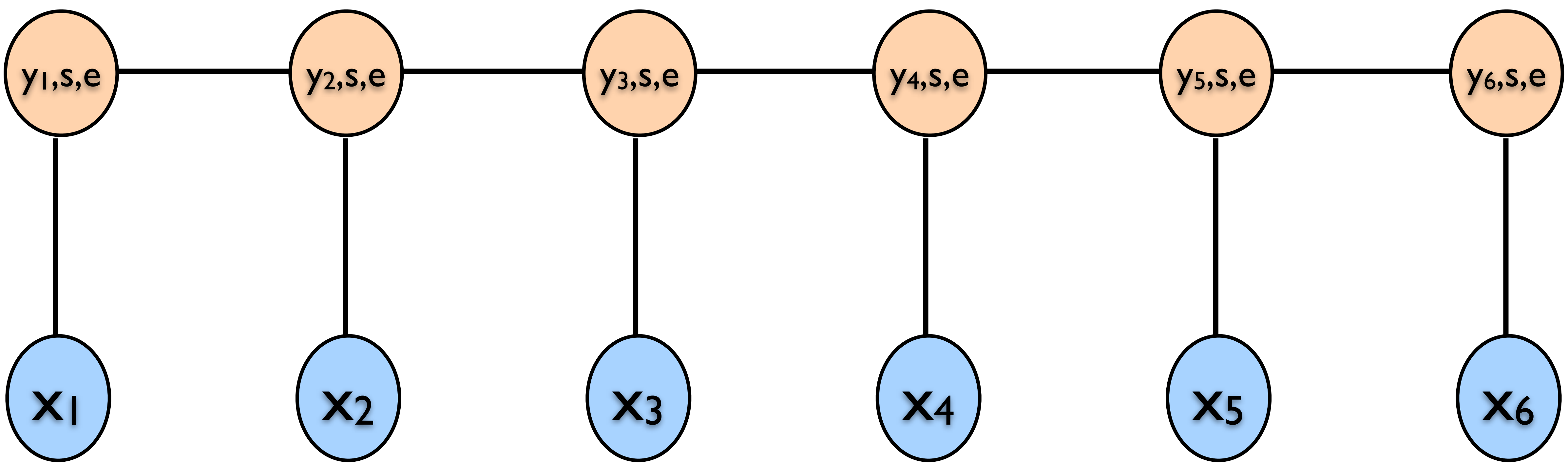
Semi-Markov CRF

Add dependency on length of segment



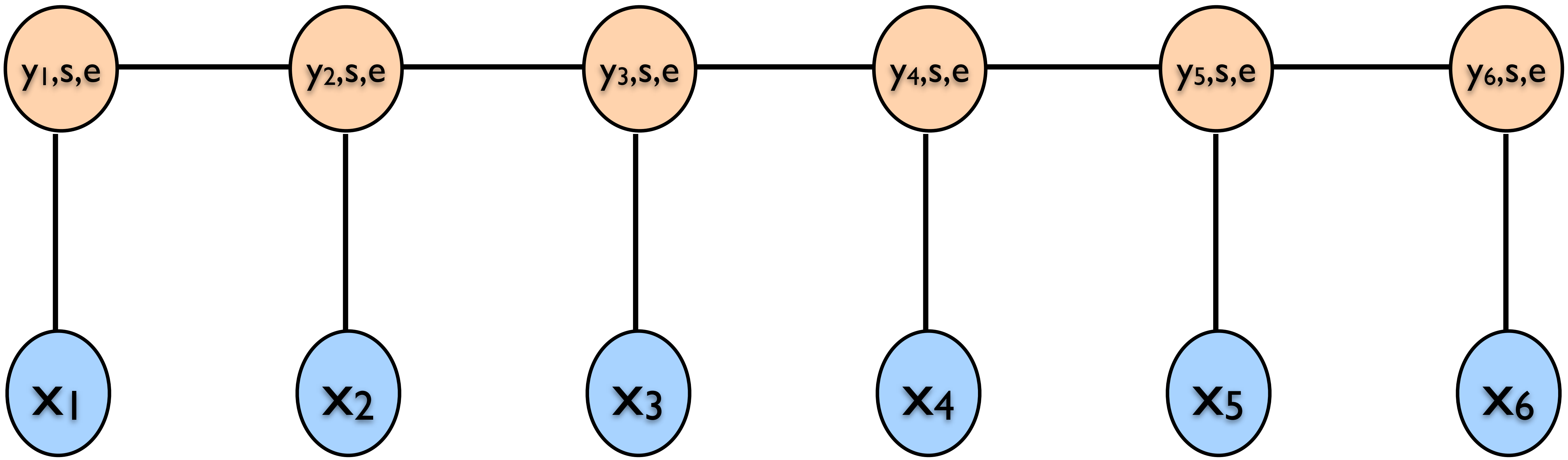
Semi-Markov CRF

Add dependency on length of segment



Semi-Markov CRF

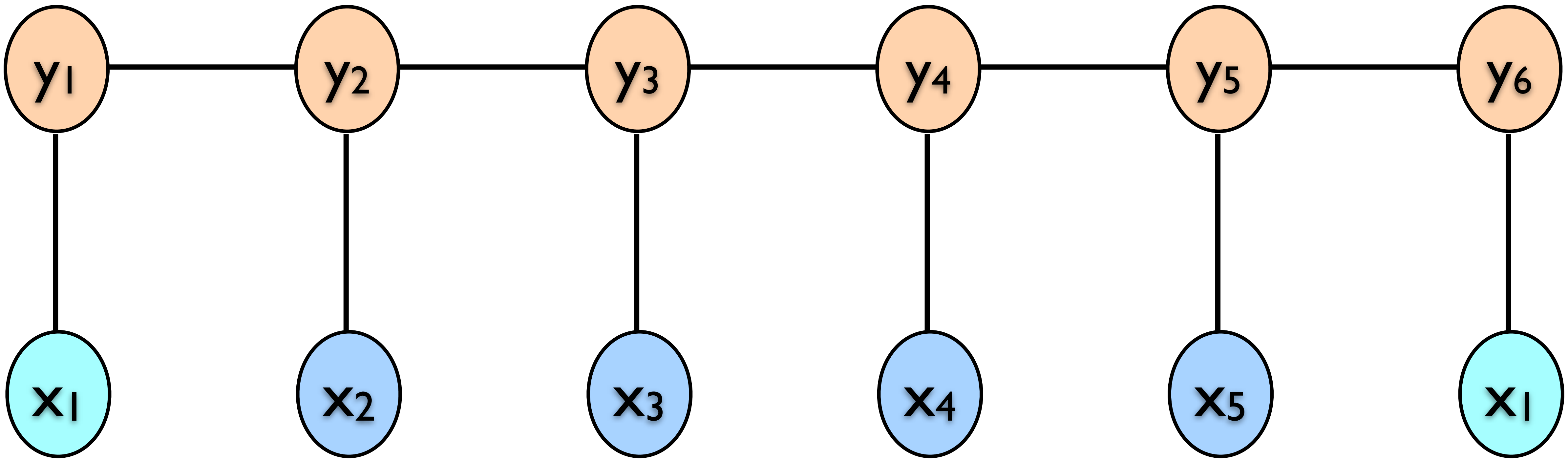
Add dependency on length of segment



Can also have small number of global dependencies:
'at least one verb'

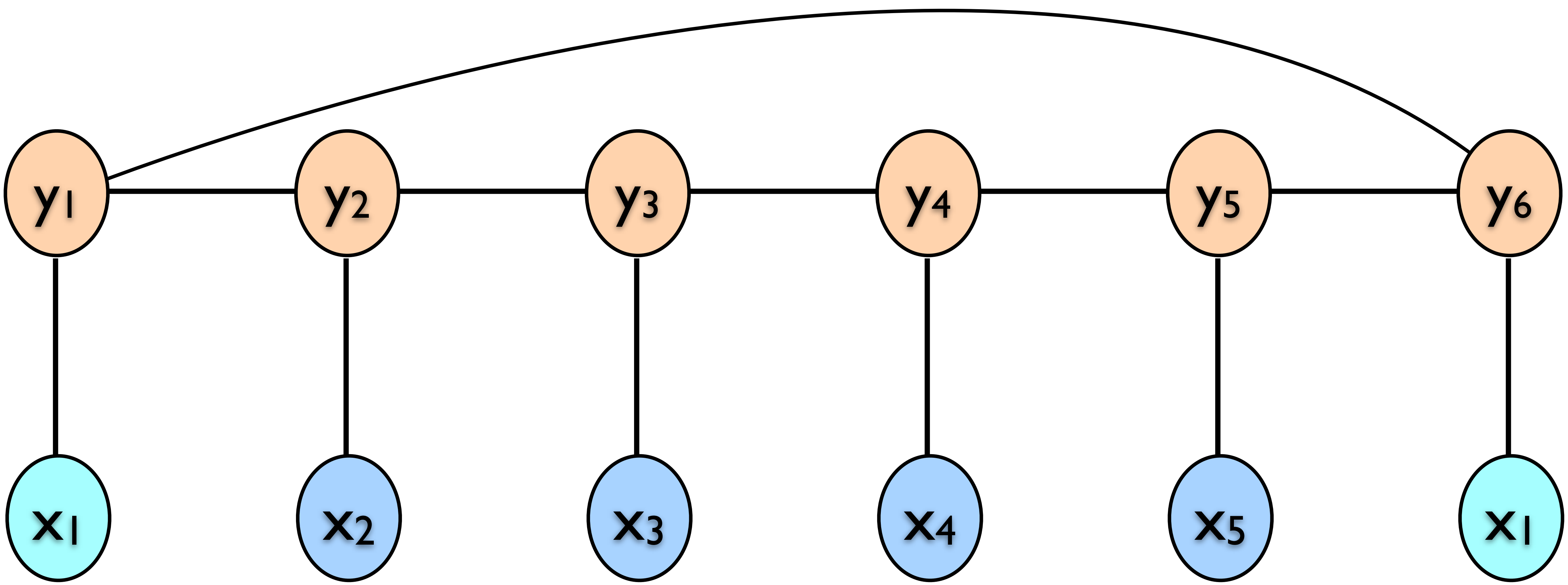
Skip-Chain CRF

Encourage repeated words to receive the same label



Skip-Chain CRF

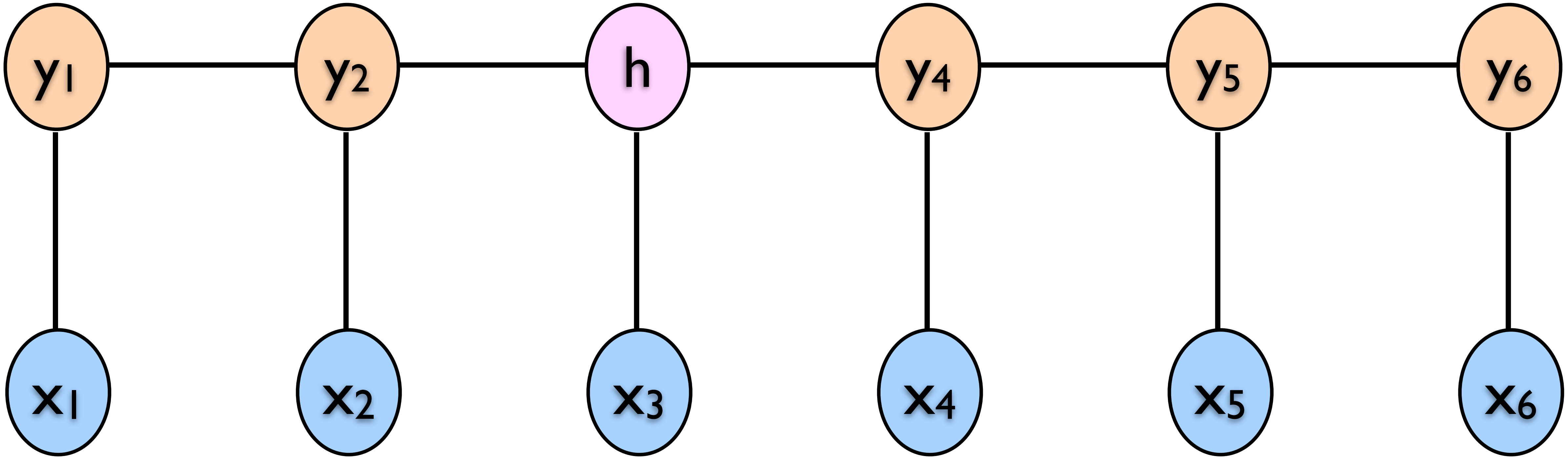
Encourage repeated words to receive the same label



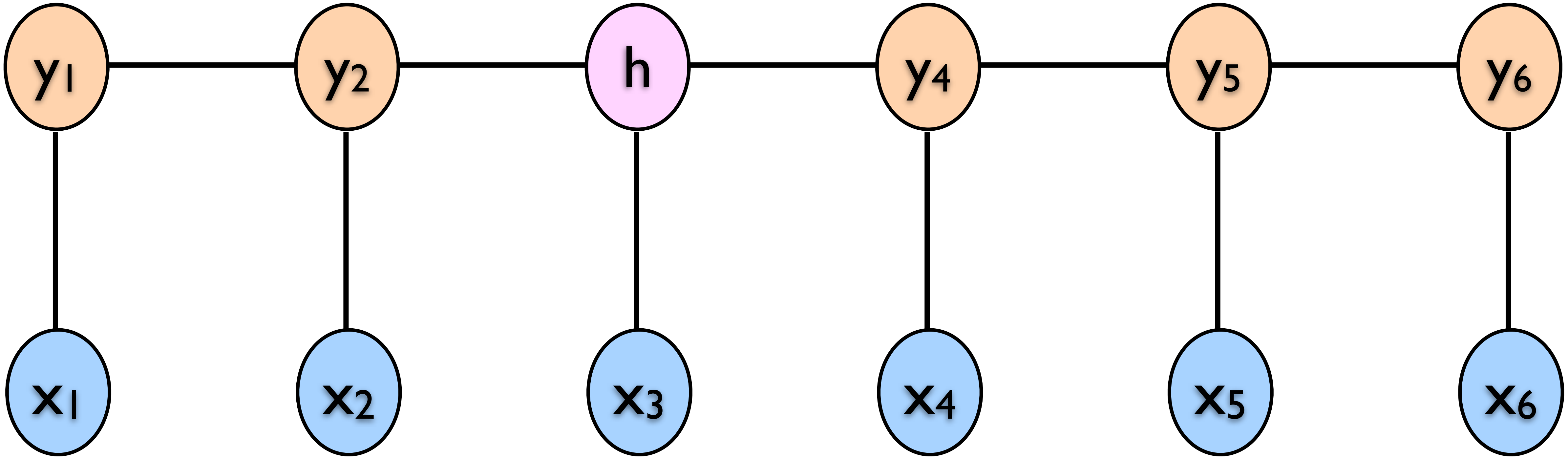
Outline

- ~~Overview of General Conditional Random Fields~~
- Conditional Random Fields with Latent Variables

Missing Labels

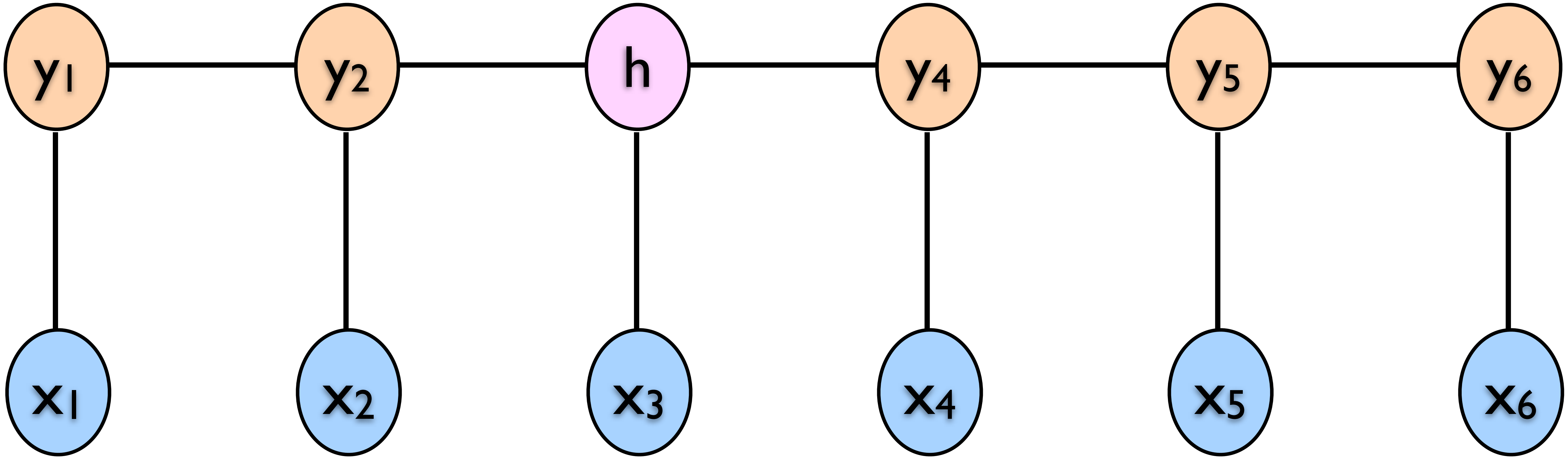


Missing Labels



$$\boxed{p(y_{1:2}=s_{1:2}, h=s_3, y_{4:N}=s_{4:N}|x,w) \propto \prod_{n=1}^{N} \exp(w_{s_n}^T x_n) \prod_{n=0}^{N} \exp(v_{s_n,s_{n+1}})} = \frac{f(y,h)}{\sum_{y',h} f(y',h)}$$

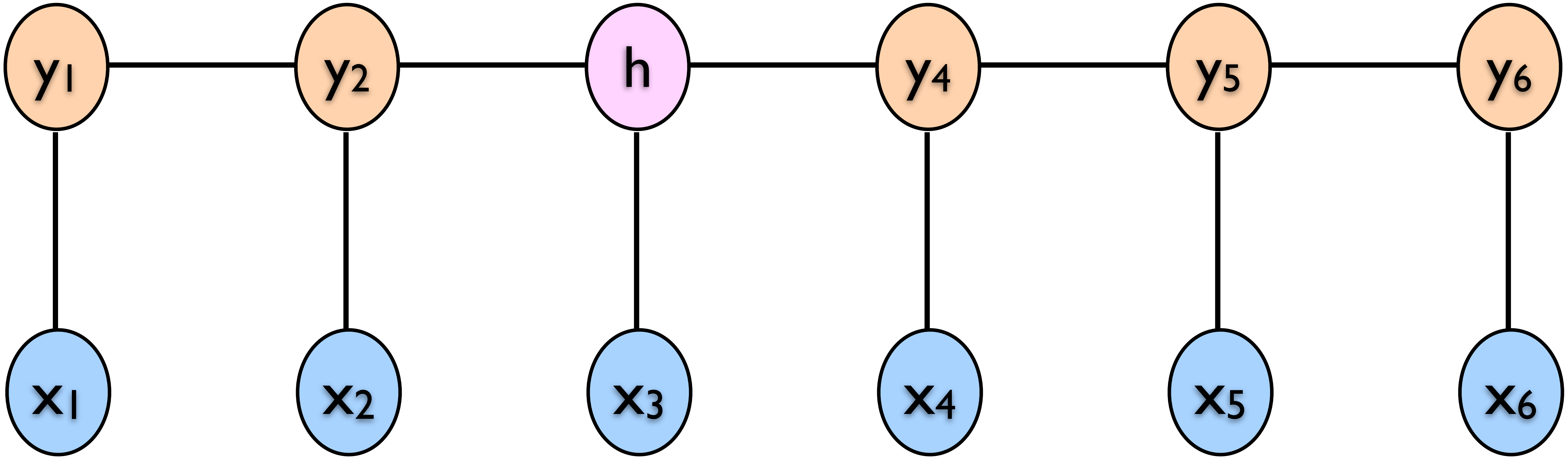
Missing Labels



$$\boxed{p(y_{1:2} = s_{1:2}, h = s_3, y_{4:N} = s_{4:N} | x, w) \propto \prod_{n=1}^{N} \exp(w_{s_n}^T x_n) \prod_{n=0}^{N} \exp(v_{s_n, s_{n+1}})} = \frac{f(y, h)}{\sum_{y', h} f(y', h)}$$

$$\boxed{p(y|x, w) = \sum_h p(y, h | x, w)} = \frac{\sum_h f(y, h')}{\sum_{y', h'} f(y', h')}$$

Missing Labels



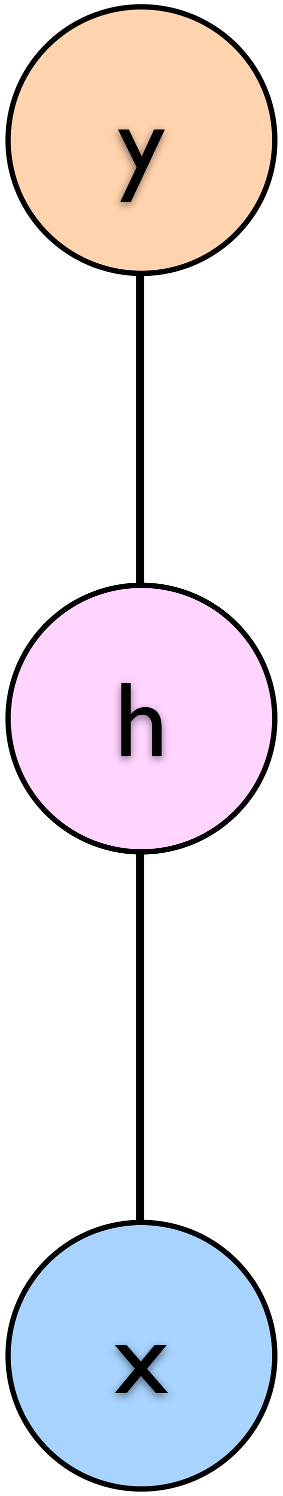
$$\boxed{p(y_{1:2} = s_{1:2}, h = s_3, y_{4:N} = s_{4:N} | x, w) \propto \prod_{n=1}^{N} \exp(w_{s_n}^T x_n) \prod_{n=0}^{N} \exp(v_{s_n, s_{n+1}})} = \frac{f(y, h)}{\sum_{y', h} f(y', h)}$$

$$\boxed{p(y|x, w) = \sum_h p(y, h|x, w)} = \frac{\sum_h f(y, h')}{\sum_{y', h'} f(y', h')}$$

If all variables hidden, cancels out

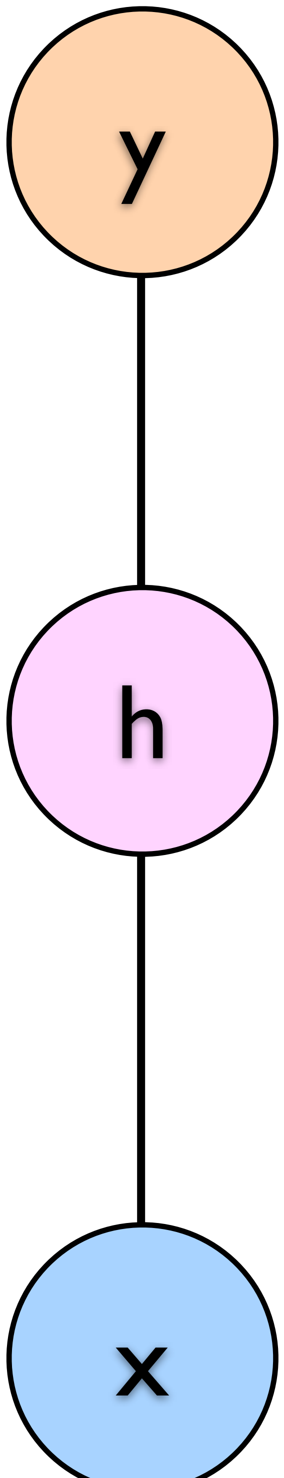
Numerator leads to non-convex optimization

Latent Logistic Regression



Latent logistic: class variables
have unknown sub-classes

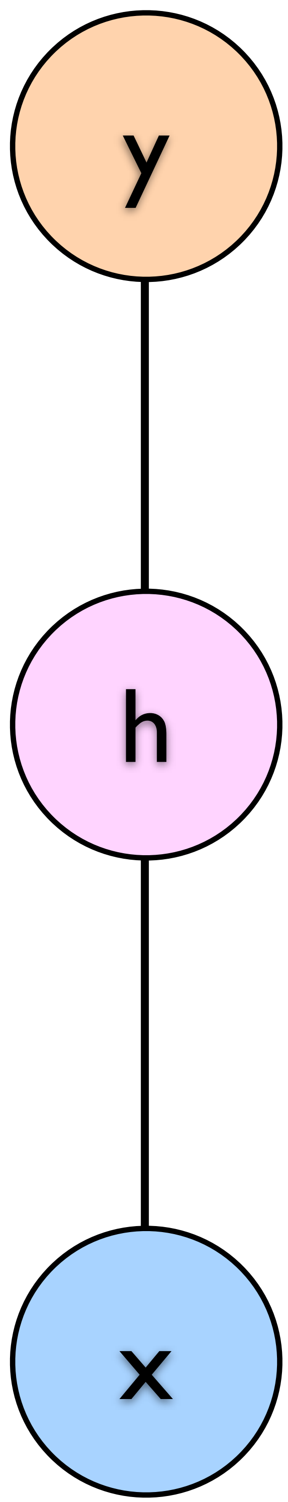
Latent Logistic Regression



Latent logistic: class variables have unknown sub-classes

$$h \in \{\underbrace{1,2,3}_{\text{Class 1}}, \underbrace{4,5,6}_{\text{Class 2}}, \underbrace{7,8,9}_{\text{Class 3}}\}$$

Latent Logistic Regression

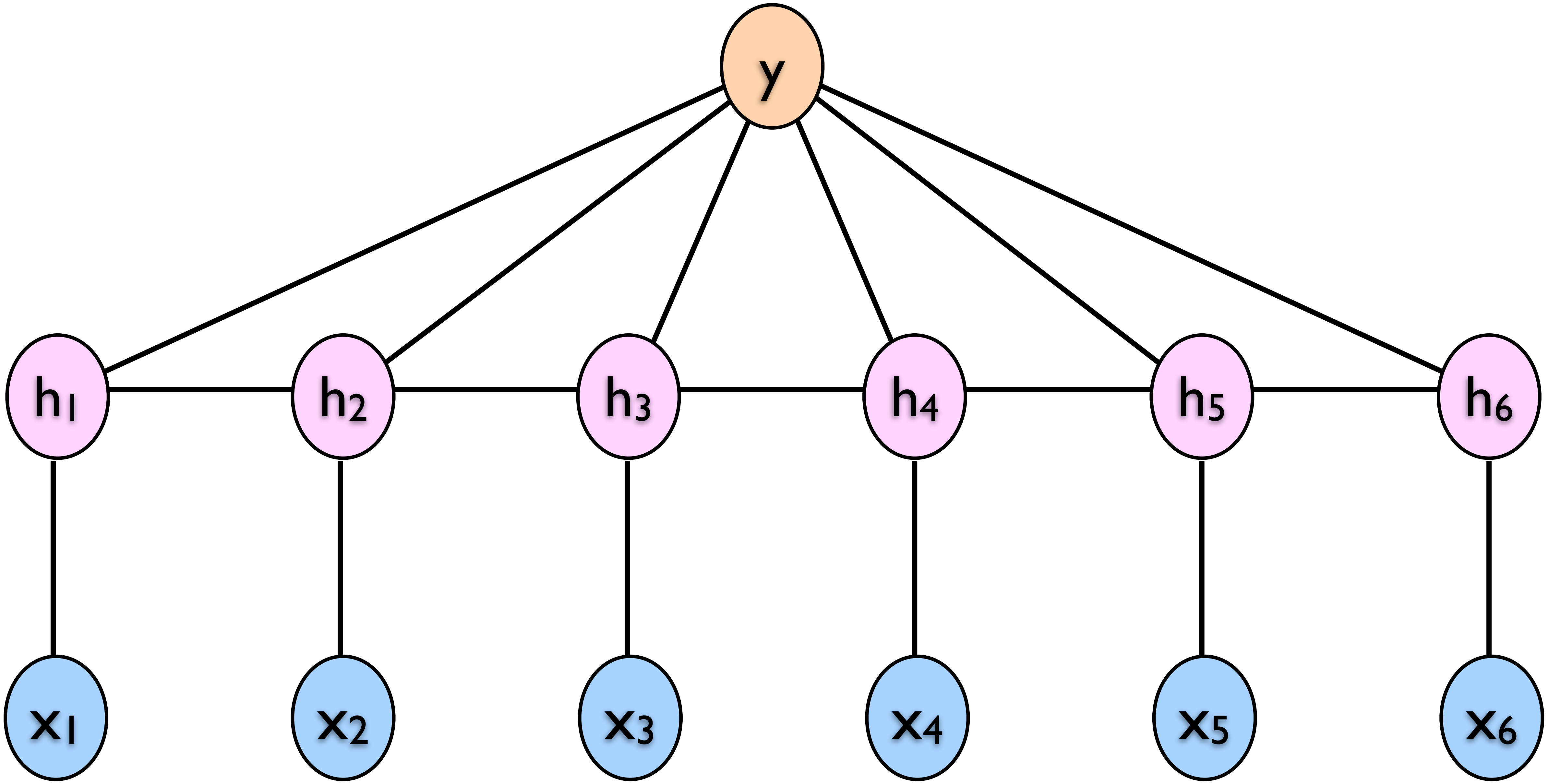


Latent logistic: class variables have unknown sub-classes

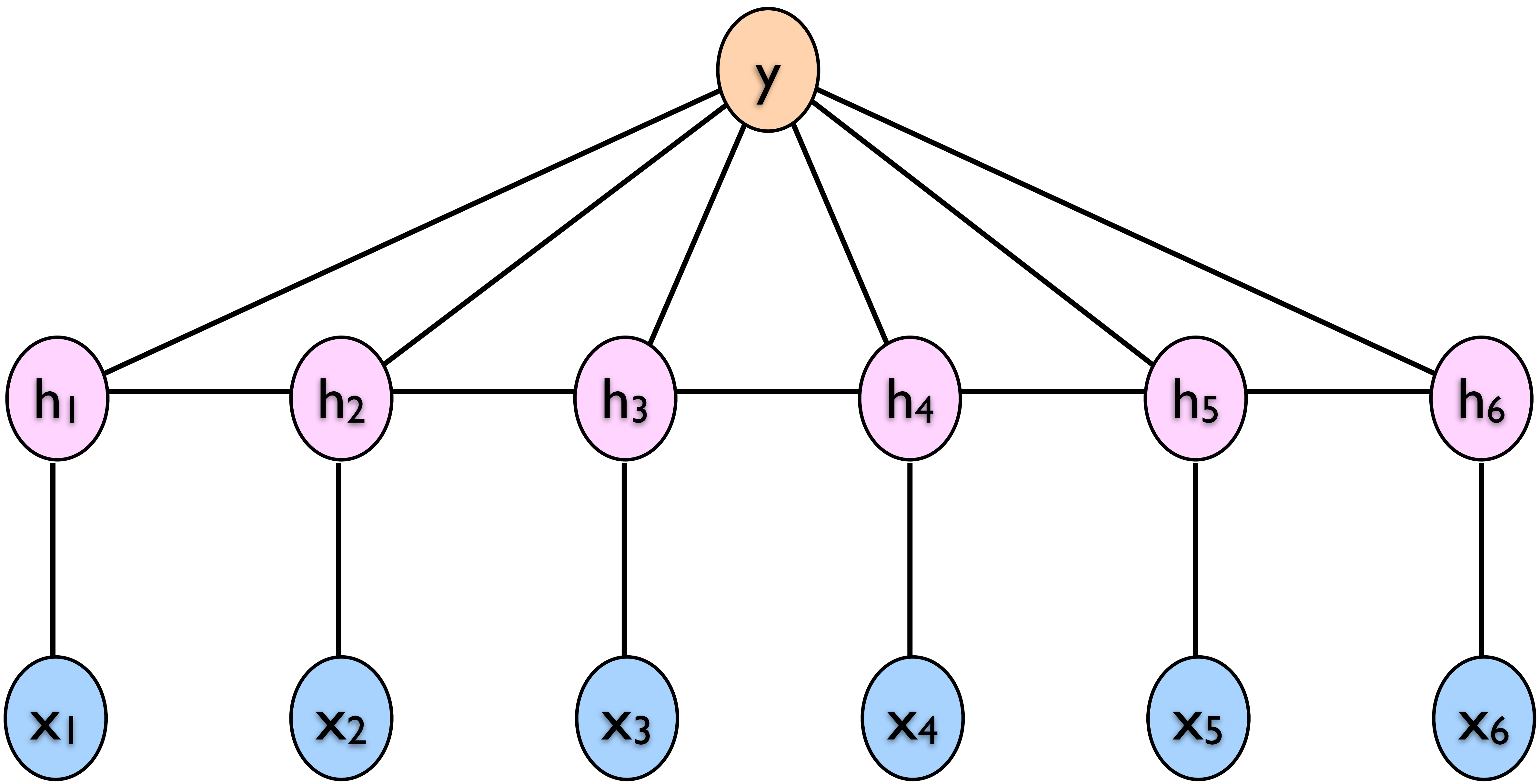
$$h \in \{\underbrace{1,2,3}_{\text{Class 1}}, \underbrace{4,5,6}_{\text{Class 2}}, \underbrace{7,8,9}_{\text{Class 3}}\}$$

$$\boxed{p(y = s|x, w) \propto \sum_{h \in s} \exp(w_h^T x)}$$

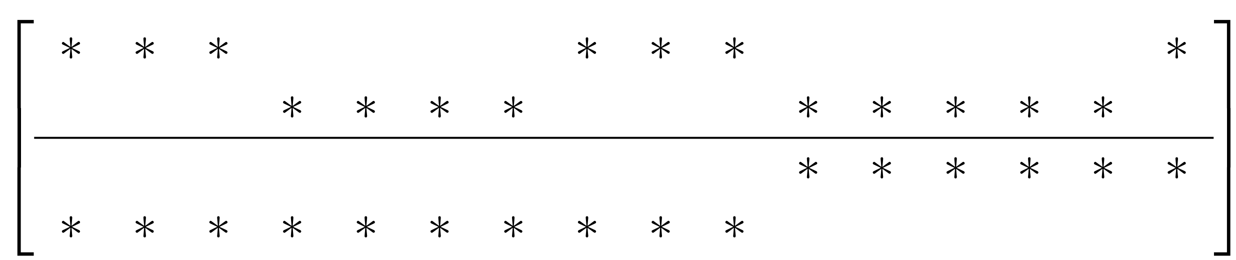
Hidden CRF



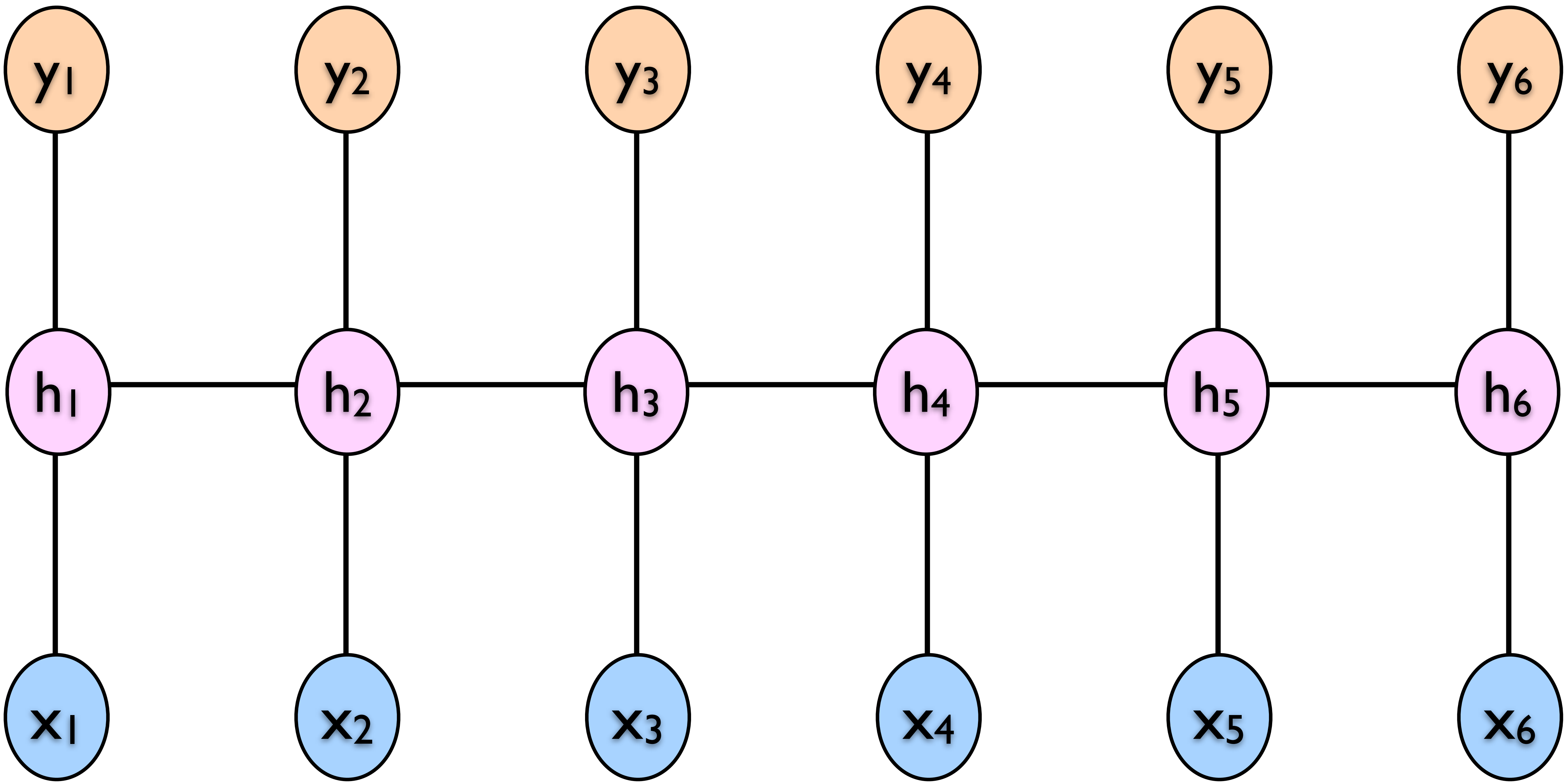
Hidden CRF



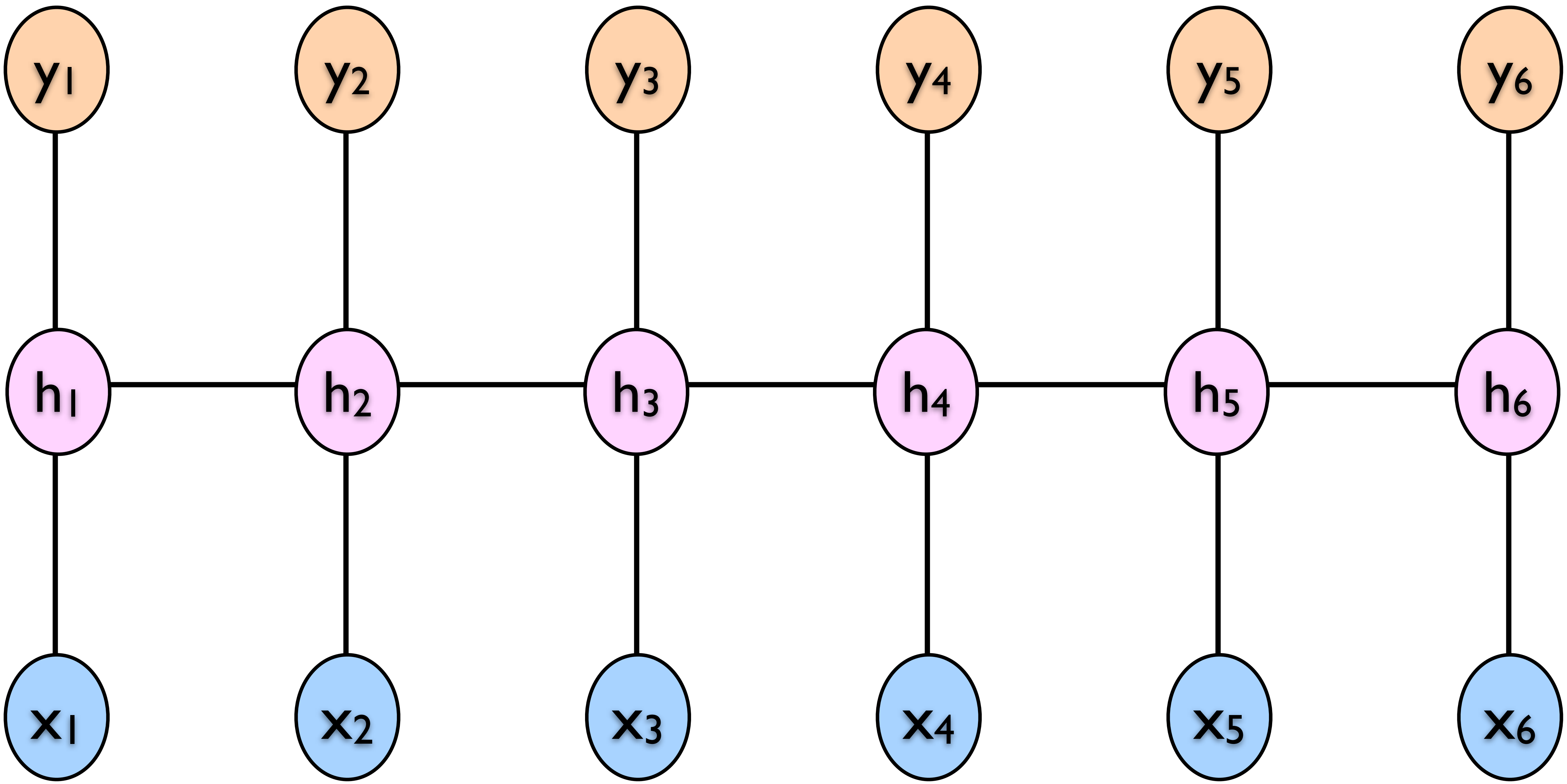
An HMM with a supervised label



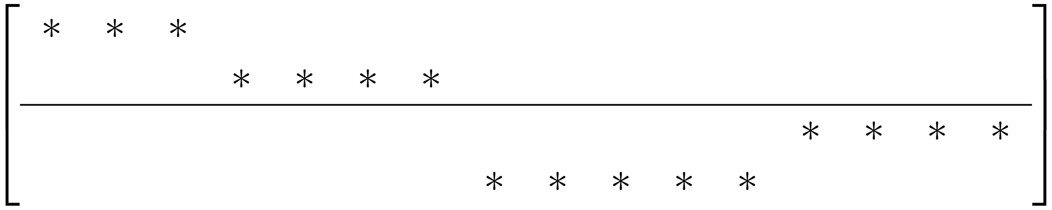
Latent Dynamic CRF



Latent Dynamic CRF

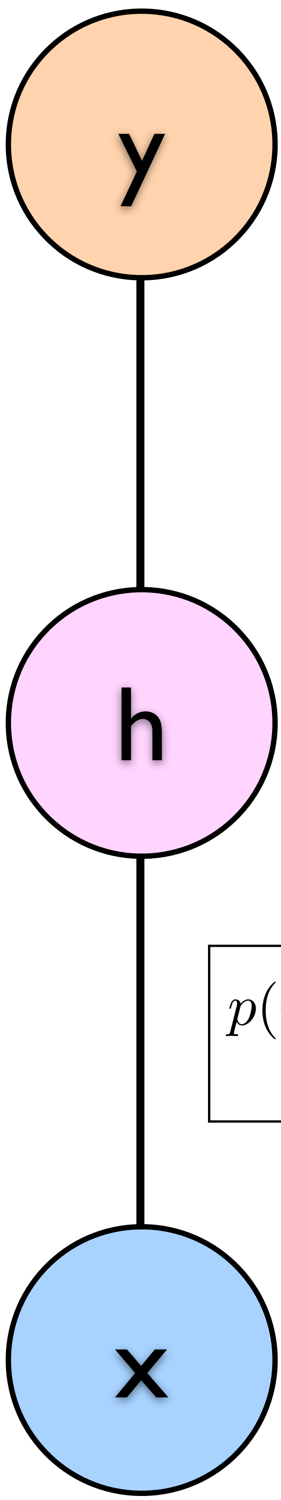


Hidden dependency structure among sub-classes



Latent Logistic Regression and Neural Networks

Latent logistic: class variables have unknown sub-classes

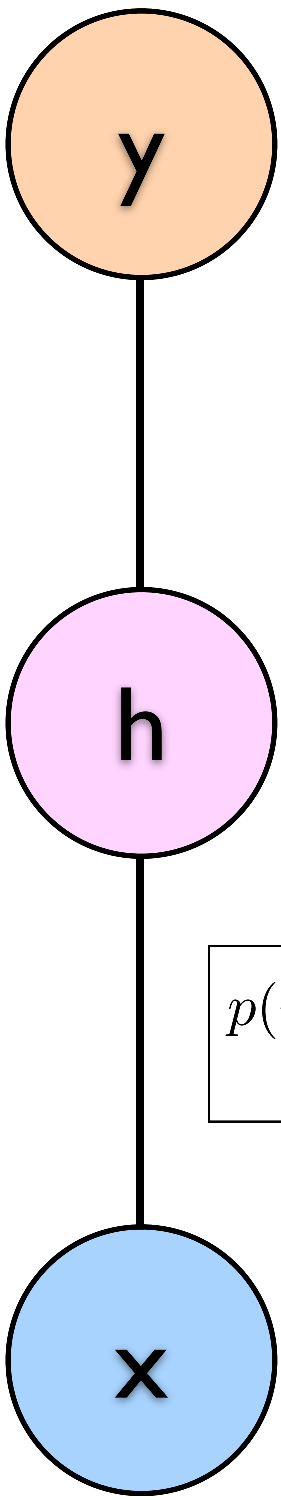


$$h \in \{\underbrace{1,2,3}_{\text{Class 1}},\underbrace{4,5,6}_{\text{Class 2}},\underbrace{7,8,9}_{\text{Class 3}}\}$$

$$\boxed{p(y=s|x,w) \propto \sum_{h \in s} \exp(w_h^T x)}$$

Latent Logistic Regression and Neural Networks

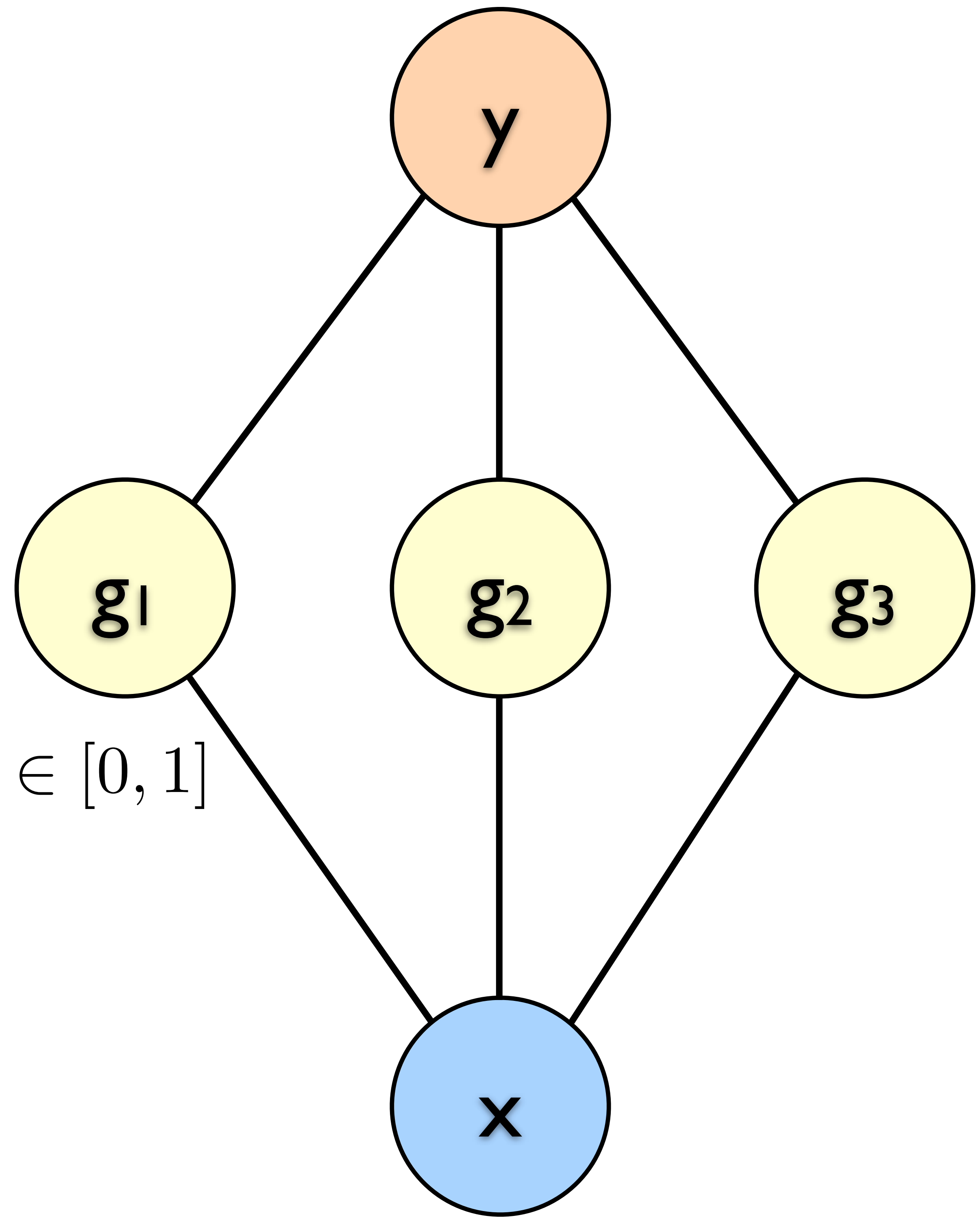
Latent logistic: class variables have unknown sub-classes



$$h \in \{\underbrace{1,2,3}_{\text{Class 1}},\underbrace{4,5,6}_{\text{Class 2}},\underbrace{7,8,9}_{\text{Class 3}}\}$$

$$\boxed{p(y=s|x,w) \propto \sum_{h \in s} \exp(w_h^T x)}$$

Neural network: combine non-linear transformations to binary variables

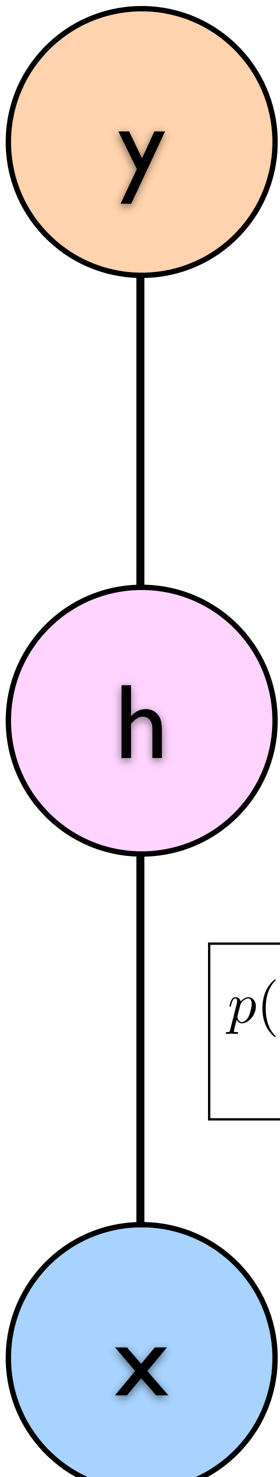


$g_i \in [0,1]$

Latent Logistic Regression and Neural Networks

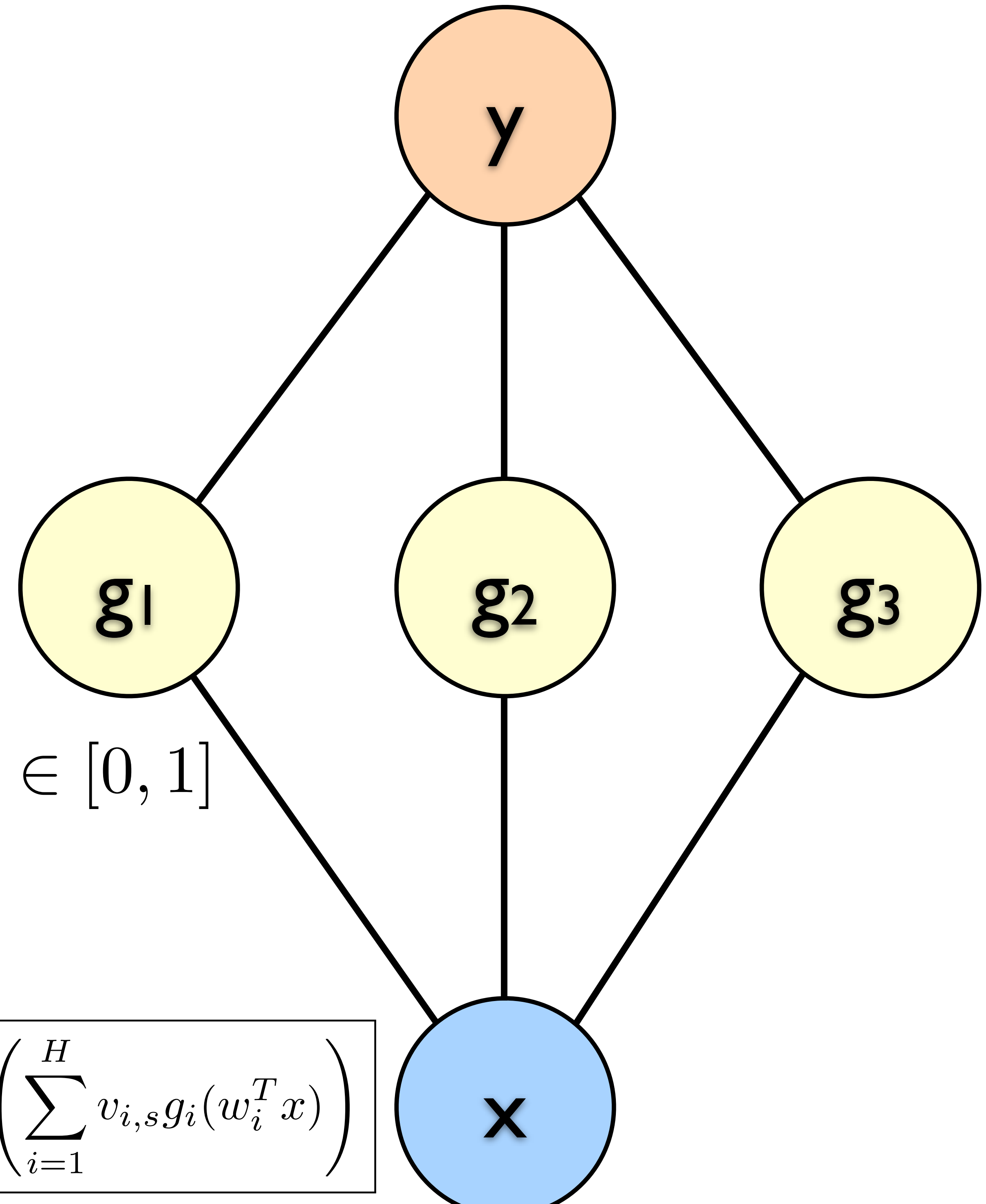
Latent logistic: class variables have unknown sub-classes

Neural network: combine non-linear transformations to binary variables



$h \in \{\underbrace{1,2,3}_{\text{Class 1}}, \underbrace{4,5,6}_{\text{Class 2}}, \underbrace{7,8,9}_{\text{Class 3}}\}$

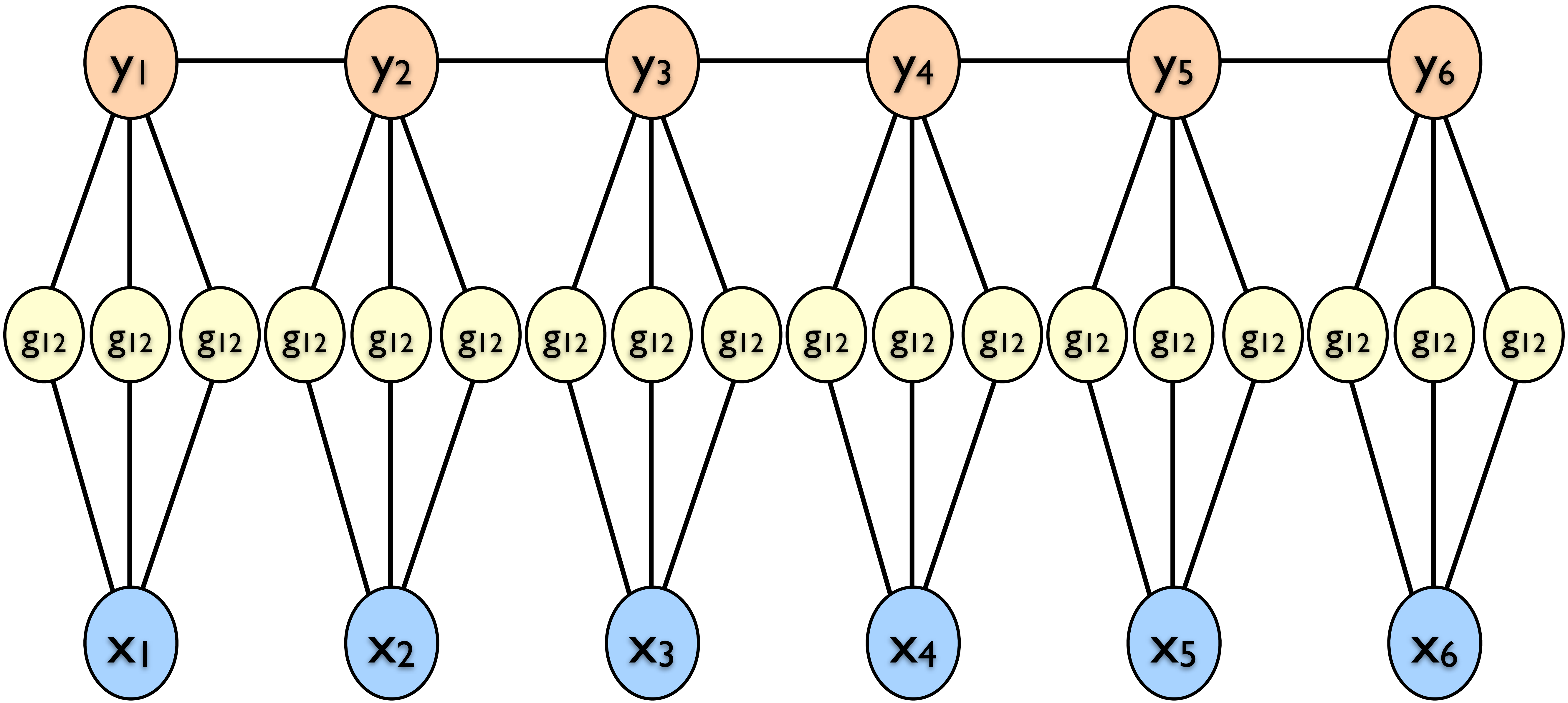
$$\boxed{p(y=s|x,w) \propto \sum_{h \in s} \exp(w_h^T x)}$$



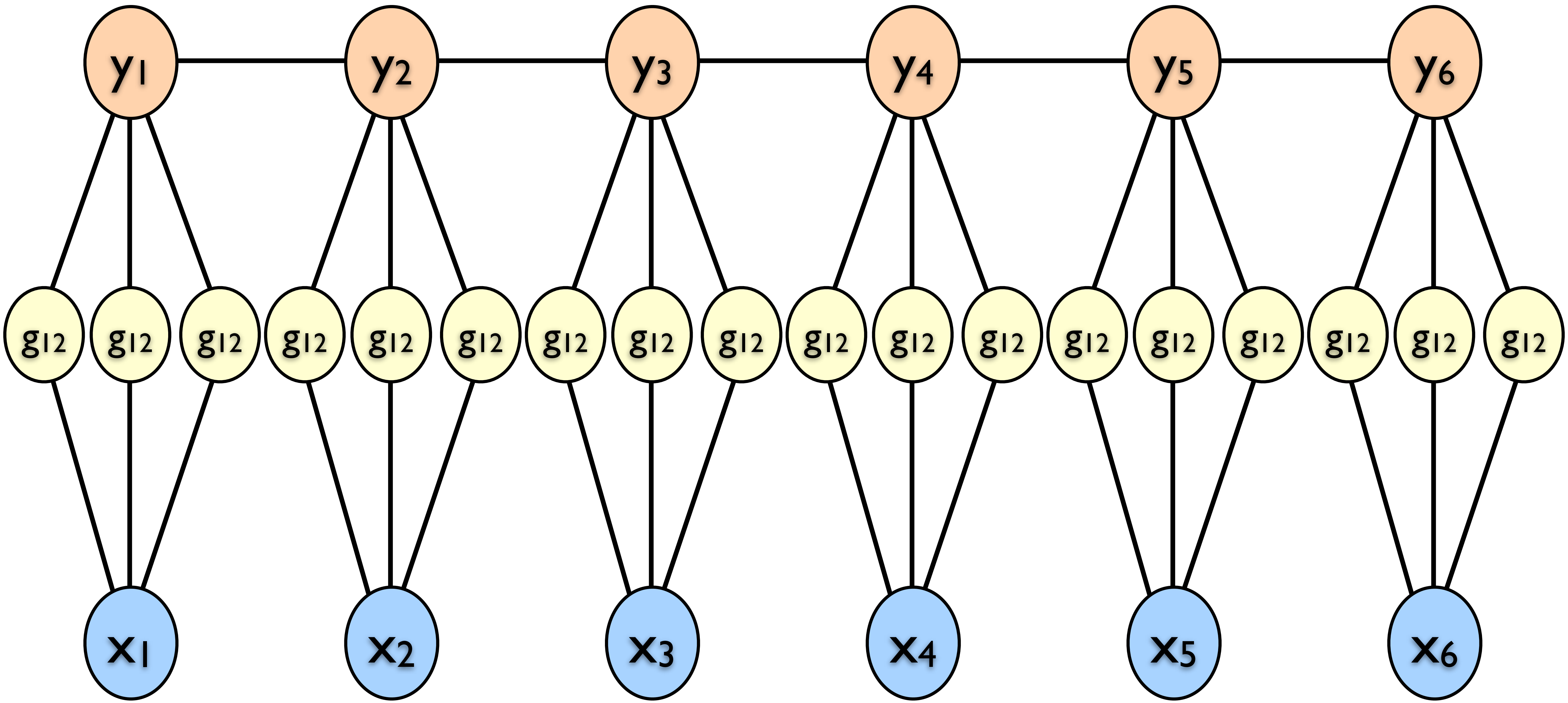
$g_i \in [0,1]$

$$\boxed{p(y=s|x,w) \propto \exp\left(\sum_{i=1}^{H} v_{i,s} g_i(w_i^T x)\right)}$$

Hidden-Unit CRF, Conditional Neural Field (CNF)



Hidden-Unit CRF, Conditional Neural Field (CNF)



A standard CRF where we learn the features

Related to earlier support vector random fields

Latent Dynamic CNF