GM Application: Sequence Labeling

Wang Houfeng
Institute of Computational Linguistics
Peking University

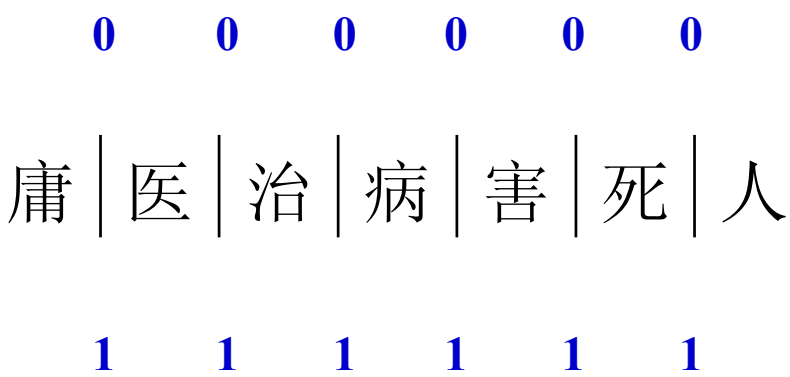
Outline

- **Sequence Labeling**
- HMM
- MEMM
- CRF

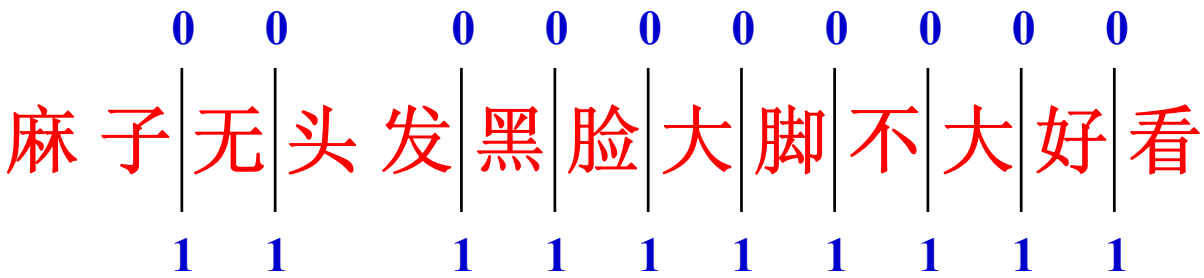
Sequence Label

- Many NLP problems can viewed as sequence labeling.
- Each token in a sequence is assigned a label.
- Labels of tokens are dependent on the labels of other tokens in the sequence, particularly their neighbors.

Word Segmentation



Sentences Segmentation



Information Extraction

- Identify phrases in language that refer to specific types of entities and relations in text.
- Named entity recognition is task of identifying names of people, places, organizations, etc. in text.

people **organizations** **places**

– 比尔.盖茨 创建了微软公司，近年多次到中国访问.

Semantic Role Labeling

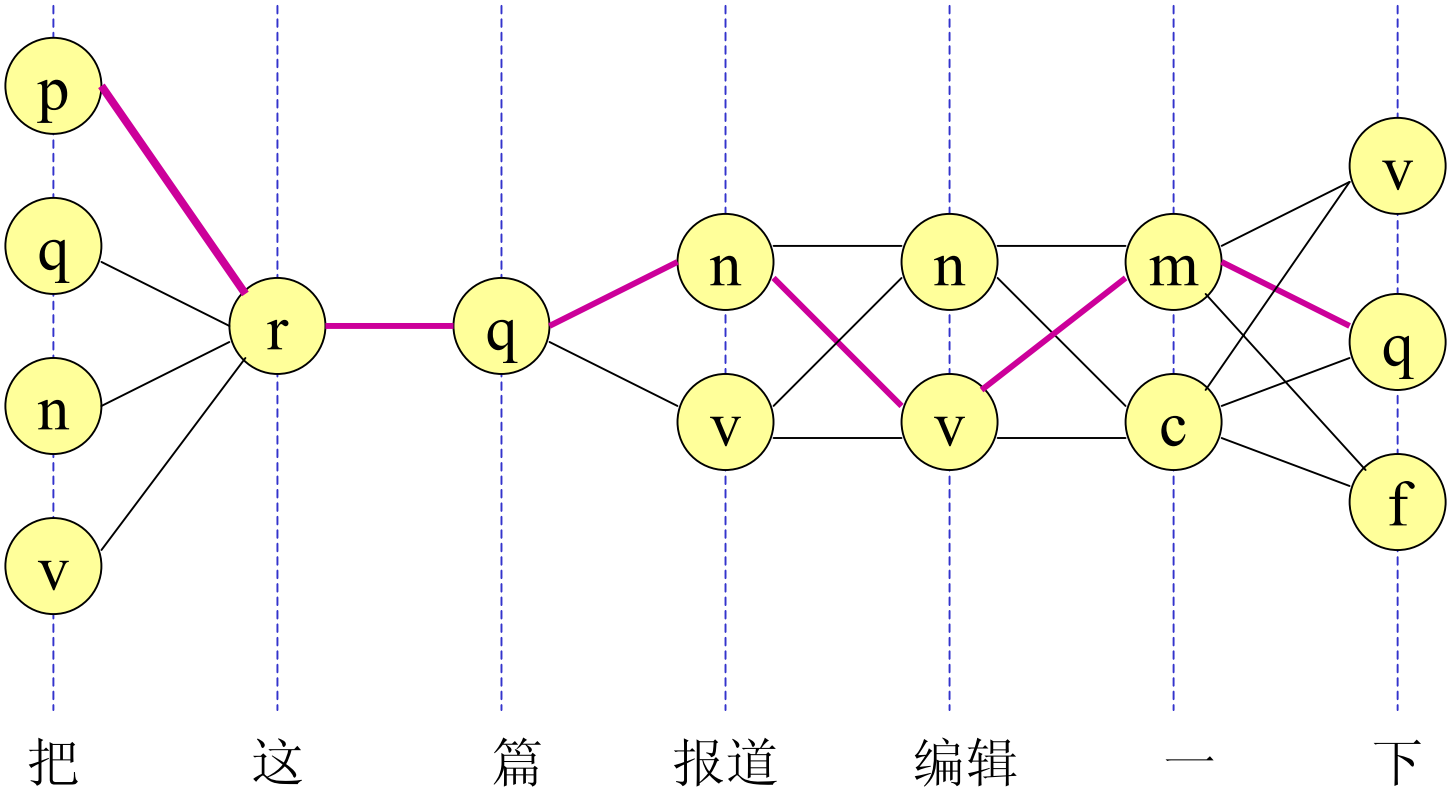
- For each clause, determine the semantic role played by each noun phrase that is an argument to the verb.

  agent **patient** **source** **destination** **instrument**

  – 张先生驾驶他的奥迪把夫人从家送到医院.

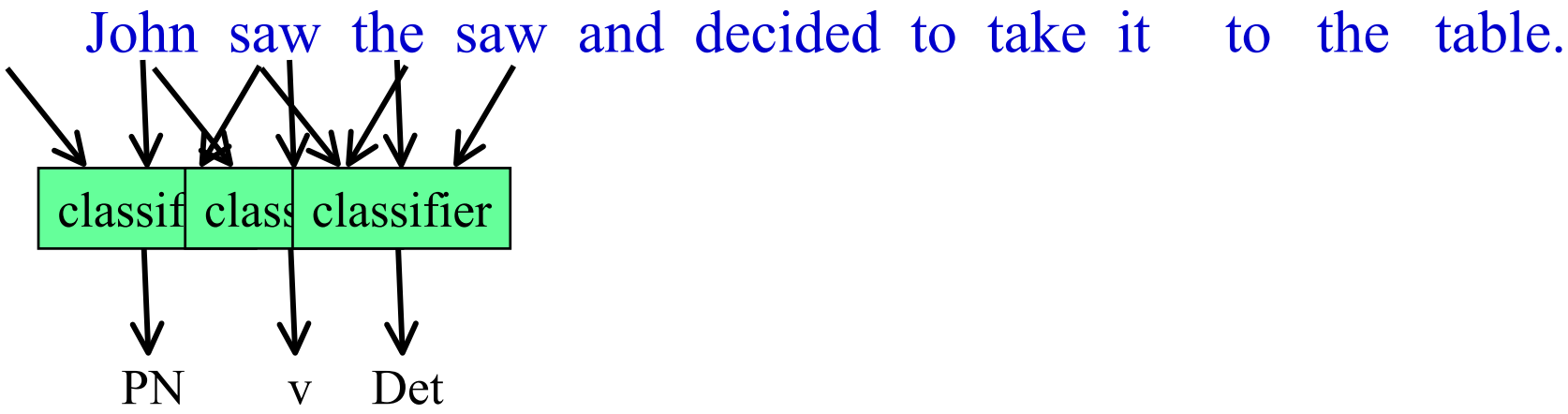
- Also referred to a “case role analysis,” “thematic analysis,” and “shallow semantic parsing”

POS



Sequence Labeling as Classification

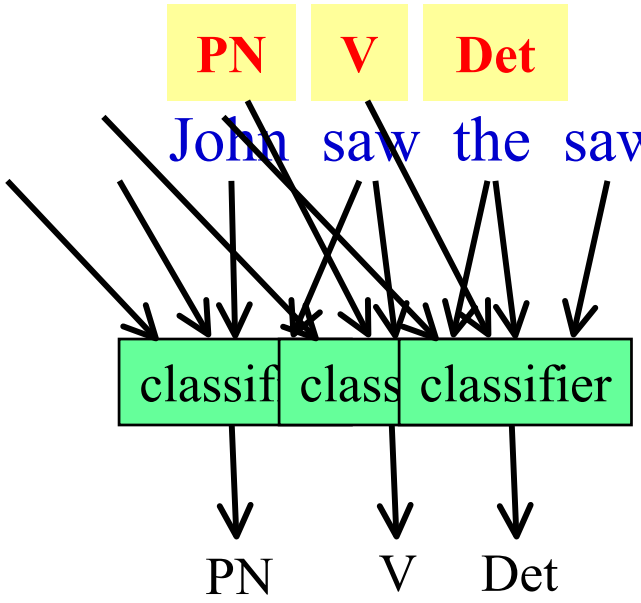
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window)



Sequence Labeling as Classification Using Outputs as Inputs

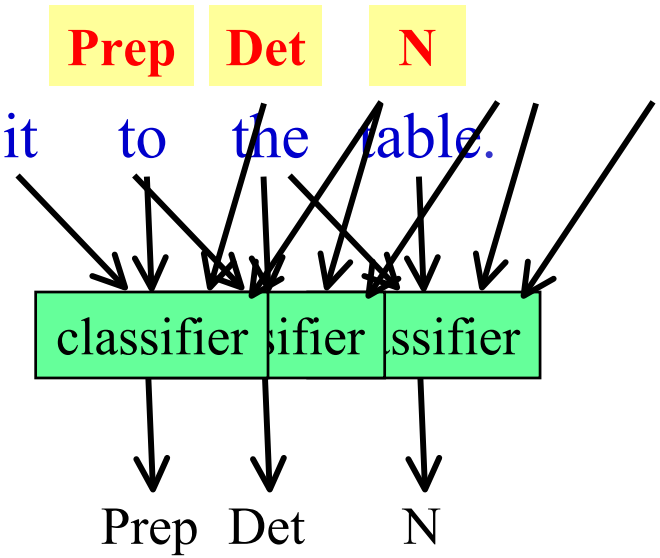
- Better input features are usually the categories of the surrounding tokens, but these are not available yet.
- Can use category of either the preceding or succeeding tokens by going forward or back and using previous output.

Forward Classification



Backward Classification

John saw the saw and decided to take it to the table.

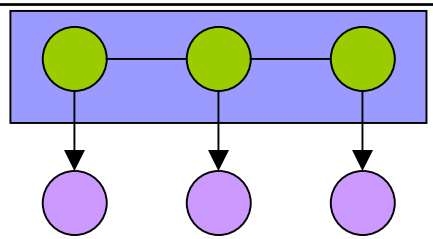
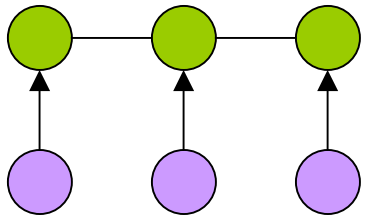


- Disambiguating “to” in this case would be even easier backward.

Probabilistic Sequence Models

- Probabilistic sequence models allow integrating uncertainty over multiple, interdependent classifications and collectively determine the most likely global assignment.
- Probabilistic Graphical models in this problem:
  - Hidden Markov Model (HMM): directed GM
  - MaxEnt Model Markov (MEMM) : directed GM
  - Conditional Random Field (CRF): undirected GM

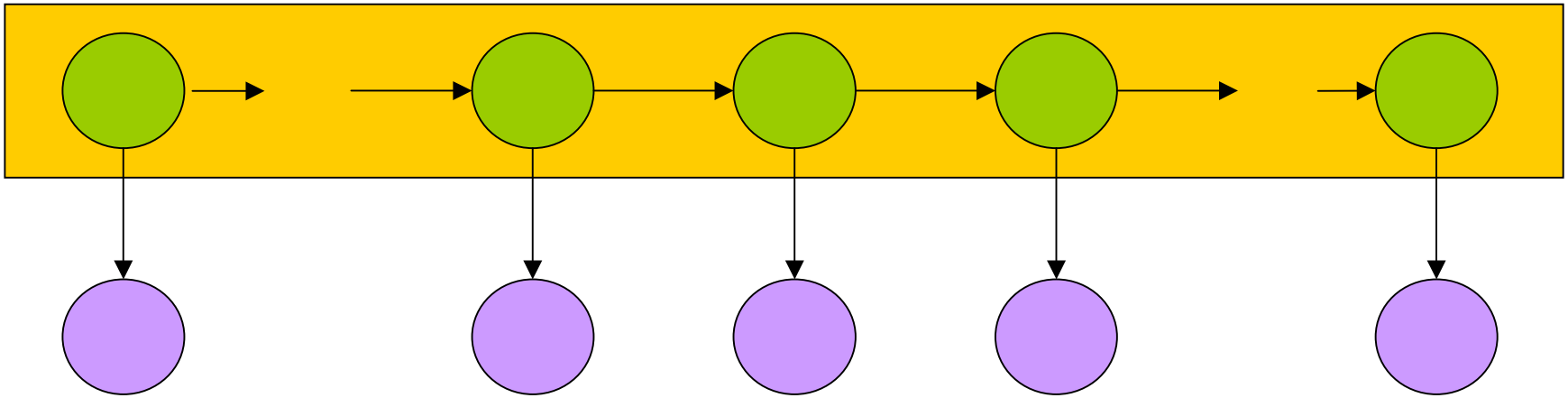
Generative vs. Discriminative

Generative	Discriminative
$P(O \mid S)$	$P(S \mid O)$
Example: HMM	Example: MaxEnt, MEMM, CRF
	
Learning = finding likelihood of observation sequence given state sequence	Learning = finding difference between state sequences given observation sequence
Tagging = finding most likely state sequence having generated given observation sequence	Tagging = finding most likely state sequence mapped from given observation sequence

Outline

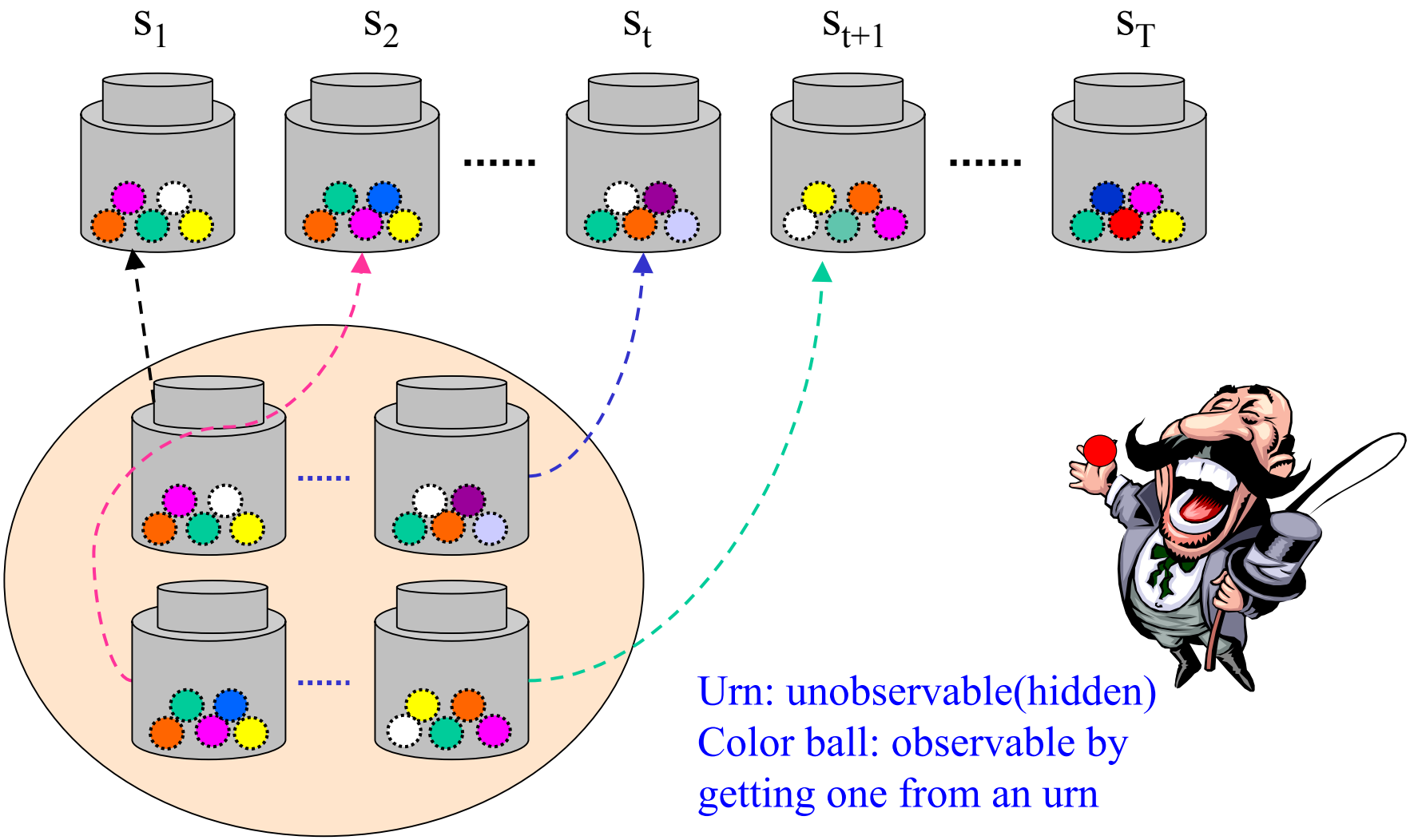
- Sequence Labeling
- **HMM**
- MEMM
- CRF

What is HMM?



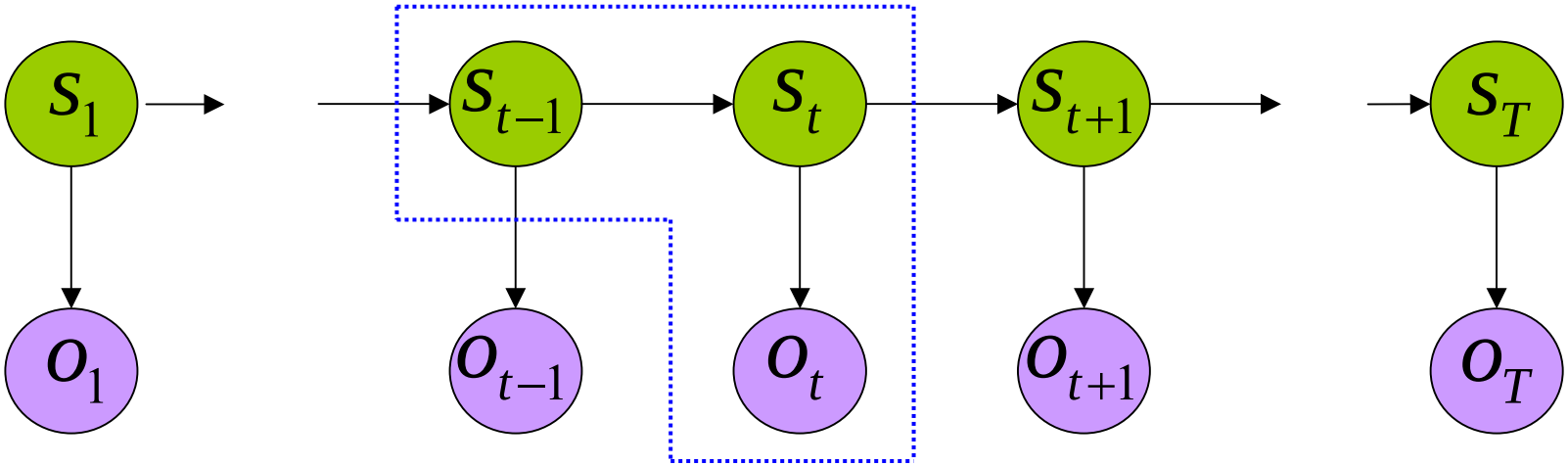
- Green nodes are ***'hidden' states***
- State depends only on previous state
- Purple nodes are ***observations***
- Each state generates an observation

Color Ball and Urn



Urn: unobservable(hidden)
Color ball: observable by getting one from an urn

HMM Formalism

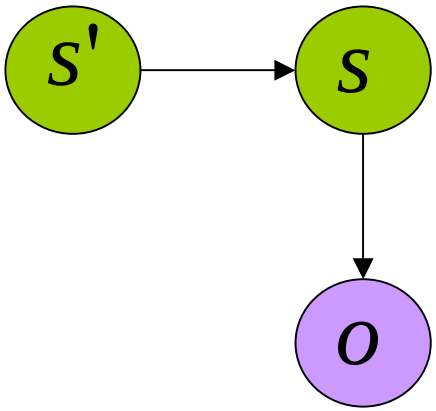


- According to Directed GM

$$P(O,S) = P(s_1).\prod_{t=1}^{T} P(o_t \mid s_t).\prod_{t=2}^{T} P(s_t \mid s_{t-1})$$

- Each state generates an observation

HMM Formalism



$$P(s \mid s')$$

$$P(o \mid s)$$

CPT

HMM Formalism

- States: $\{y_1, y_2, \ldots, y_N\}$
- State transition probabilities:

$$A=\{a_{ij} \mid a_{ij} = P(s_t = y_j \mid s_{t-1} = y_i)\}$$

- Initial state distribution:

$$\pi_i = P[s_1 = y_i]$$

prob of moving from state i to state j

- Observations:

$$\{x_1, x_2, \ldots, x_M\}$$

- Observation probabilities:

$$B=\{b_{jk} \mid b_{jk} = P(o_t = x_k \mid s_t = y_j)\}$$

emit output k in state j

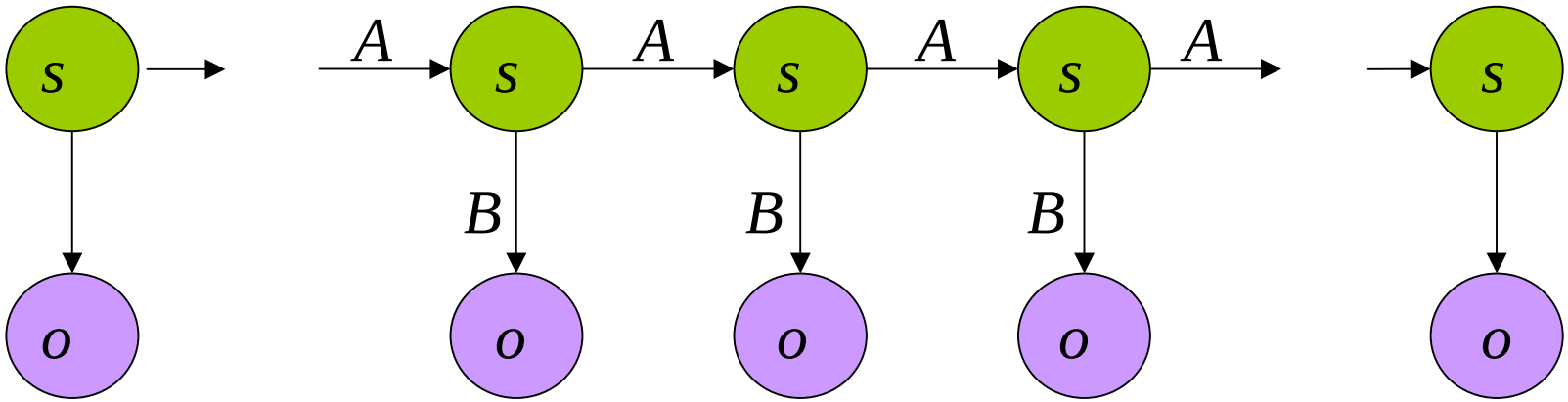
Three Classical Questions (GM)

- **Prediction** Inference: Given a model $\mu=(A, B, \Pi)$, how do we efficiently compute how likely a certain observation is, that is, $P(O|\ \mu)$
- **Tagging** Inference : Given the observation sequence O and a model μ, how do we choose a state sequence $(s_1, \ldots, s_T)$ that best explains the observations? That is $P(S\ |O, \mu)$
- **Learning** problem: given observation sequence and set of possible models, find model most likely having generated the data

Question One: Prediction

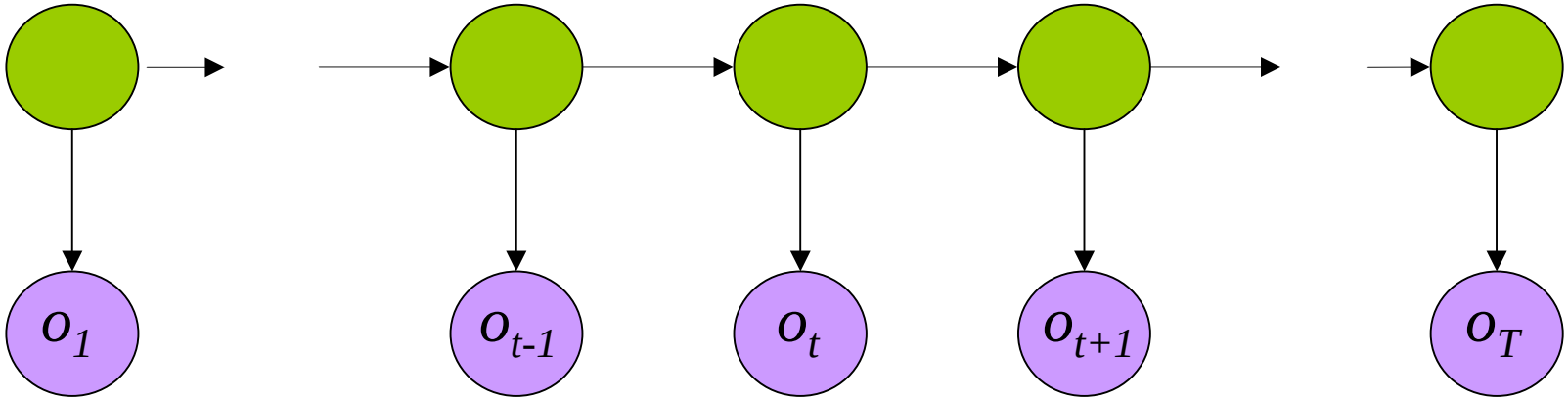
- Given the observation sequence $O=(o_1, \ldots, o_T)$ and a model $\mu= (A, B, \Pi)$, we wish to know how to efficiently compute $P(O| \mu)$.
- For any state sequence $X= (s_1, \ldots, s_T)$, we find: $P(O|\mu)=\Sigma_S P(O | S, \mu) P(S | \mu)$
- How to compute it quite efficiently?

Prediction



- $\Pi = \{\pi_\iota\}$ are initial state probabilities
- $A = \{a_{ij}\}$ are state transition probabilities
- $B = \{b_{ik}\}$ are observation state probabilities

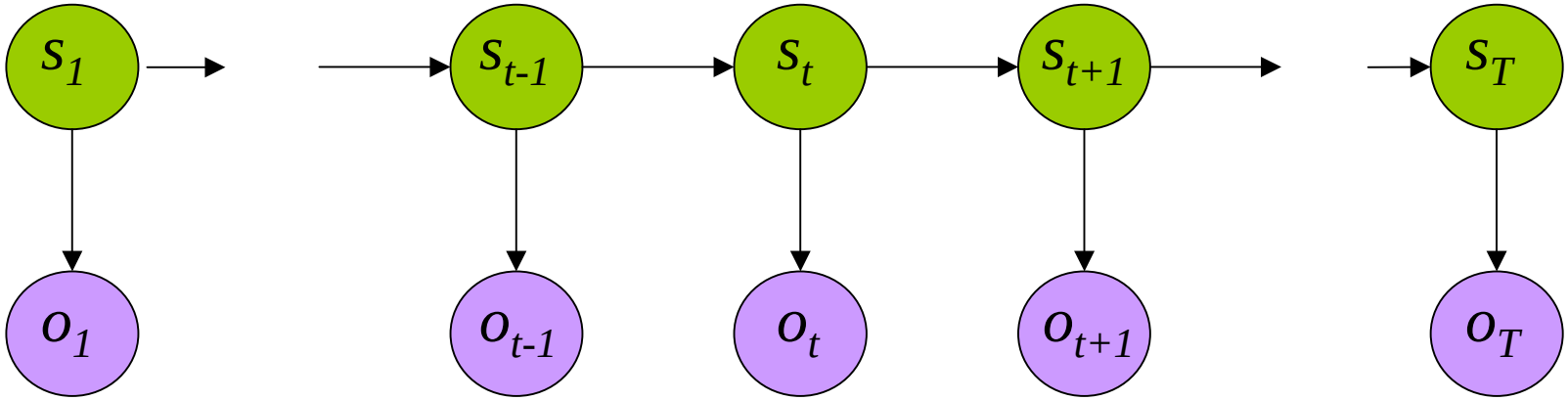
Prediction



$$O = (o_1 \ldots o_T),\ \mu = (A, B, \Pi)$$

Compute $P(O \mid \mu)$

Prediction



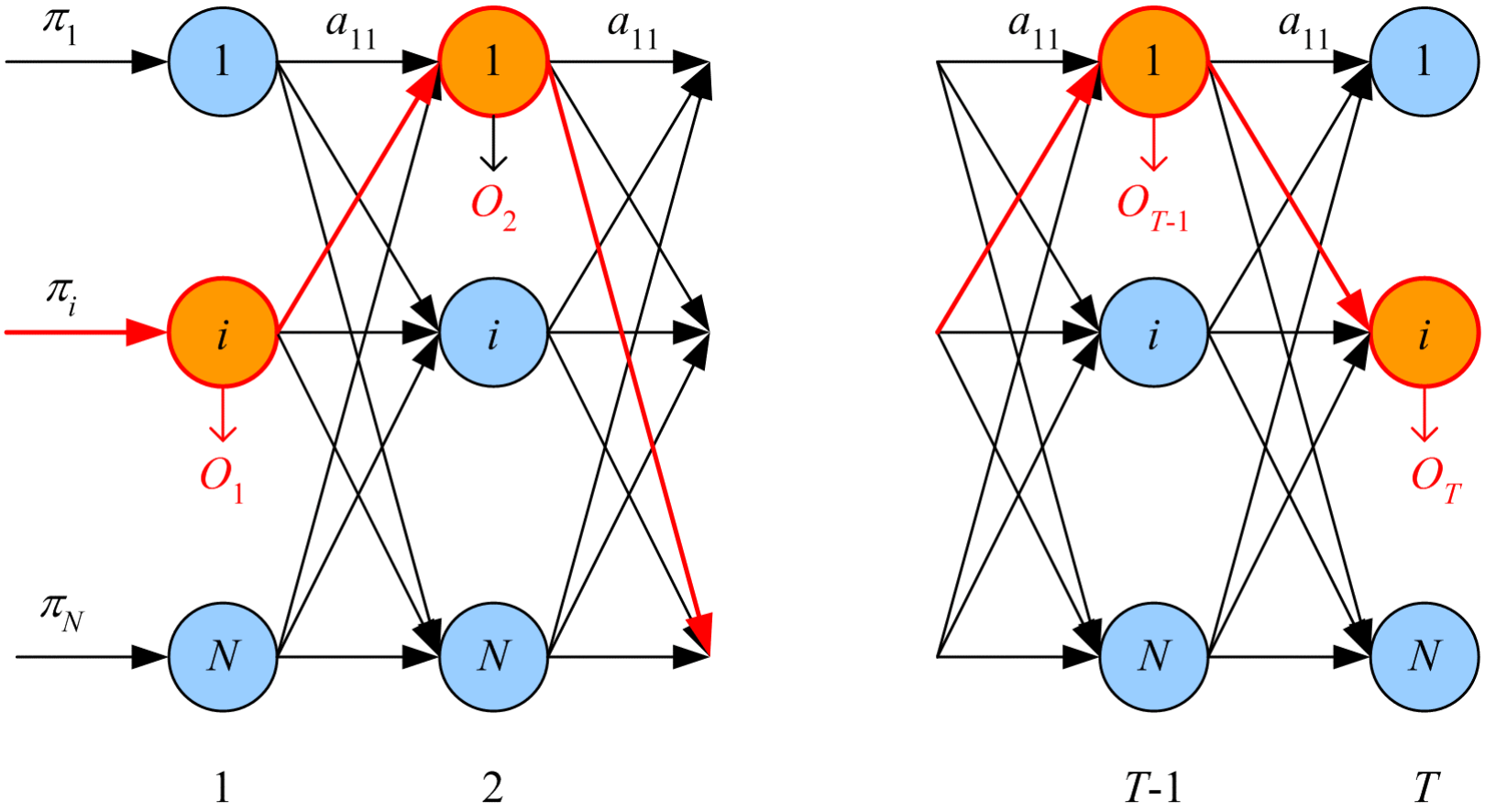
According to Bayesian Network,

$$P(O \mid \mu) = \sum_{\{s_1 \ldots s_T\}} P(S, O \mid \mu)$$

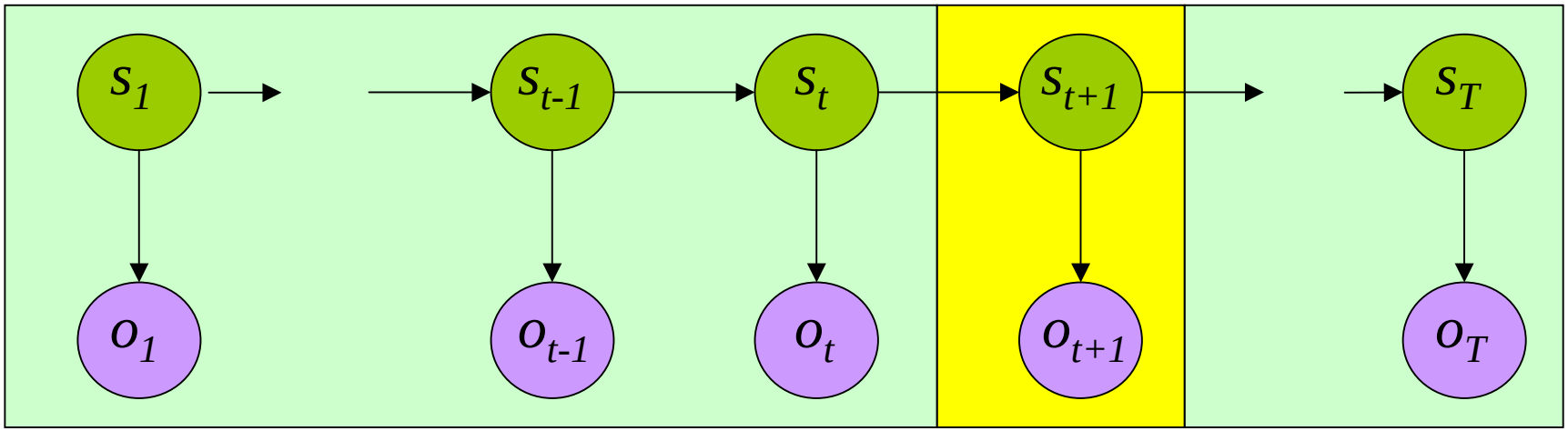
$$P(O \mid \mu) = \sum_{\{s_1 \ldots s_T\}} \pi_{s_1} \prod_{t=1}^{T} b_{s_t o_t} \cdot \prod_{t=1}^{T-1} a_{s_t s_{t+1}}$$

$$\prod P(a \mid parent(a))$$

Prediction



Forward Backward Algorithm



- Variable elimination using *dynamic programming*.

$$\alpha_t(i) = P(o_1 \ldots o_t, s_t = y_i \mid \mu) \quad , \quad \text{or}$$

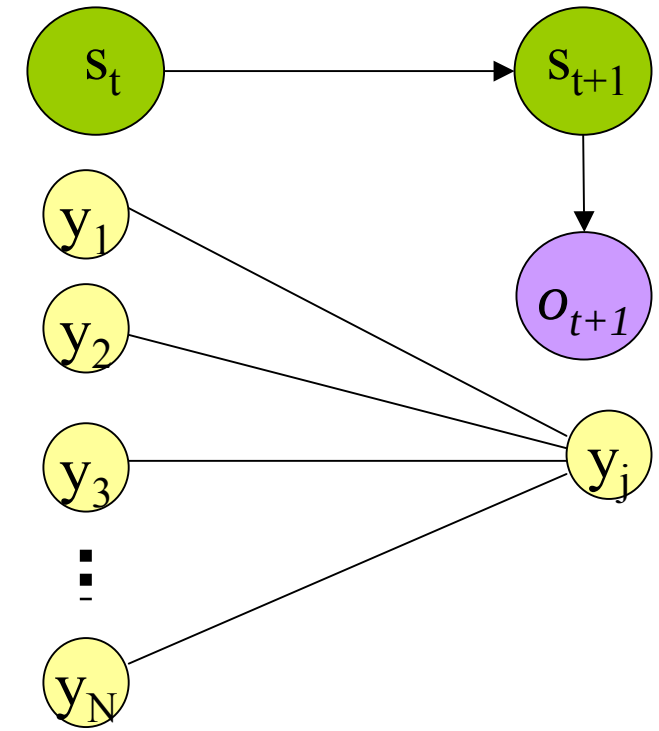
$$\alpha_t(i) = P(o_1 \ldots o_t, s_t = i \mid \mu) \quad \text{for simplicity}$$

forward

$$\beta_t(i) = P(o_t \ldots o_T \mid s_t = i, \mu)$$

Backward

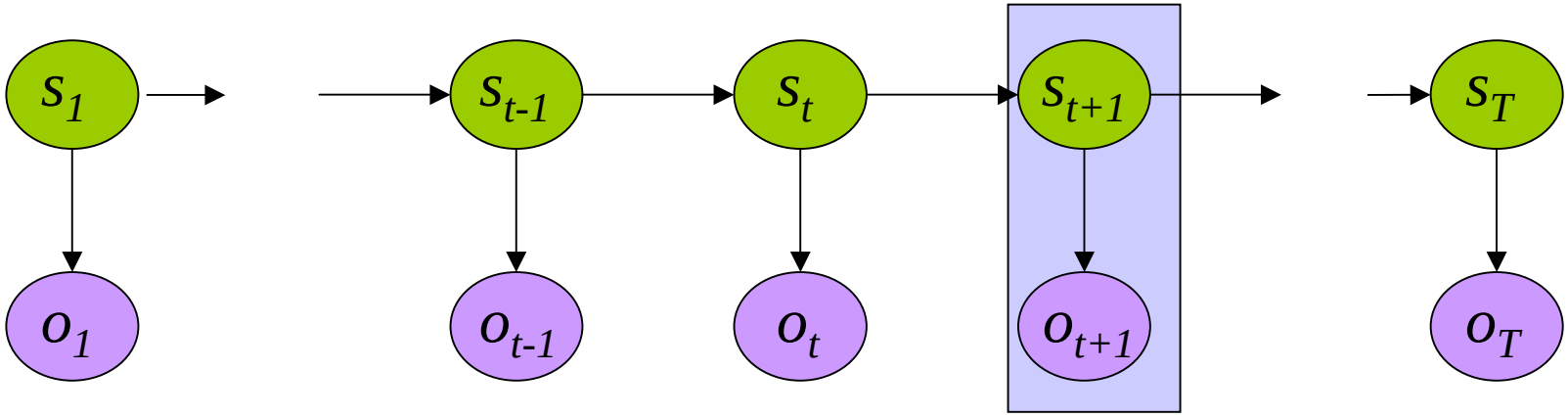
Forward



$$\alpha_t(i) = P(o_1 \ldots o_t, s_t = y_i \mid \mu)$$

$$\alpha_{t+1}(j) = \sum_{i=1 \ldots N} \alpha_t(i) a_{ij} b_{jo_{t+1}}$$

Where,i=1,2,…N, and j=1,2,…N



According to Bayesian Inference

$$P(o_1 o_2 \ldots o_t o_{t+1}, s_{t+1} = j \mid \mu)$$

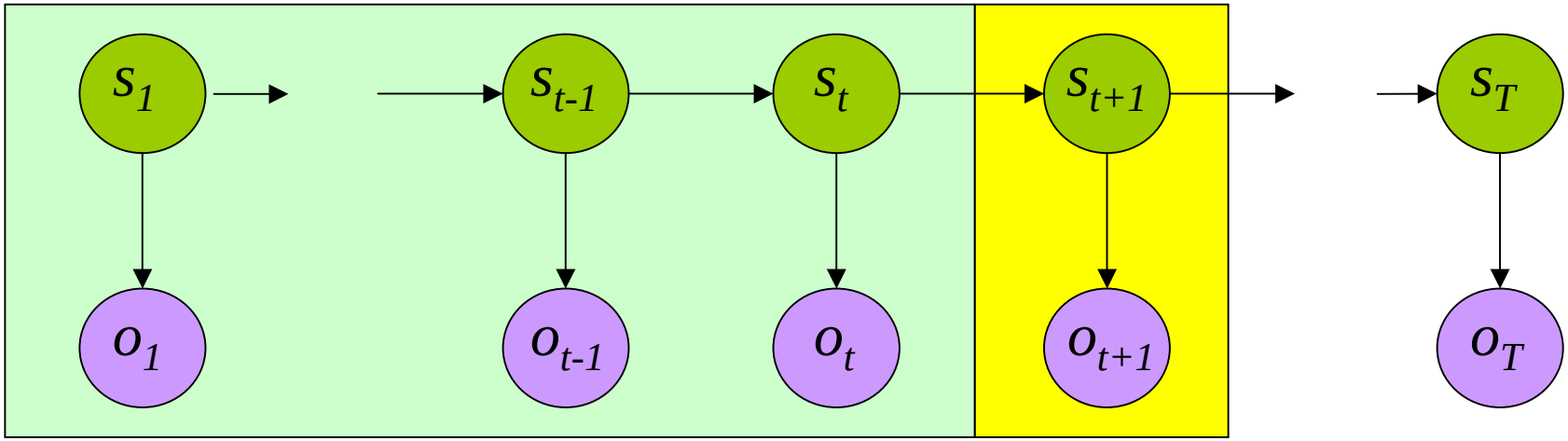
$$= \sum_{\{s_1 \ldots s_t\}} \pi_{s_1} b_{s_1 o_1} \prod_{k=1}^{t} a_{s_k s_{k+1}} b_{s_{k+1} o_{k+1}}$$

$$= \sum_{s_t} a_{s_t s_{t+1}} b_{s_{t+1} o_{t+1}} \sum_{\{s_1 \ldots s_{t-1}\}} \pi_{s_1} b_{s_1 o_1} \prod_{k=1}^{t-1} a_{s_k s_{k+1}} b_{s_{k+1} o_{k+1}}$$

$$= \sum_{i} a_{ij} b_{j o_{t+1}} \sum_{\{s_1 \ldots s_{t-1}\}} \pi_{s_1} b_{s_1 o_1} \prod_{k=1}^{t-1} a_{s_k s_{k+1}} b_{s_{k+1} o_{k+1}} \rightarrow a_t(i)$$

$$= [\sum_{i} a_{ij} \alpha_t(i)] b_{j o_{t+1}} \quad , \quad \text{where, } s_t = i, \quad s_{t+1} = j$$

Forward Probability

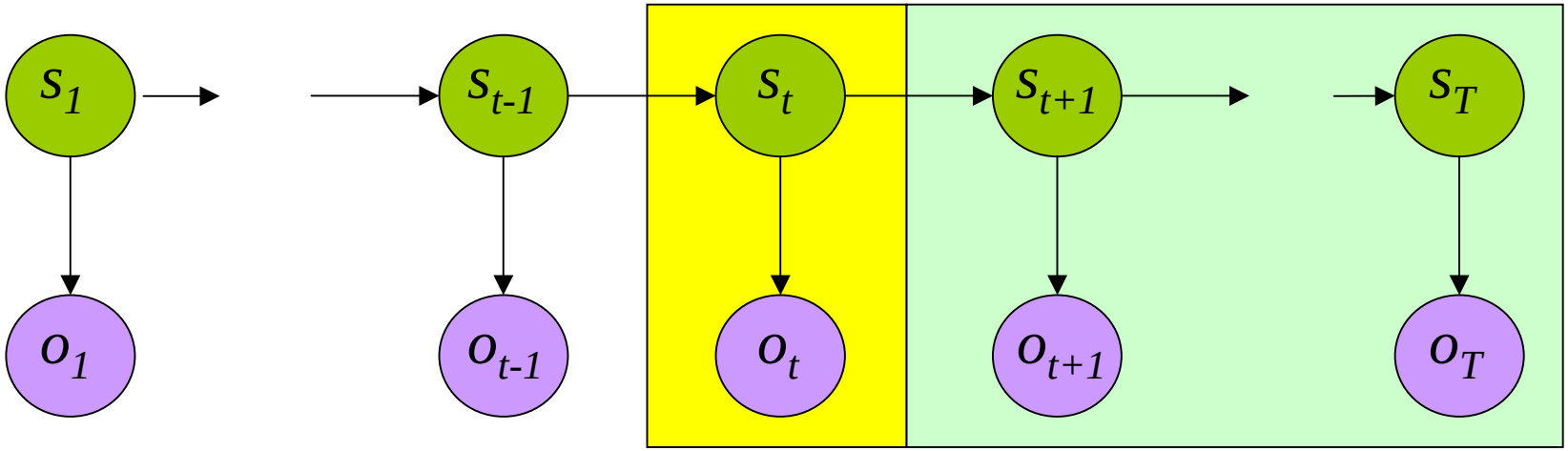


$$\alpha_t(i) = P(o_1 \ldots o_t, s_t = y_i \mid \mu)$$

$$\alpha_{t+1}(j) = \sum_{i=1 \ldots N} \alpha_t(i) a_{ij} b_{jo_{t+1}}$$

$$\alpha_1(i) = \pi_i b_{io_1}$$

Backward Probability

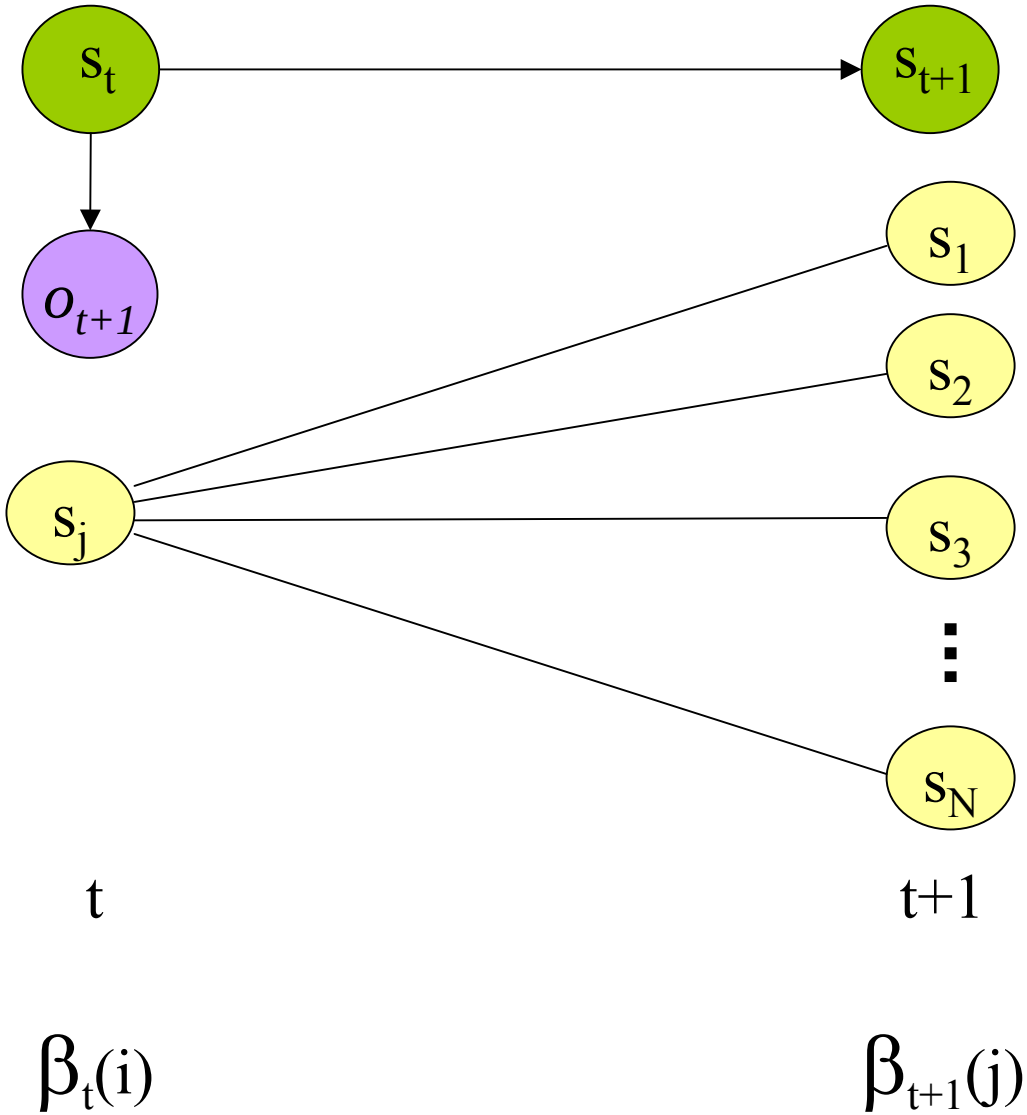


$$\beta_T(i) = 1$$

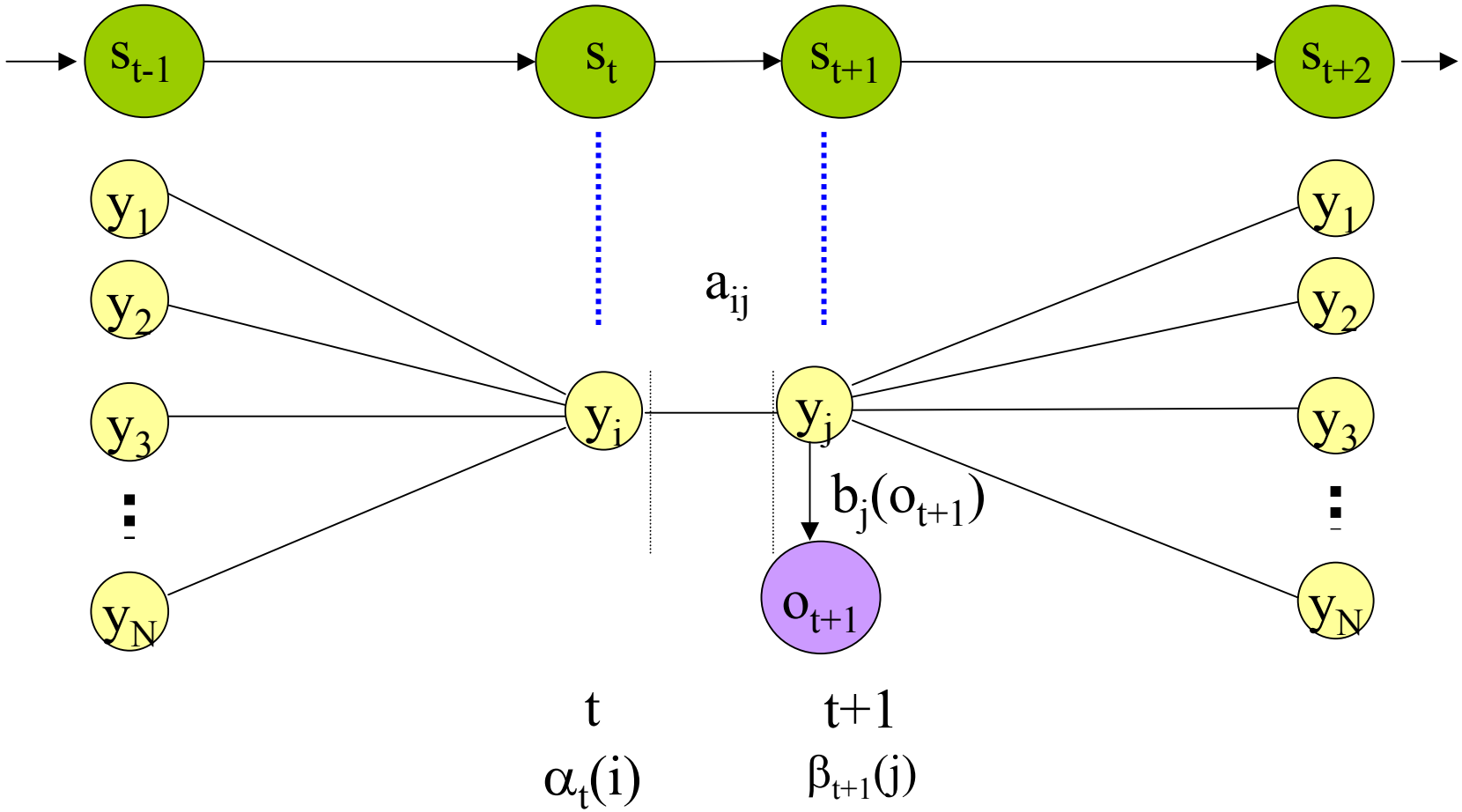
$$\beta_t(i) = P(o_t \ldots o_T \mid s_t = i)$$

$$\beta_t(i) = \sum_{j=1\ldots N} a_{ij} b_{io_t} \beta_{t+1}(j)$$

Backward Probability



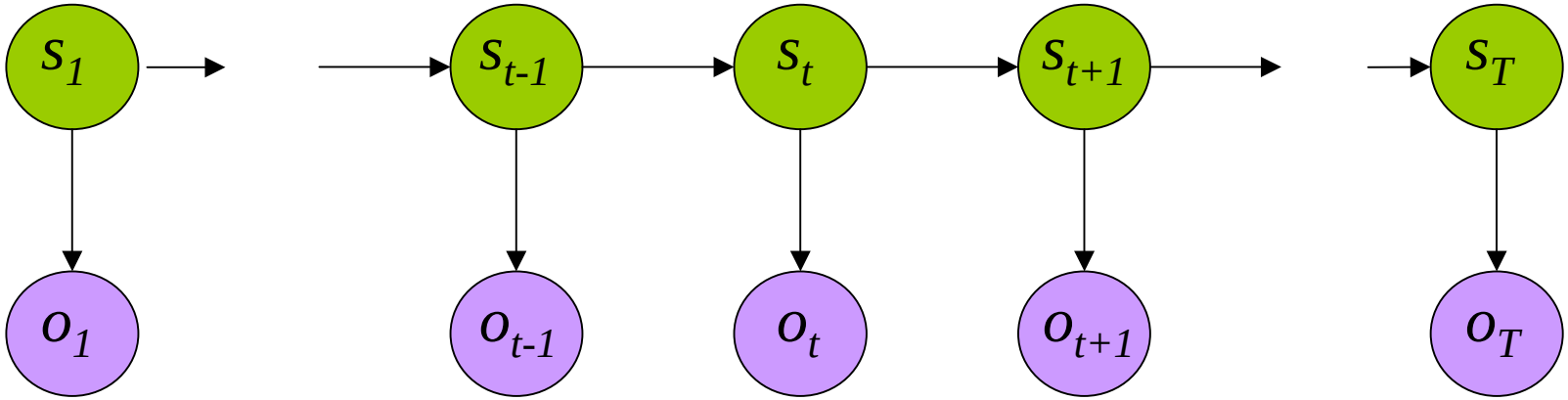
Forward-Backward



$$\alpha_t(i) = P(o_1 \ldots o_t, s_t = y_i \mid \mu)$$

$$\beta_{t+1}(j) = P(o_{t+1} \ldots o_T \mid s_{t+1} = j, \mu)$$

Prediction Solution



$$P(O \mid \mu) = \sum_{i=1}^{N} \alpha_T(i)$$ Forward Procedure

$$P(O \mid \mu) = \sum_{i=1}^{N} \pi_i \beta_1(i)$$ Backward Procedure

$$P(O \mid \mu) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i)$$ Combination

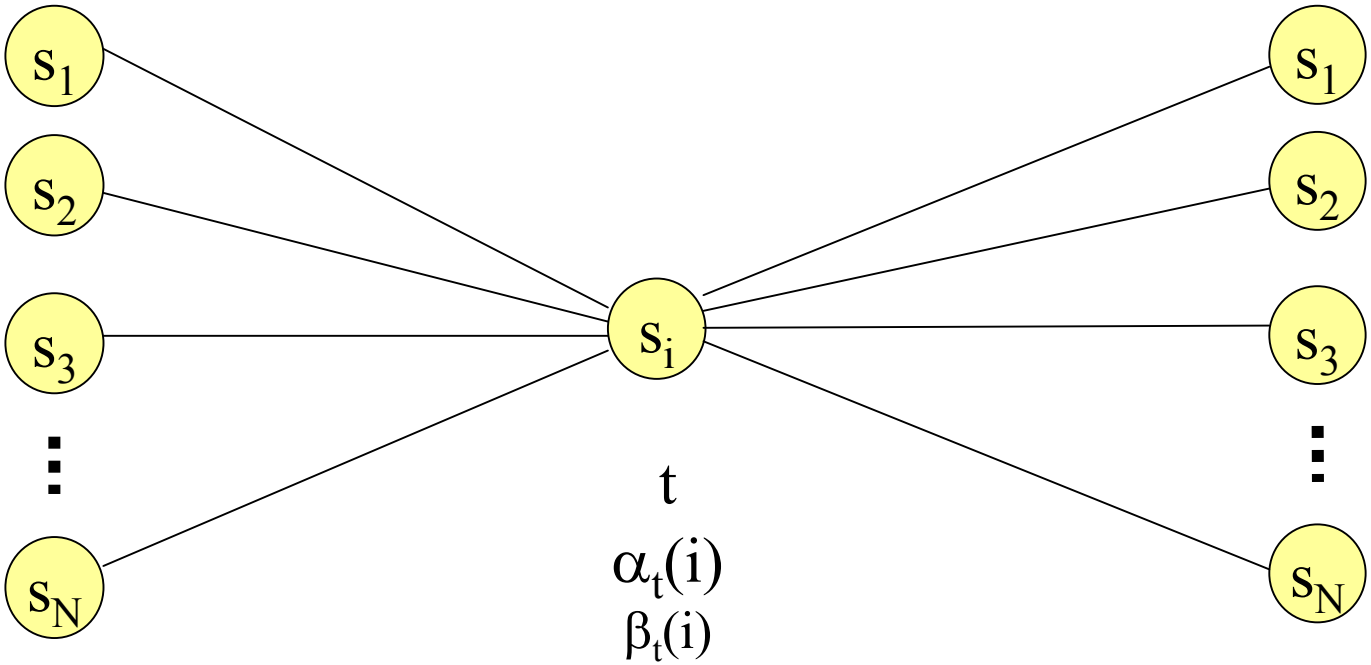
Question 2:State Sequence—POS

- General: Given the observation sequence O and a model μ, how do we choose a state sequence (s_1, …, s_T) that best explains the observations?
- POS: Given Sentence S=$w_1w_2…w_N$, How to find the best tag sequence T=$t_1t_2…t_N$?

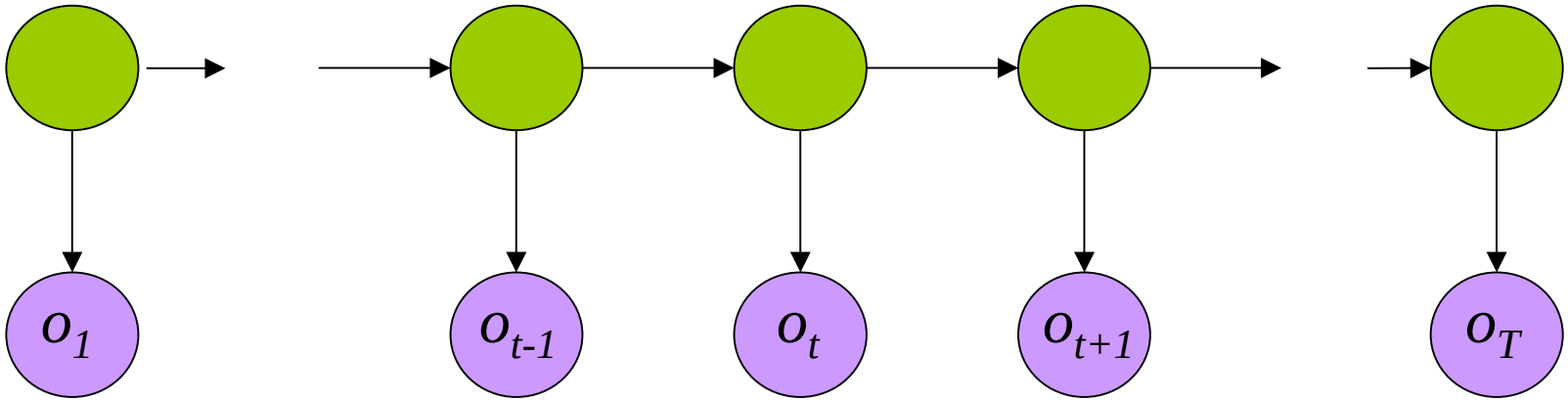
Method-1: Optimizing individually

- For each t, $1 \le t \le T$, we would like to find s_t that maximizes $P(s_t|O, \mu)$.
- Let $\gamma_t(i) = P(s_t = i \,|O, \mu) = P(s_t = i, O|\mu)/P(O|\mu) = (\alpha_t(i)\beta_t(i))/\Sigma_{j=1,N}\, \alpha_t(j)\beta_t(j)$
- The individually most likely state is

  $s_t = Argmax_{\,1 \le i \le N}\, \gamma_t(i),\ 1 \le t \le T$
- This quantity maximizes the expected number of states that will be guessed correctly. However, it may yield a quite unlikely state sequence


s_1
s_2
s_3
s_N
s_i
t
$\alpha_t(i)$
$\beta_t(i)$
s_1
s_2
s_3
s_N

Method-2: Best **State Sequence**: Viterbi



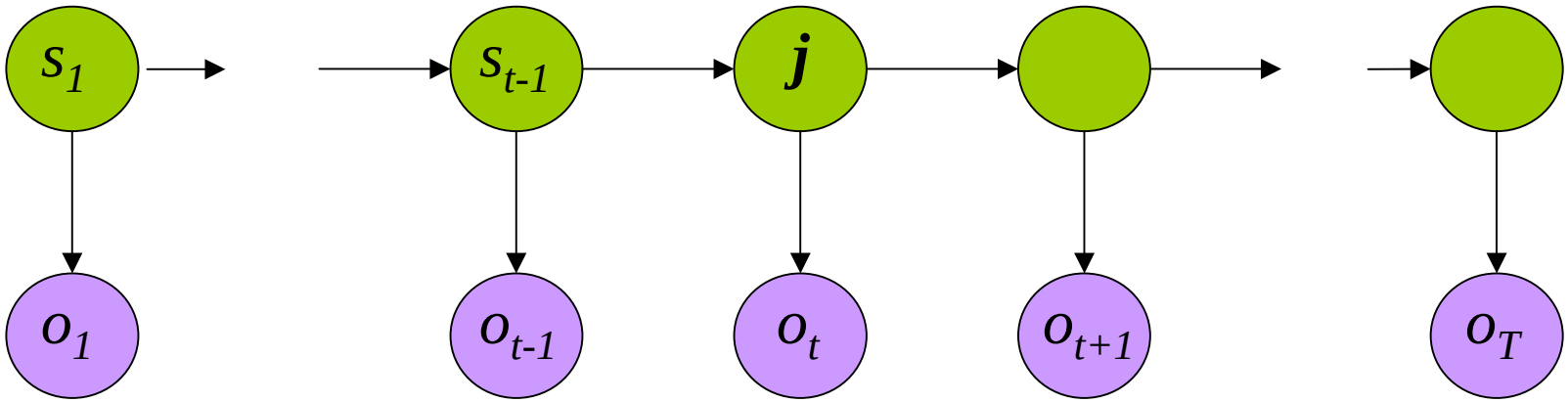
- Find the **state sequence** that best explains the observation sequence

$$\underset{S}{\text{Argmax}}[P(S \mid O,\mu)] = \underset{S}{\text{Argmax}}[\frac{P(O \mid S,\mu)P(S \mid \mu)}{P(O \mid \mu)}] = \underset{S}{\text{Argmax}}[P(O,S \mid \mu)]$$

- Viterbi algorithm

$$\arg\max_S P(S \mid O) = \arg\max_S P(S,O)$$

Viterbi Algorithm



$$\delta_t(j) = \max_{s_1 \ldots s_{t-1}} P(s_1 \ldots s_{t-1}, o_1 \ldots o_{t-1}, s_t = j, o_t)$$

$$\delta_T(j) = \max_{s_1 \ldots s_{T-1}} P(s_1 \ldots s_{T-1}, o_1 \ldots o_{T-1}, s_T = j, o_T)$$

The state sequence which maximizes the probability of seeing the observations to time t-1, landing in state j.

Viterbi Algorithm

$$P(s_1 \ldots s_t, o_1 \ldots o_t, s_{t+1} = j)$$

$$= P(s_1 \ldots s_t, o_1 \ldots o_t) P(s_{t+1} = j \mid s_1 \ldots s_t, s_t, o_1 \ldots o_t)$$

$$= P(s_1 \ldots s_t, o_1 \ldots o_t) P(s_{t+1} = j \mid s_t) \qquad S_t = Parent(S_{t+1})$$

$$= P(s_1 \ldots s_t, o_1 \ldots o_{t-1}) P(o_t \mid s_1 \ldots s_t, s_t) P(s_{t+1} = j \mid s_t)$$

$$= P(s_1 \ldots s_t, s_{t+1}, o_1 \ldots o_{t-1}) P(o_t \mid s_t) P(s_{t+1} = j \mid s_t)$$

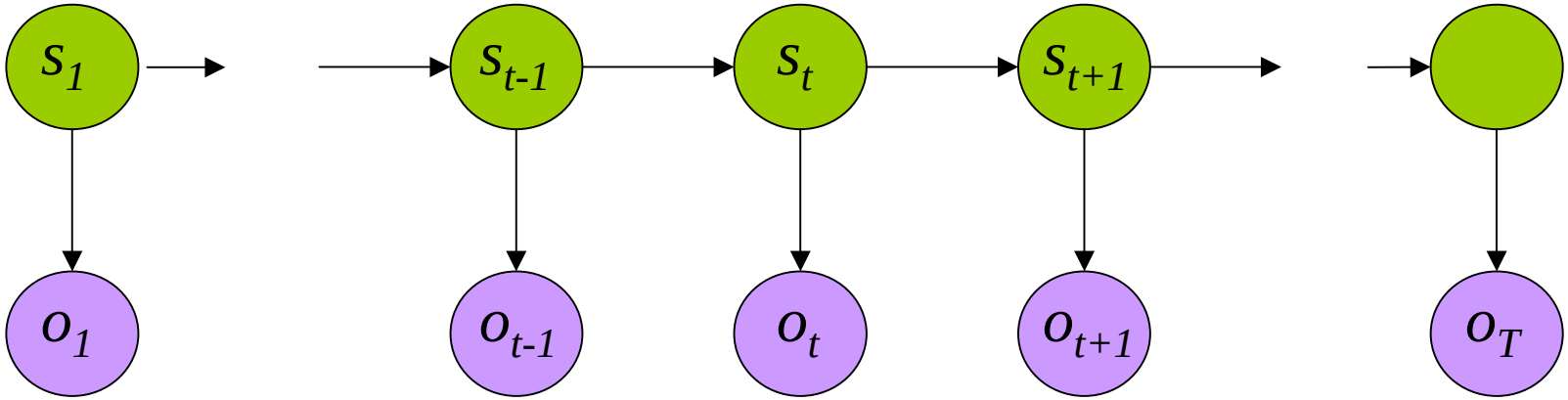
$$= P(s_1 \ldots s_t, o_1 \ldots o_{t-1}) a_{s_t j} b_{s_t o_t}$$

So,

$$\delta_{t+1}(j) = \max_{s_1 s_2 \ldots s_t} P(s_1 \ldots s_t, o_1 \ldots o_t, s_{t+1})$$

$$= \max_i [(\max_{s_1 s_2 s_{t-1}} \delta_t(i)) a_{ij} b_{io_t}]$$

Viterbi Algorithm



$$\delta_t(j) = \max_{s_1 \ldots s_{t-1}} P(s_1 \ldots s_{t-1}, o_1 \ldots o_{t-1}, s_t = j, o_t)$$

$$\delta_{t+1}(j) = \max_i \delta_t(i) a_{ij} b_{jo_{t+1}}$$

$$\psi_{t+1}(j) = \arg\max_i \delta_t(i) a_{ij} b_{jo_{t+1}}$$

- Iterative(or Recursive) Computation

Question 3: Learning

- Given a certain observation Data, find the values of the model parameters $\mu=(A, B, \pi)$, which best explain what the observation sequence.

Learning:Fully Observed Data

Data : $D=\{D_1, D_2, \ldots D_M\}$

$D_m = \{(o_{m1}, o_{m2}, \ldots o_{mT}, s_{m1}, s_{m2}, \ldots s_{mT})\}$

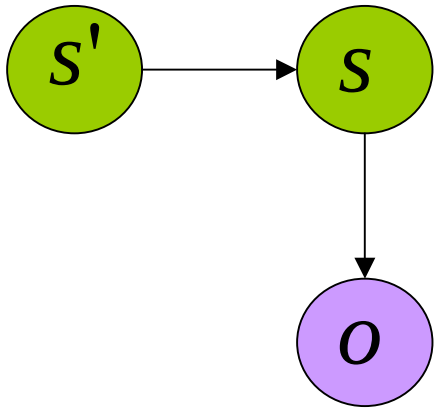
f(s',s): number of times state s following s' in D;

f(s): number of times state s in D;

f(s,o): number of times state s is paired with o in D;

$$\log P(D|\theta) = \log \prod P(D_m|\theta) = \sum_{m=1}^{M} \log P(D_m|\theta)$$

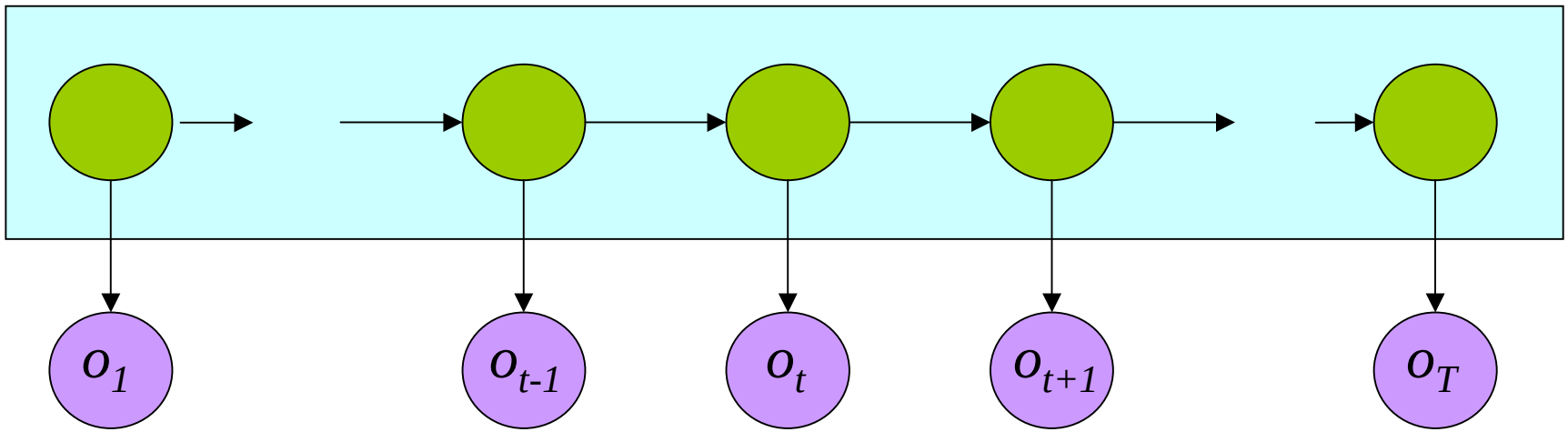
Learning: Maximum Likelihood



$$\hat{P}(s \mid s') = \frac{f(s, s')}{f(s')}$$

$$\hat{P}(o \mid s) = \frac{f(o, s)}{f(s)}$$

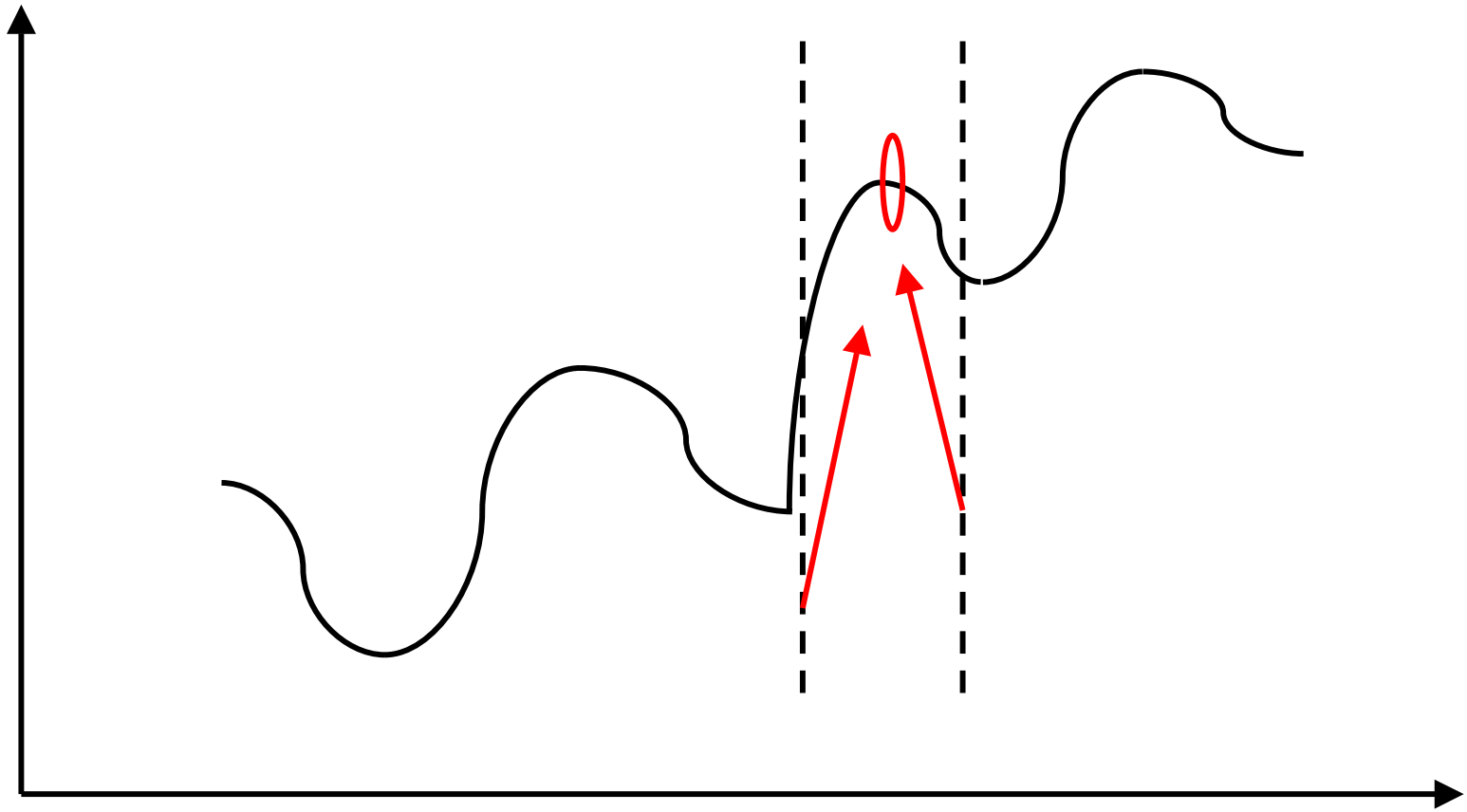
Learning: partially Observed Data



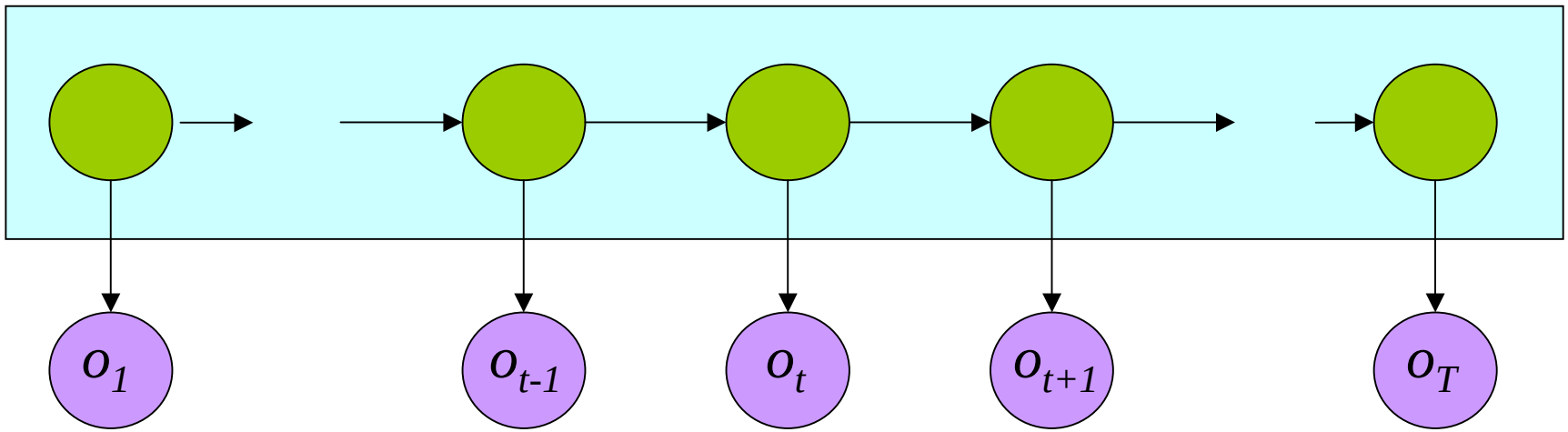
$$D = \{O_1, O_2, ..., O_M\}, \qquad where, O_m = (o_{m1}, o_{m2}, ...o_{mT})$$

$$l(\theta) = \sum_{m=1}^{M} \log \sum_{s} P(O_m, S \mid \theta) \quad \text{where, S is losing data}$$

$$Q(\theta, \theta^{k-1}) = \sum_{m=1}^{M} \sum_{S} P(S \mid O_m, \theta^{k-1}) \log P(O_m, S \mid \theta)$$



Learning: partially Observed Data



$$Q(\theta,\theta^{k-1}) = \sum_{m=1}^{M}\sum_{s} P(S \mid O_m,\theta^{k-1}) \log P(O_m,S \mid \theta)$$

$$p(O_m,S \mid \theta) = \pi_{s_1} b_{s_1 o_{m,1}} \prod_{t=1}^{T-1} a_{s_t s_{t+1}} b_{s_{t+1} o_{m,t+1}} = \pi_{s_1} \prod_{t=1}^{T-1} a_{s_t s_{t+1}} \prod_{t=1}^{T} b_{s_t o_{m,t}}$$

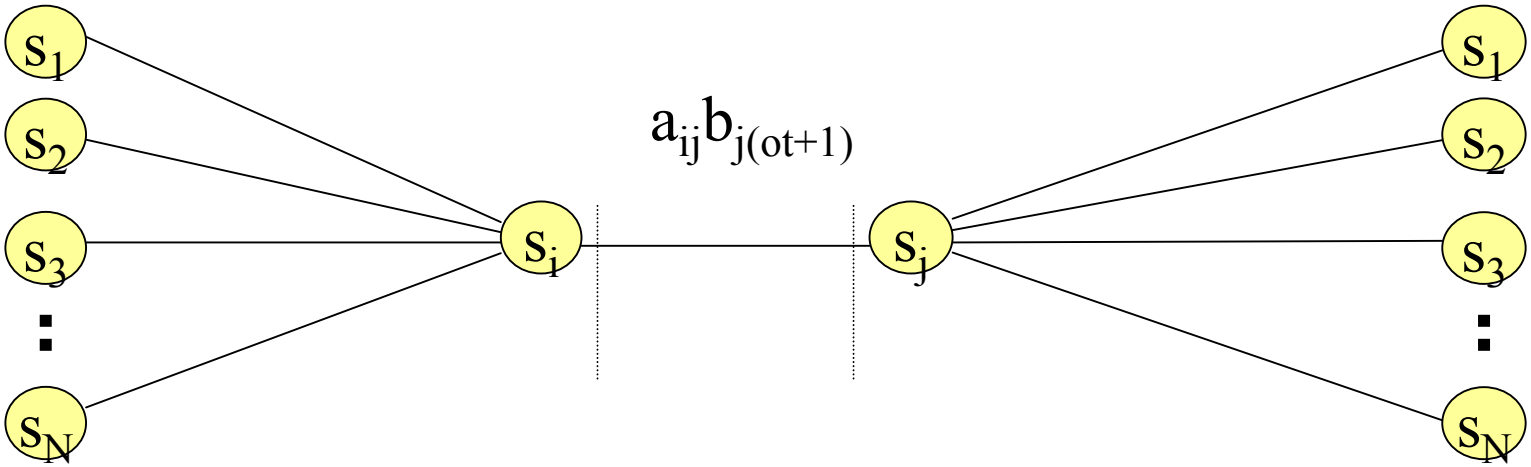
Expection

Learning: partially Observed Data

Maximization

$$\frac{\partial Q(\theta,\theta^{k-1})}{\partial \pi_i}=0,\frac{\partial Q(\theta,\theta^{k-1})}{\partial a_{i,j}},\frac{\partial Q(\theta,\theta^{k-1})}{\partial b_{i,o}}$$

Baum-Welch Algorithm



$$p_t(i,j) = p(s_t = i, s_{t+1} = j \mid O, \mu)$$

$$= \frac{p(s_t = i, s_{t+1} = j, O \mid \mu)}{p(O \mid \mu)}$$

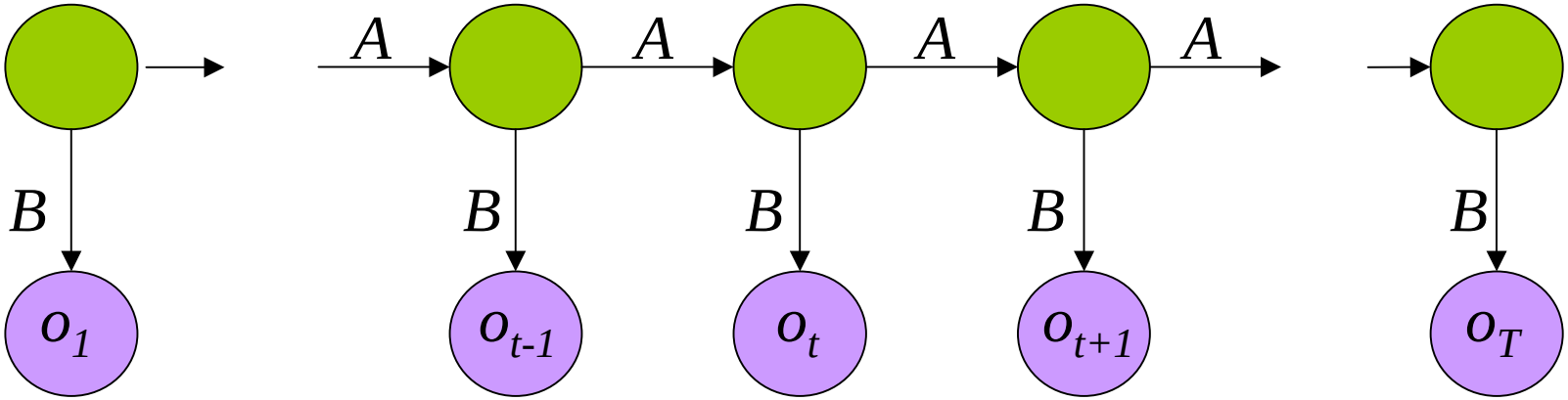
$$= \frac{\alpha_t(i) a_{ij} b_{jo_{t+1}} \beta_{t+1}(j)}{\sum_{l=1\ldots N} \alpha_t(l)\beta_t(l)}$$

$$\alpha_t(i) = P(o_1 \ldots o_t, s_t = i)$$

$$\beta_{t+1}(j) = P(o_{t+1} \ldots o_T \mid s_{t+1} = j)$$

Let $\gamma_t(i) = \sum_{j=1\ldots N} p_t(i,j)$

Baum-Welch Algorithm



$$\hat{\pi}_i = \gamma_1(i)$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T} p_t(i,j)}{\sum_{t=1}^{T} \gamma_t(i)} \qquad \hat{b}_{ik} = \frac{\sum_{\{t:o_t=k\}} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}$$

Definition (cont.)

- The expected number of transitions from state *i* to *j* in O

  $\Sigma_{t=1..T} P_t(i, j)$

- The expected number of transitions from state *i* in O:

  $\Sigma_{t=1..T} \gamma_t(i)$

Reestimation Procedure

- Begin with model μ perhaps selected at random
- Run O through the current model to estimate the expectations of each model parameter.
- Change model to maximize the values of the path that are used a lot while respecting stochastic constraints.
- Repeat this process (hoping to converge on optimal values for the model parameters μ).

Reestimation Formula

From $: \mu = (A, B, \Pi)$, derive $\hat{\mu} = (\hat{A}, \hat{B}, \hat{\Pi})$.

Note $: P(O \mid \hat{\mu}) \geq P(O \mid \mu)$

The expected frequency in state i at time t=1: $\hat{\pi}_i = \gamma_1(i)$

$$\hat{a}_{ij} = \frac{\text{expected\#transitions from state } i \text{ to } j}{\text{expected\#transitions from state } i}$$

$$= \frac{\sum_{t=1}^{T} \mathrm{P}_t(i, j)}{\sum_{t=1}^{T} \gamma_t(i)}$$

Reestimation Formula (cont.)

$$\hat{b}_{ijk} = \frac{\text{expected\#transitions from state } i \text{ to } j \text{ observing } k}{\text{expected\#transitions from state } i \text{ to } j}$$

$$= \frac{\sum_{\text{t: o}_t = k, 1 \le t \le \text{T}} \text{P}_t(i, j)}{\sum_{t=1}^{T} \text{P}_t(i, j)}$$

$$\hat{b}_{ij} = \frac{\sum_{\text{t: o}_t = k} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}$$

Summary on HMM

- Generative model
- Learning: both supervised and unsupervised
- In general, accuracy is lower than discriminative model

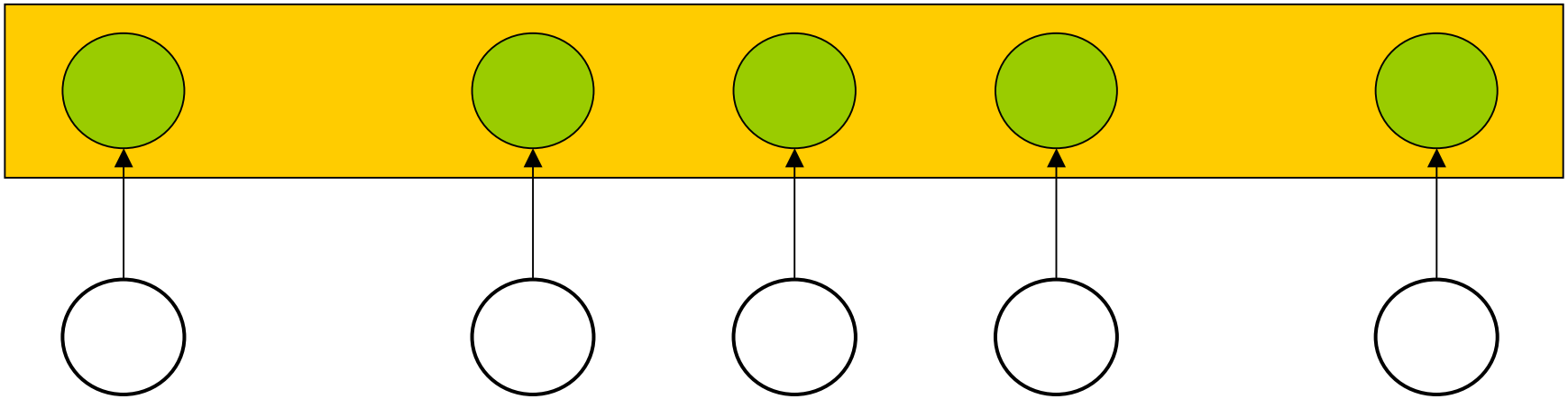
Disadvantage of HMMs (1)

- No Rich Feature Information
  - Rich information are required
- Example: POS Tagging
  - How to evaluate $P(w_k|t_k)$ for unknown words w_k ?
  - Useful features
    - Suffix, e.g., -ed, -tion, -ing, etc.
    - Capitalization

Outline

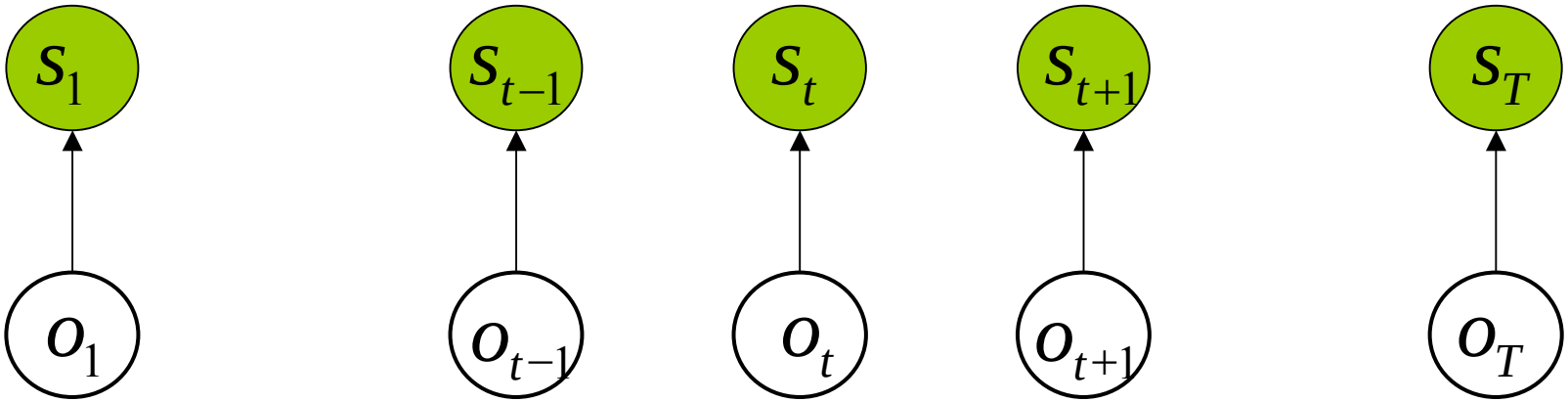
- Sequence Labeling
- HMM
- ➢ **MEMM**
- CRF

MaxEnt on Sequence



- Green nodes are ***states;***
- States are mutually **independent**;
- White nodes are ***observations;***
- Observations **determine states**.

MaxEnt on Sequence



$$P(S \mid O) = \prod_{t=1}^{T} P(s_t \mid O)$$

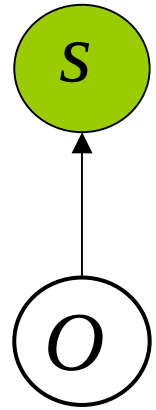
MaxEnt Relation

$$s \leftarrow y, O \leftarrow x$$

$$P(s \mid O) = P(y \mid x)$$

$$P(y \mid x) = \frac{\exp(\sum_k \lambda_k f_k(x, y))}{Z(x)}$$

$$Z(x) = \sum_y \exp(\sum_k \lambda_k f_k(x, y))$$

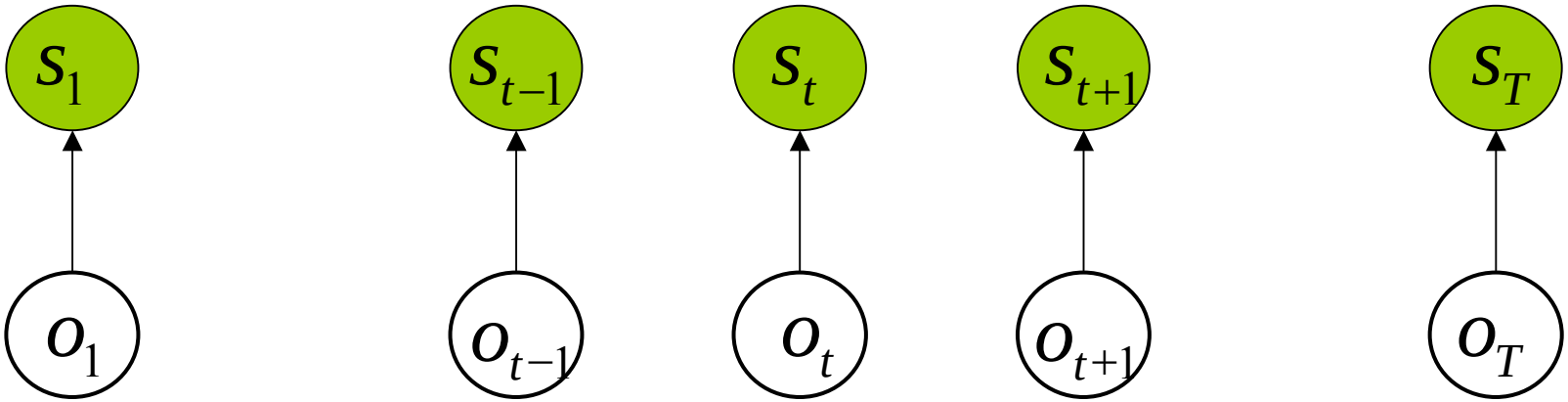


λ : weight

$f(x, y)$: feature

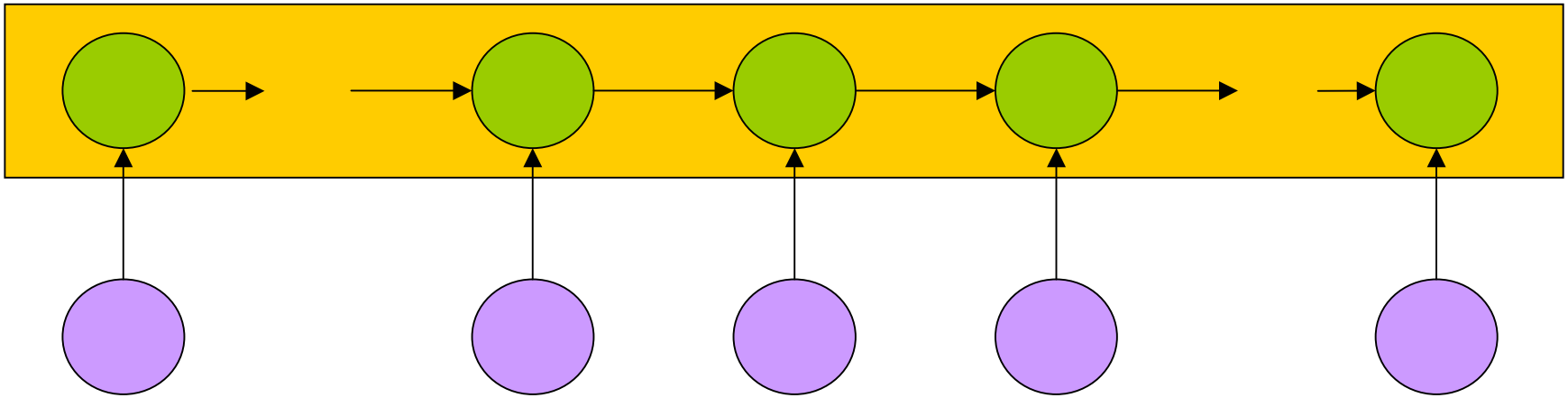
- States are mutually **independent**;
- Observations **determine states**.

Tagging



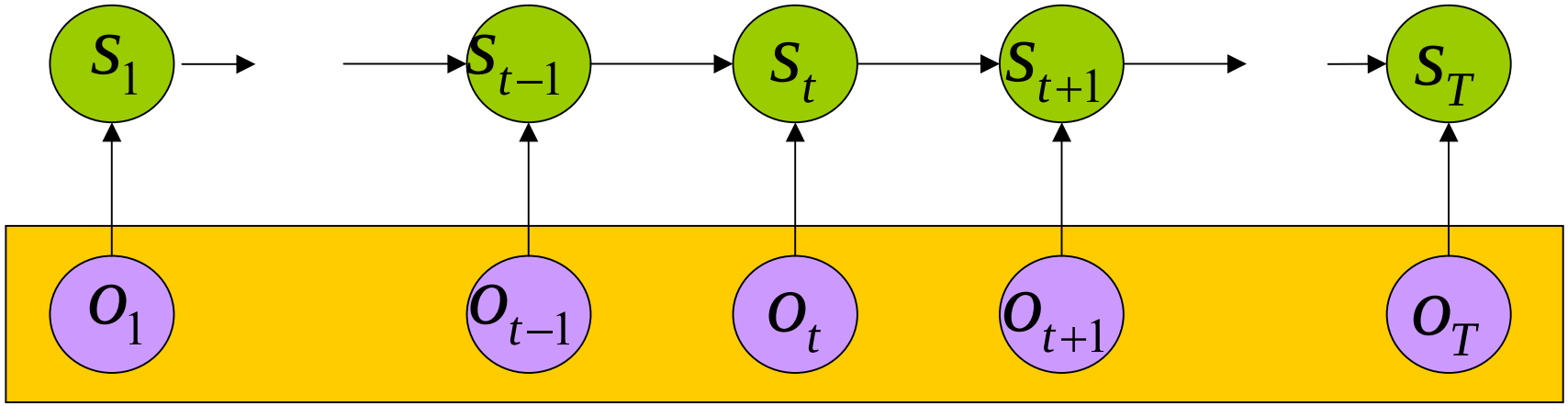
$$\arg\max_S P(S \mid O) = \arg\max_S \prod_{t=1}^{T} P(s_t \mid O)$$

What is MEMM?



- Green nodes are ***states***
- State depends only on previous state
- Purple nodes are ***observations***
- Observations determine states

MEMM Formalism

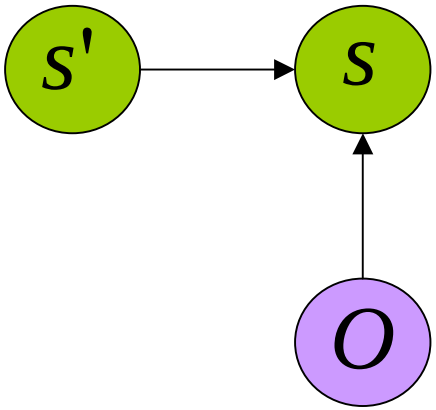


$$P(S \mid O)$$

$$= P(s_1 \mid O) \prod_{t=2}^{T} P(s_t \mid s_{t-1}, O)$$

$$= P(s_1 \mid o_1) \prod_{t=2}^{T} P(s_t \mid s_{t-1}, o_t)$$

MEMM Formalism



$$s \leftarrow y, s' \leftarrow y', O \leftarrow x$$

$$P(s \mid s', O) = P(y \mid y', x)$$

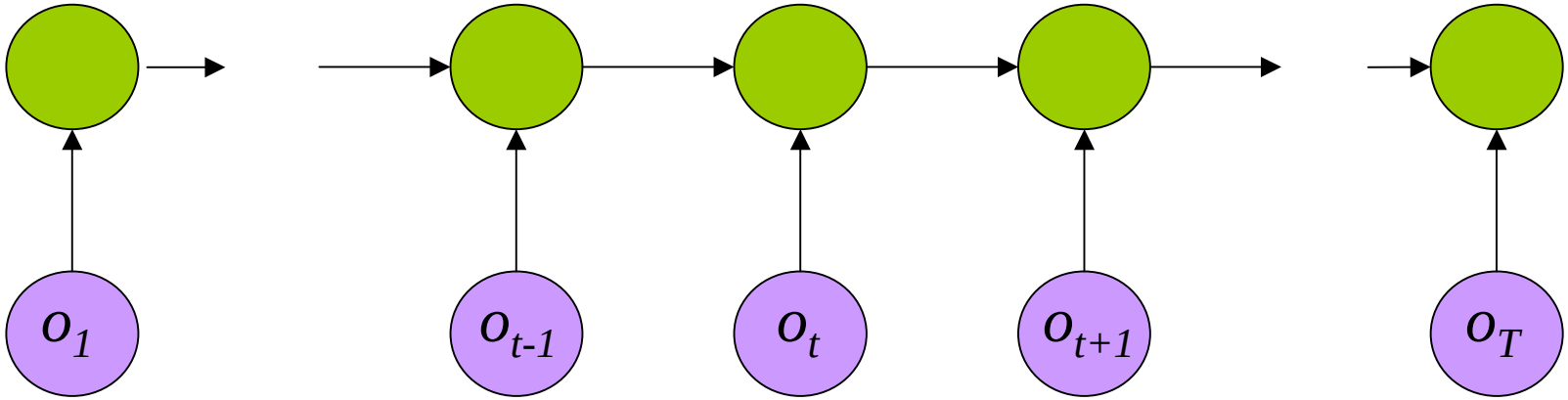
$$P(y \mid y', x) = \frac{\exp(\sum_k \lambda_k f_k(x, y', y))}{Z(y', x)}$$

$$Z(y', x) = \sum_y \exp(\sum_k \lambda_k f_k(x, y', y))$$

Inference in MEMM

- Tagging: given observation sequence, find most likely corresponding state sequence
- Learning: given observation sequence and corresponding state sequence, find model that best explains the matching

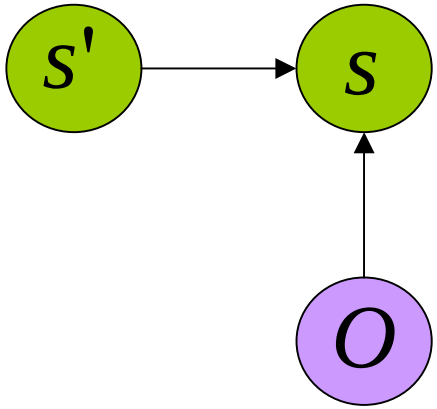
Tagging



- Viterbi algorithm

$$\arg\max_S P(S \mid O) = \arg\max_S P(s_1 \mid O) \prod_{t=2}^{T} P(s_t \mid s_{t-1}, O)$$

Learning



$$(x_1, y_1', y_1), (x_2, y'_2, y_2), \cdots, (x_n, y'_n, y_n)$$

$$P(y \mid y', x) = \frac{\exp(\sum_k \lambda_k f_k(x, y', y))}{Z(x, y')}$$

$$Z(x, y') = \sum_y \exp(\sum_k \lambda_k f_k(x, y', y))$$

GIS/IIS

$$\arg\max \sum_{i=1}^{n} \log P(y_i \mid y'_i, x_i)$$

Summary of MEMM

- Discriminative model
- Accuracy is higher than MaxEnt, lower than CRF
- Problem: local model → label bias problem

Disadvantage of MEMMs (1)

- Complex Algorithm of Maximum Entropy Solution
  - Both IIS and GIS are difficult to implement
  - Require many tricks in implementation
- Slow in Training
  - Time consuming when data set is large
    - Especially for MEMM

Disadvantage of MEMMs (2)

- Maximum Entropy Markov Model
  - Maximum entropy model as a sub model
  - Optimization of entropy on sub models, not on global model
- Label Bias Problem

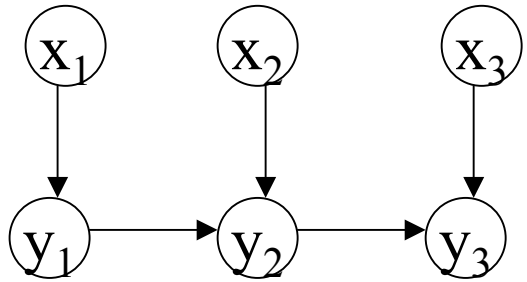
Label Bias Problem

Training Data

X:Y
rib:123
rib:123
rib:123
rob:456
rob:456

Parameters

$$P(1 \mid r) = 0.6,\ P(4 \mid r) = 0.4,$$
$$P(2 \mid i, 1) = P(2 \mid o, 1) = 1,$$
$$P(5 \mid i, 4) = P(5 \mid o, 4) = 1,$$
$$P(3 \mid b, 2) = P(6 \mid b, 5) = 1$$



In the training data, label value 2 is the only label value observed after label value 1
Therefore **P(2 | 1) = 1, so P(2 |x, 1) = 1 for all x**

Label Bias Problem

Training Data

X:Y

rib:123

rib:123

rib:123

rob:456

rob:456

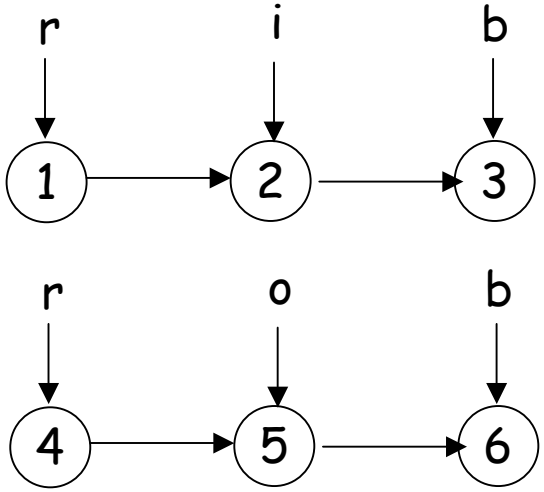
New input: rob

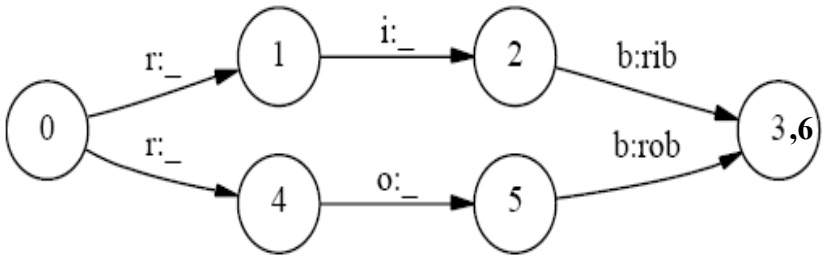
Parameters

$$P(1 \mid r) = 0.6,\ P(4 \mid r) = 0.4,$$
$$P(2 \mid i,1) = P(2 \mid o,1) = 1,$$
$$P(5 \mid i,4) = P(5 \mid o,4) = 1,$$
$$P(3 \mid b,2) = P(6 \mid b,5) = 1$$

$$P(123 \mid \text{rob}) = P(1 \mid r)P(2 \mid o,1)P(3 \mid b,2)$$
$$= 0.6 \times 1 \times 1 = 0.6$$
$$P(456 \mid \text{rob}) = P(4 \mid r)P(5 \mid o,4)P(6 \mid b,5)$$
$$= 0.4 \times 1 \times 1 = 0.4$$

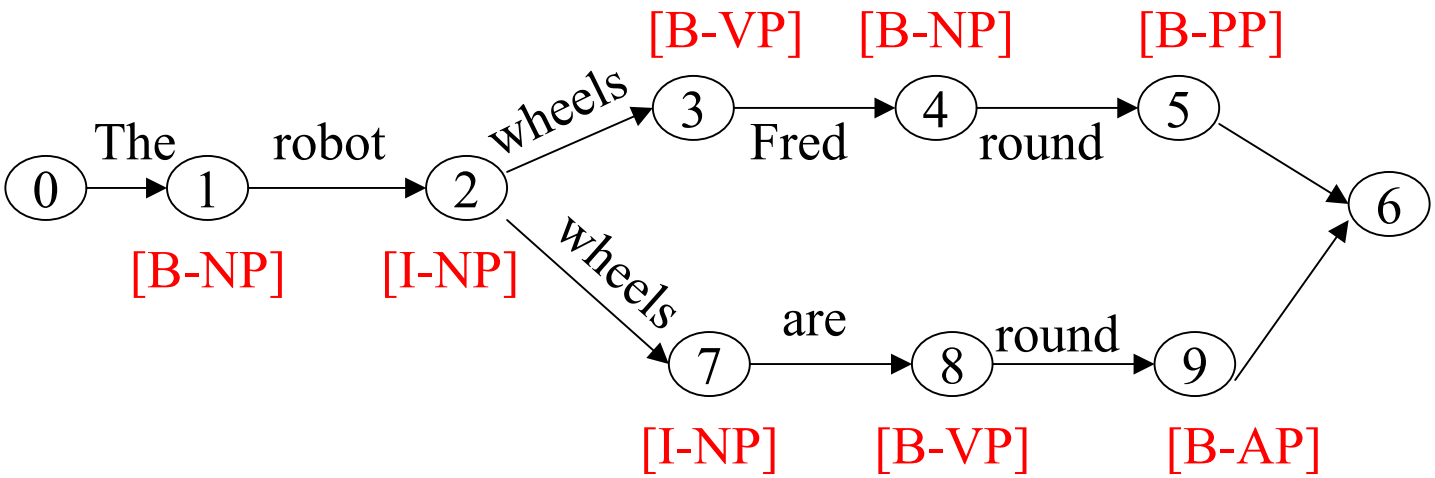
Model





Example

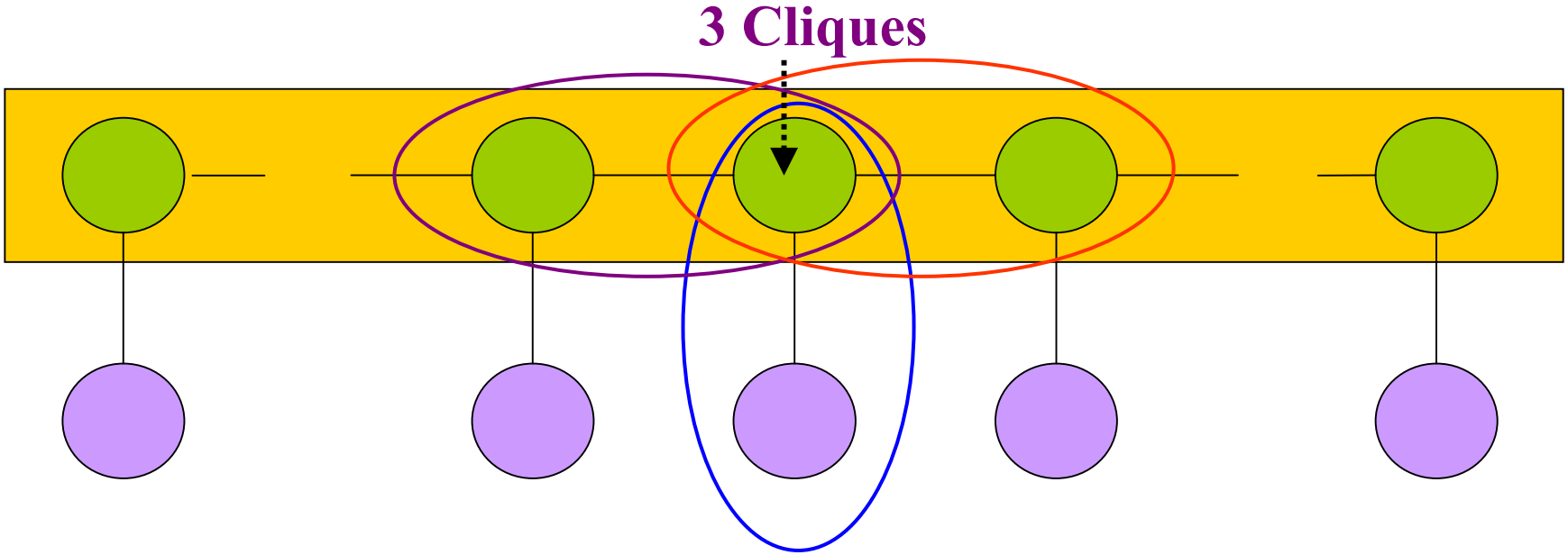
- Shallow parsing two sentences:chunking



Outline

- Sequence Labeling
- HMM
- MEMM
- **CRF**

What is CRF?



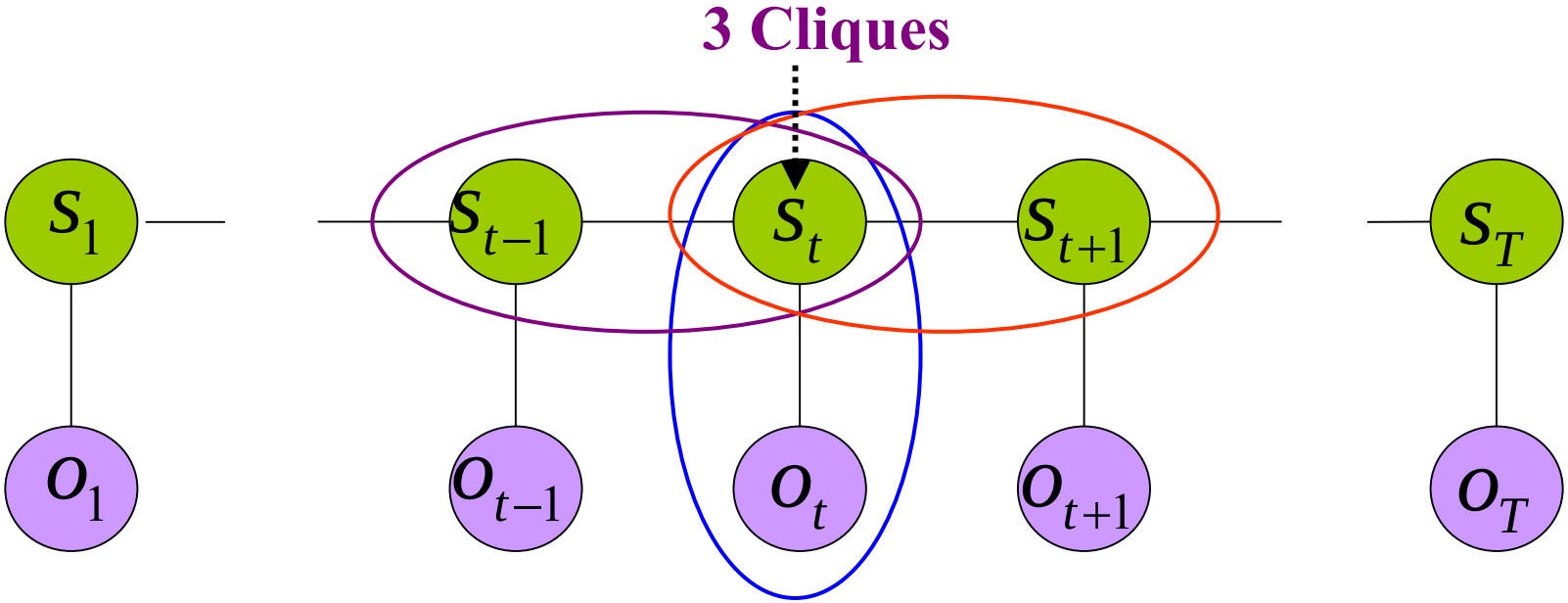
- Green nodes are ***states***
- State depends on neighboring states
- Purple nodes are ***observations***
- Observations determine states

Markov Random Fields and CRFs

- A CRF is a Markov Random Field globally conditioned on O

$$P(S \mid O)$$

CRF Formalism



$P(S \mid O)$

Conditional Markov Random Fields

$$P(S,O)=\frac{1}{Z}\prod_{c\in C(G)}\phi_c(S_c,O_c),$$ C(G): Set of Clique

$$Z=\sum_{s,o}\prod_{c\in C(G)}\phi_c(S_c,O_c)$$

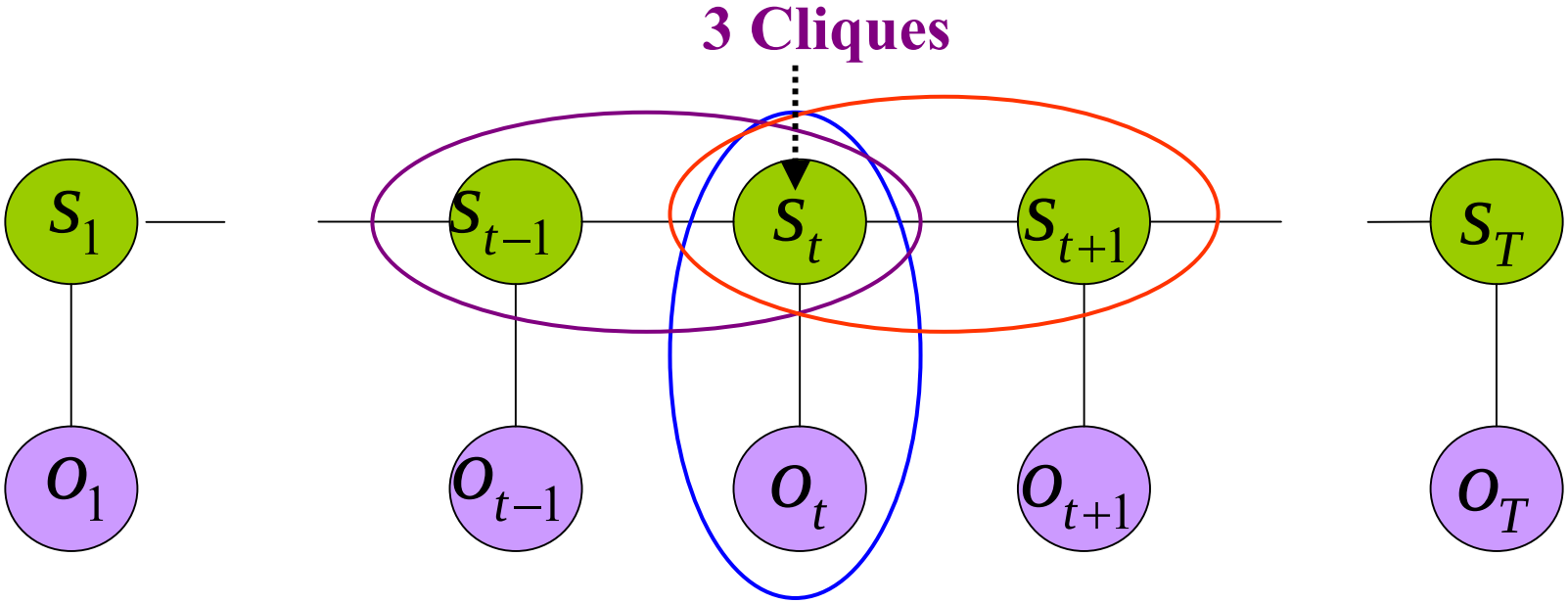
$$P(O)=\sum_S P(S,O)=\sum_S\frac{1}{Z}\prod_{c\in C(G)}\phi_c(S_c,O_c)=\frac{1}{Z}\sum_S\prod_{c\in C(G)}\phi_c(S_c,O_c)$$

$$P(S\mid O)=\frac{P(S,O)}{P(O)}=\frac{\frac{1}{Z}\prod_{c\in C(G)}\phi_c(S_c,O_c)}{\frac{1}{Z}\sum_S\prod_{c\in C(G)}\phi_c(S_c,O_c)}=\frac{1}{\sum_S\prod_{c\in C(G)}\phi_c(S_c,O_c)}\prod_{c\in C(G)}\phi_c(S_c,O_c)$$

So,

$$P(S\mid O)=\frac{1}{Z(O)}\prod_{c\in C(G)}\phi_c(S_c,O_c),\quad \text{where, } Z(O)=\sum_S\prod_{c\in C(G)}\phi_c(S_c,O_c)$$

CRF Formalism



$$C_G = \{\{s_{t-1}, s_t\}, \{s_t, o_t\} : t = 1, 2, \ldots n\}$$

Let, $c_{s_t} = \{s_t, s_{t+1}\}$, $\quad c_{o_t} = \{s_t, o_t\}$

$$C_G = \{c_{s_t}, \ c_{o_t} : t = 1, 2, \ldots n\}$$

$$\phi(c_{s_t}) = \exp(\sum_k \lambda_k f_k(s_{t-1}, s_t))$$

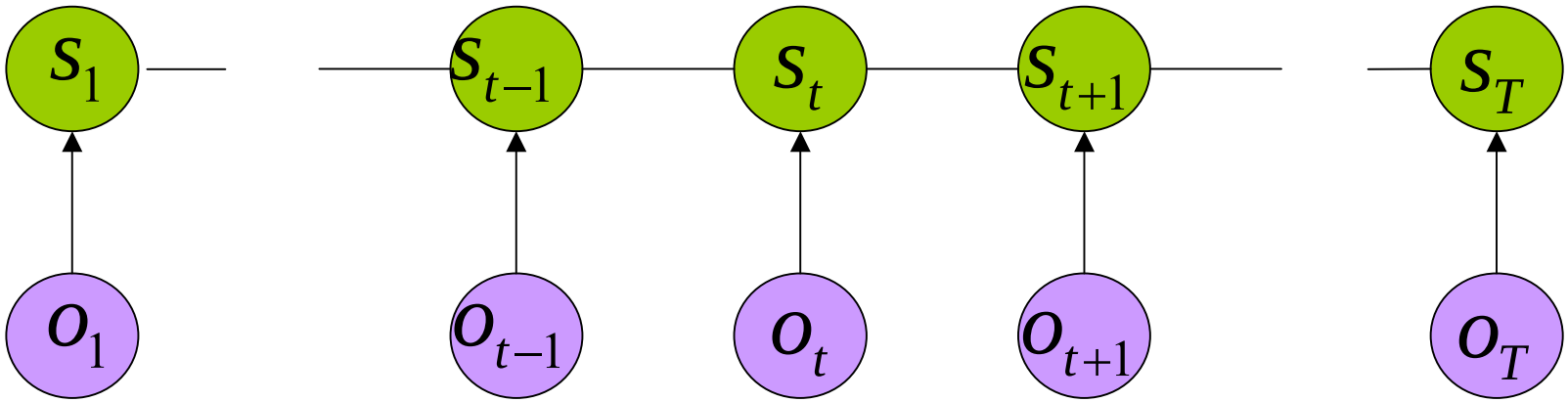
$$\phi(c_{o_t}) = \exp(\sum_k \mu_k g_k(s_t, o_t))$$

$$P(S \mid O) = \frac{1}{Z(O)} \exp(\sum_t \sum_k \lambda_k f_k(s_{t-1}, s_t) + \sum_t \sum_k \mu_k g_k(s_t, o_t))$$

Inference in CRF

- Tagging: given an observation sequence, compute the most likely hidden state sequence
- Learning: given an observation sequence and set of possible models, which model most closely fits the data?

Learning



Maximum Likelihood Estimation

IIS

GIS

…

MLE for CRF

$$L(\theta) = P(D \mid \theta), \text{where, } D = \{D_m = (o_m, s_m) \mid m = 1, \ldots M\}$$
$$= \prod P(D_m \mid \theta)$$
$$l(\theta) = \log L(\theta)$$
$$= \log \prod P(S \mid O, \theta)^{\tilde{P}(S,O)}$$
$$= \sum_{o,s} \tilde{P}(S,O) \log P(S \mid O, \theta)$$

$$P(S \mid O, \theta) = \frac{1}{Z(O)} \exp\left(\sum_t \sum_k \lambda_k f_k(s_{t-1}, s_t, O) + \sum_t \sum_k \mu_k g_k(s_t, O)\right)$$

$$\theta = (\lambda_1, \lambda_2, \ldots, \mu_1, \mu_2, \ldots)$$

$$l(\theta) = \sum_{o,s} \tilde{P}(S,O) \log P(S \mid O, \theta)$$
$$= \sum_{o,s} \tilde{P}(S,O)[\sum_t \boldsymbol{\lambda}\mathbf{f} + \sum_t \boldsymbol{\mu}\mathbf{g}] - \sum_o \tilde{p}(O) \log Z(O)$$

MLE for CRF

$$\frac{\partial\, l(\theta)}{\partial\, \lambda_k}, \frac{\partial\, l(\theta)}{\partial\, \mu_k}$$

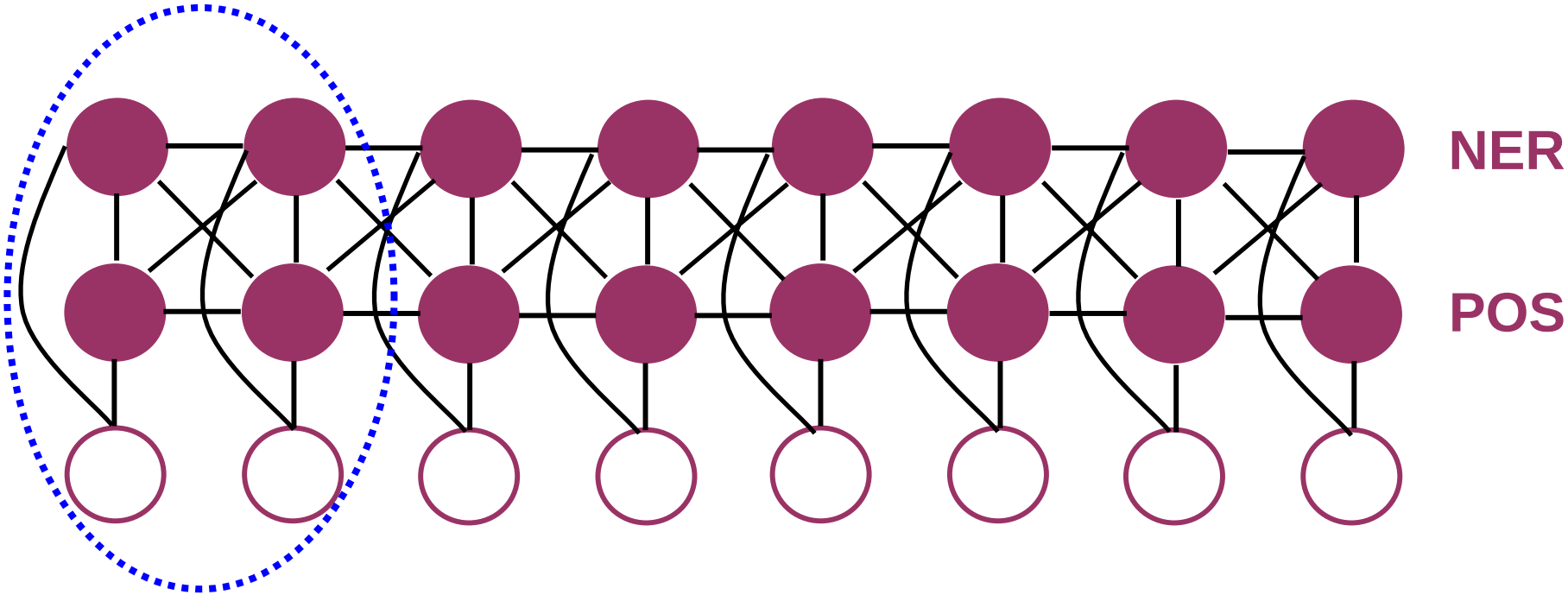
- It is not generally possible to find the value of θ analytically
- Instead, solve this problem iteratively.
- GIS, IIS

Summary of CRF

- Discriminative model
- Global model
- Accuracy is high
- Training speed is lower than MaxEnt and MEMM

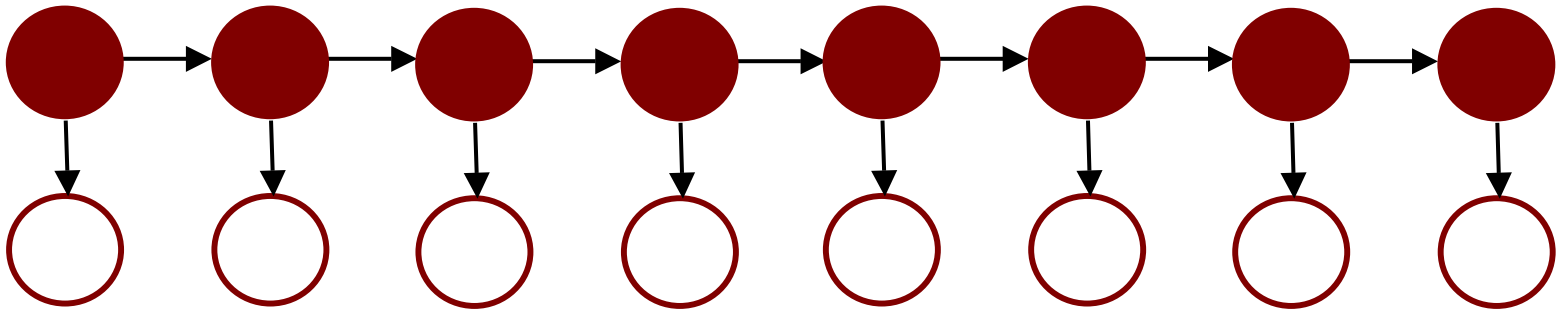
Multi−Level CRFs(Joint Labeling)

- It's a pretty straightforward extension to then label more than one type of sequence at the same time:



Summary

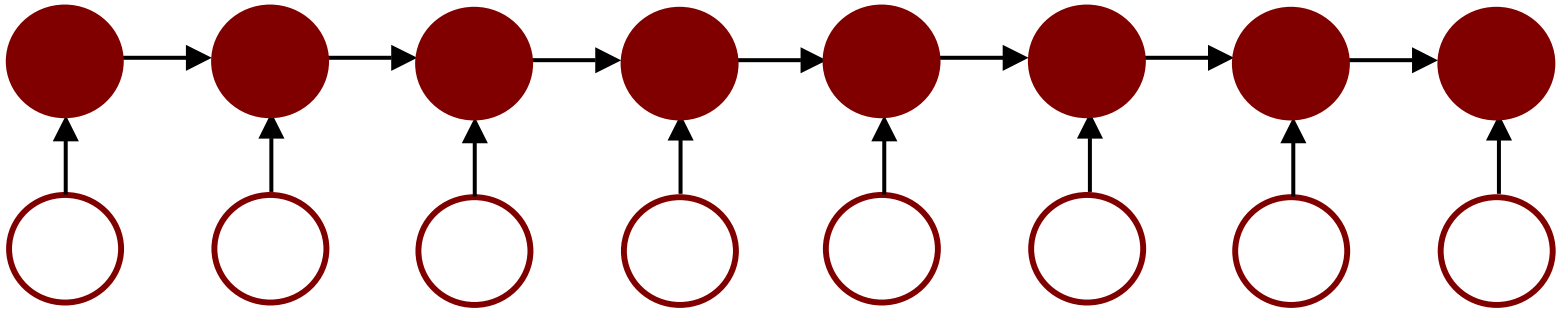
Hidden Markov Models(HMM)



- Generative
  - Find parameters to maximize P(O,X)
- Assumes features are independent
- When labeling X_i future observations are taken into account (forward−backward)

Summary

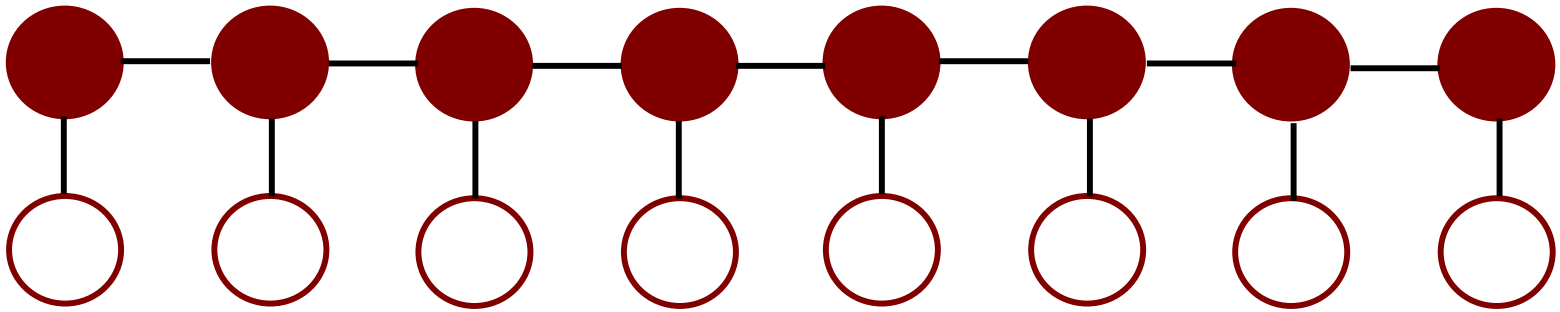
MaxEnt Markov Models (MEMM)



- Discriminative
  - Find parameters to maximize P(S|O)
- No longer assume that features are independent
- Do not take future observations into account (no forward−backward)

Summary

Conditional Random Fields (CRFs)



- Discriminative
- Doesn't assume that features are independent
- When labeling s_i future observations are taken into account.

References

- L.R. Rabiner, B.H. Juang, An introduction to Hidden Markov Model. IEEE ASSP Magazine, 1986.
- A. L. Berger, S. A. D. Pietra, V. J. D. Pietra. A maximum entropy approach to natural language processing. Computational Linguistics, 1996
- A. McCallum and F. Pereira, Maximum entropy Markov models for information extraction and segmentation. ICML'00
- J. Lafferty, A. McCallum, F. Pereira. Conditional random fields: probabilistic models for segmentation and labeling sequence data. ICML'01
- F. Sha and F. Pereira. Shallow parsing with conditional random fields. NAACL'03