Named Entity Recognition

Using Conditional Random Field

Named Entity Recognition

⟨Google⟩$_{1}$, headquartered in ⟨Mountain View⟩$_{2}$ ⟨Mountain View (1600 Amphitheatre Pkwy⟩$_{12}$ ⟨1600⟩$_{14}$ ⟨Amphitheatre Pkwy⟩$_{7}$, ⟨Mountain View⟩$_{2}$, ⟨CA 940430⟩$_{8}$ ⟨940430⟩$_{15}$), unveiled the new ⟨Android⟩$_{3}$ ⟨phone⟩$_{5}$ for ⟨$799⟩$_{13}$ ⟨799⟩$_{16}$ at the ⟨Consumer Electronic Show⟩$_{11}$. ⟨Sundar Pichai⟩$_{4}$ said in his ⟨keynote⟩$_{9}$ that ⟨users⟩$_{6}$ love their new ⟨Android⟩$_{3}$ ⟨phones⟩$_{10}$.

ORGANIZATION

1. **Google**

[Wikipedia Article]

Salience: 0.19

LOCATION

2. **Mountain View**

[Wikipedia Article]

Salience: 0.18

CONSUMER GOOD

3. **Android**

[Wikipedia Article]

Salience: 0.14

PERSON

4. **Sundar Pichai**

[Wikipedia Article]

Salience: 0.11

CONSUMER GOOD

5. **phone**

Salience: 0.10

PERSON

6. **users**

Salience: 0.09

Use a classifier?

- Only based on the features of the current state

- Not include sequential information
  (Classes of the entities may require context information)

A Sequence Labeling Problem

- The task of assigning label sequences to a set of observation sequences

- Given: an input sequence
  $X = x_1, x_2, ..., x_T$

- Output: a sequence with the same length
  $Y = y_1, y_2, ..., y_T$

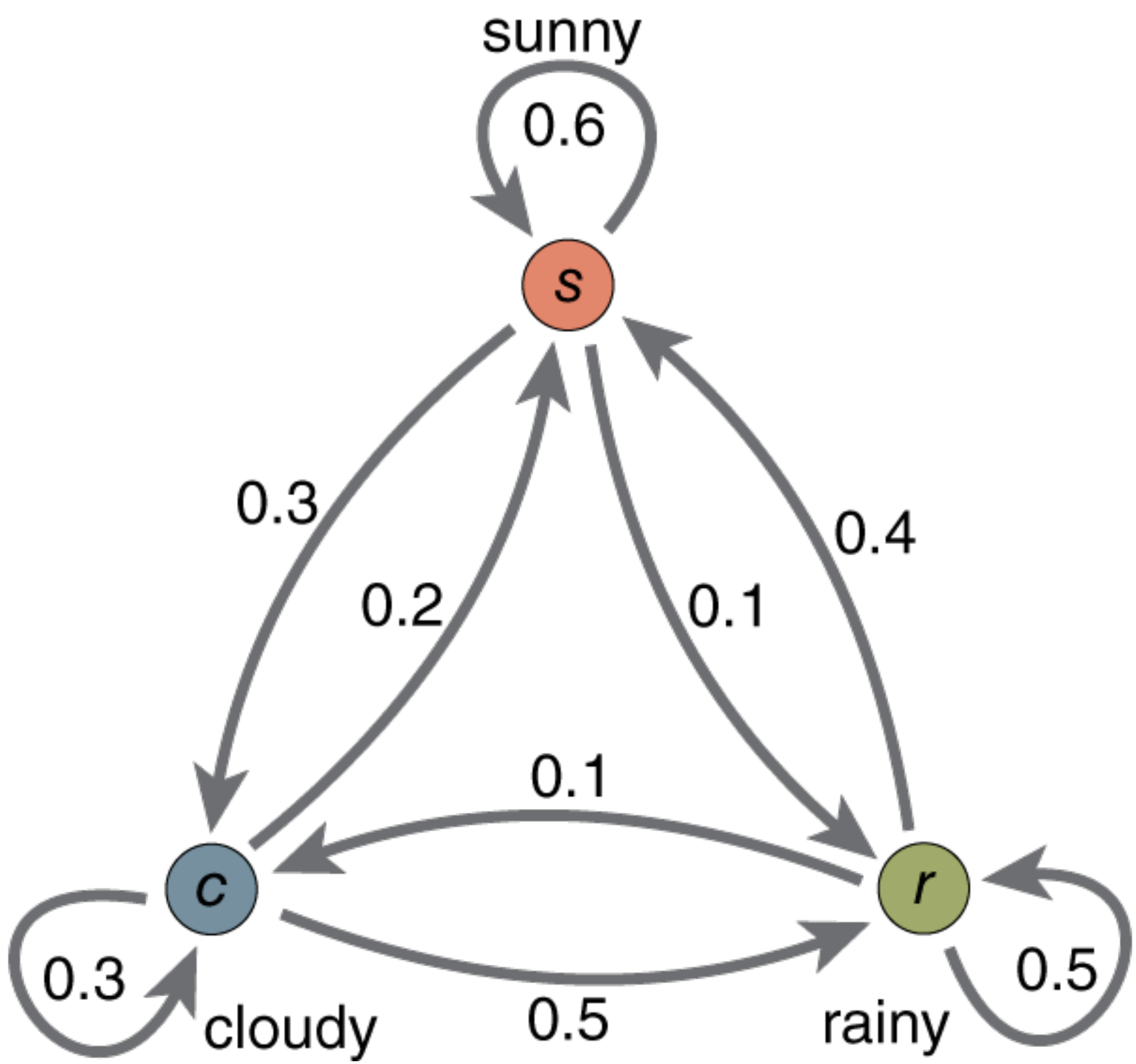
- Also used for solving part-of-speech tagging problem
  X = I, have, a, dream, .
  Y = pron, verb, determiner, noun, punct

Markov Process

- For a Markov Process, the next state only depends on the current state

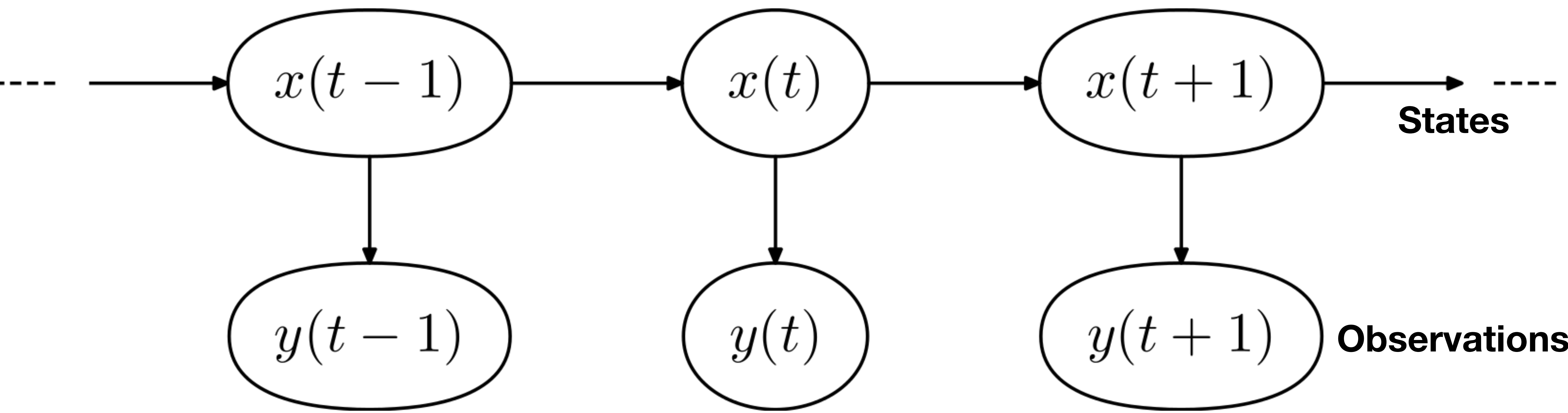
$$p(x_{t+1} \mid x_0, \ldots, x_t) = p(x_{t+1} \mid x_t)$$

Markov Chain



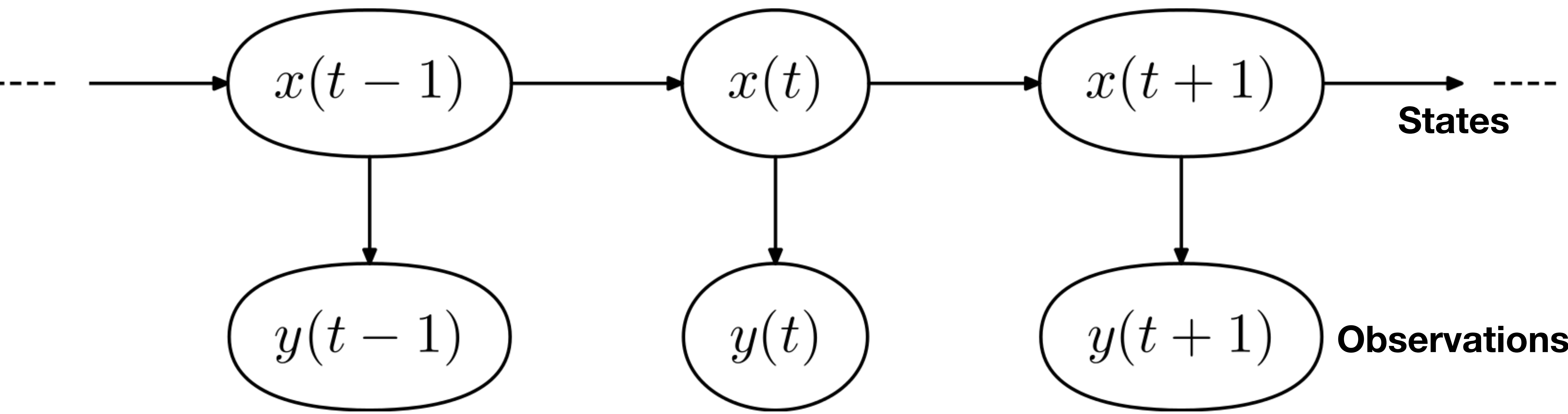
Hidden Markov Model

- Each (hidden) state is dependent on a fixed number of previous states
- Each (visible) observation only depends on the corresponding state



Hidden Markov Model

- What is the most likely states $x_0, x_1, \dots, x_T$ producing the observations $y_0, y_1, \dots, y_T$?



The Viterbi Algorithm

- Suppose we are given
  - Hidden Markov Model (HMM) with states $X = x_1, x_2, ..., x_T$
  - Initial prob. π_i, transition prob. $a_{i,j}$, and emission prob. $b_{i,k}$
  - Observed sequence $Y = y_1, y_2, ..., y_T$
- The Viterbi path, or the most likely state sequence $x_0, x_1, ..., x_T$ is:

$$V_{0,k} = \mathrm{P}(y_0 \mid k) \;.\; \pi_i$$
$$V_{t,k} = \mathrm{P}(y_t \mid k) \;.\; \max_{x \text{ in } X}(a_{x,k} \;. V_{t-1,x})$$
$$\mathrm{Ptr}(t, k) = \mathrm{P}(y_t \mid k) \;.\; \mathrm{argmax}_{x \text{ in } X}(a_{x,k} \;. V_{t-1,x})$$

- The size of V is T x $|X|$ and each $V_{t,k}$ take $|X|$ steps to compute
- The complexity of Viterbi algorithm is $O(T \text{ x } |X|^2)$

Define the parameters

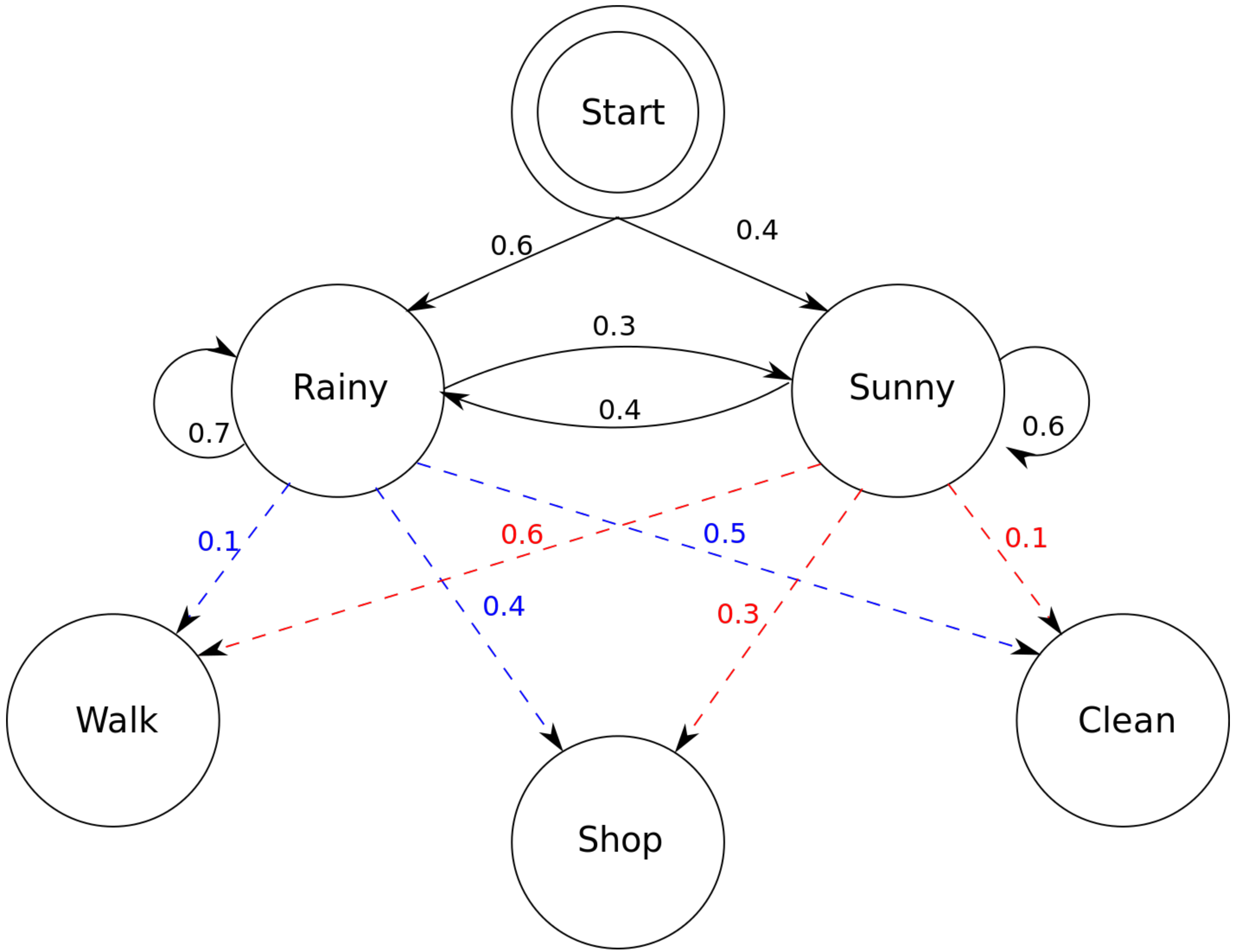
```
states = ('Rainy', 'Sunny')

observations = ('walk', 'shop', 'clean')

start_probability = {'Rainy': 0.6, 'Sunny': 0.4}

transition_probability = {
   'Rainy' : {'Rainy': 0.7, 'Sunny': 0.3},
   'Sunny' : {'Rainy': 0.4, 'Sunny': 0.6},
   }

emission_probability = {
   'Rainy' : {'walk': 0.1, 'shop': 0.4, 'clean': 0.5},
   'Sunny' : {'walk': 0.6, 'shop': 0.3, 'clean': 0.1},
   }
```



Start
0.6
0.4
Rainy
0.3
0.4
Sunny
0.7
0.6
0.1
0.6
0.5
0.1
0.4
0.3
Walk
Shop
Clean

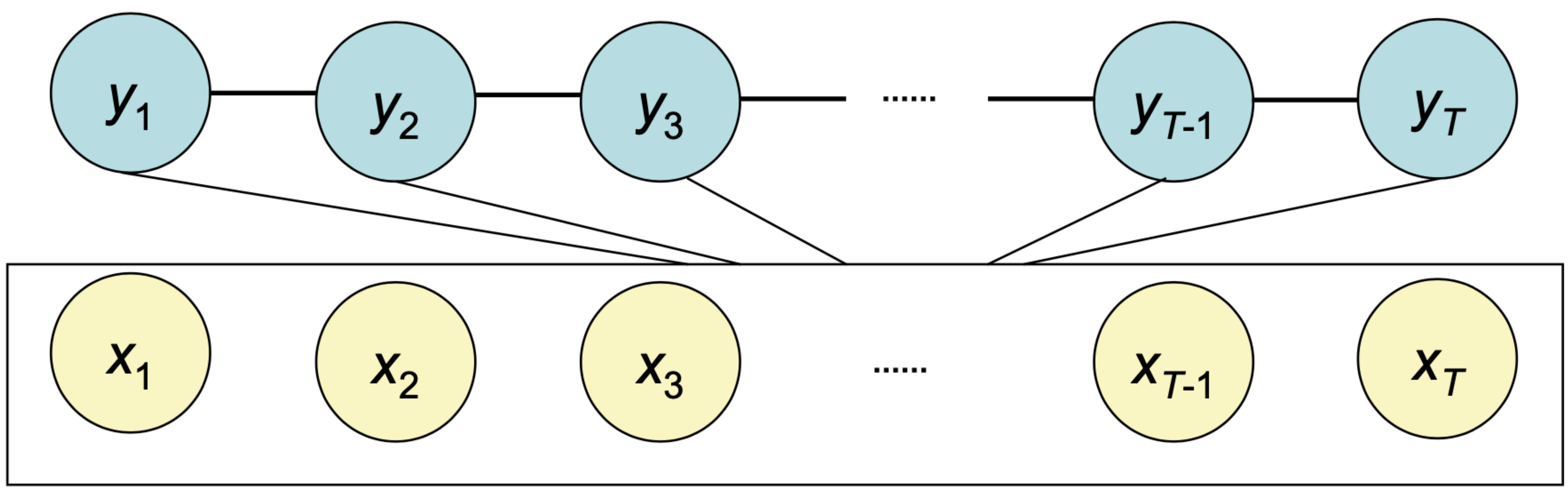
Label Bias Problem

- Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
- States with low-entropy next state distributions will take little notice of observations
  (i.e., States with a single outgoing transition effectively ignores their observation)

Conditional Random Field

- A CRF defines the conditional probability for sequences

$$P(y_1,\ldots,y_T \mid x_1,\ldots,x_T)$$



Conditional Random Field

- Can be seen as sequential version of logistic regression
- Global normalization instead of local (e.g. HMMs)
- Relax strong independence assumptions made in HMM

$$\exp(\sum_l \lambda_l \boxed{t_l(y_{i-1}, y_i, \boldsymbol{x}, i)} + \sum_k \mu_k \boxed{s_k(y_i, \boldsymbol{x}, i)})$$

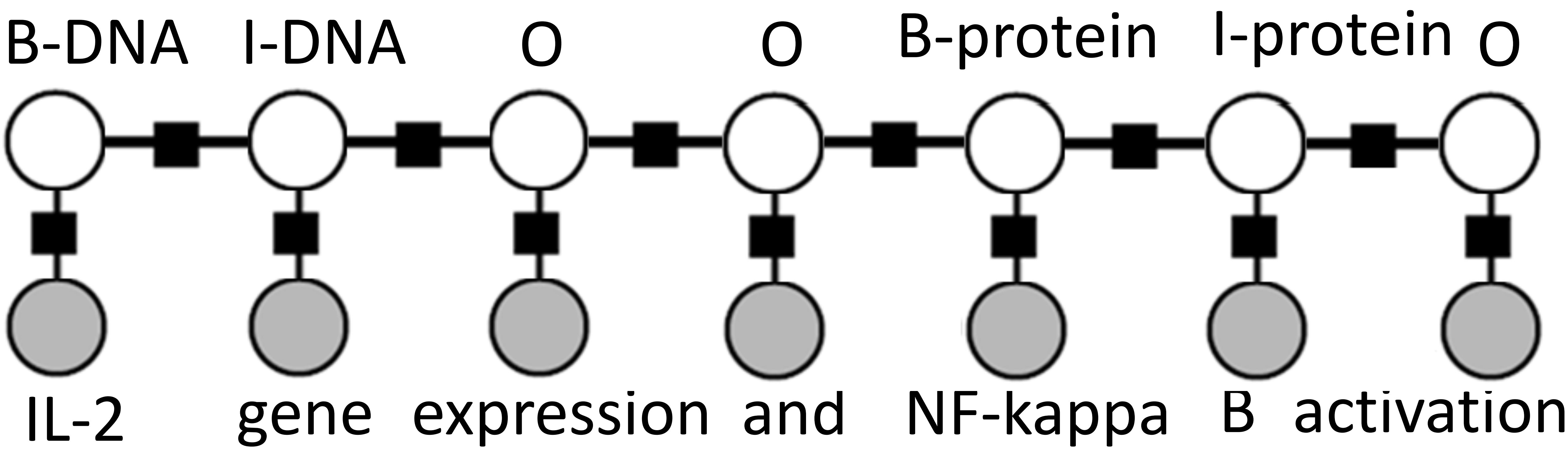
transition feature function

state feature function

Transition Feature Function

$$t_1(y_{i-1}, y_i, \boldsymbol{x}, i) = \begin{cases} 1 & \text{if } y_{i-1} = \text{B}-\text{DNA and } y_i = \text{I}-\text{DNA} \\ 0 & \text{otherwise} \end{cases}$$

$i = 2$: $t_1 = 1$;
$i = 3$: $t_1 = 0$;
$i = 4$: $t_1 = 0$;
...



State Feature Function

$$s_1(y_i, \boldsymbol{x}, i) = \begin{cases} 1 & \text{if } y_i = \text{I}-\text{DNA and } x_i = \text{gene} \\ 0 & \text{otherwise} \end{cases}$$

$i = 1: s_1 = 0;$

$i = 2: s_1 = 1;$

$i = 3: s_1 = 0;$

…