知能システム論
自然言語処理(2)
構造予測2

宮尾 祐介

yusuke@is.s.u-tokyo.ac.jp

https://mynlp.github.io/

※本講義資料は、2017年度知能システム論講義資料（佐藤一誠先生）をベースにしています

系列ラベリング (Sequence Labeling)

- 単語列などの系列データに対するラベリング問題
  - 形態素解析
  - 固有名認識（人名、地名、病名、などのラベル）
  - DNA解析
  - 音声認識
  - 動作認識
- 入力：記号列，出力：ラベル列

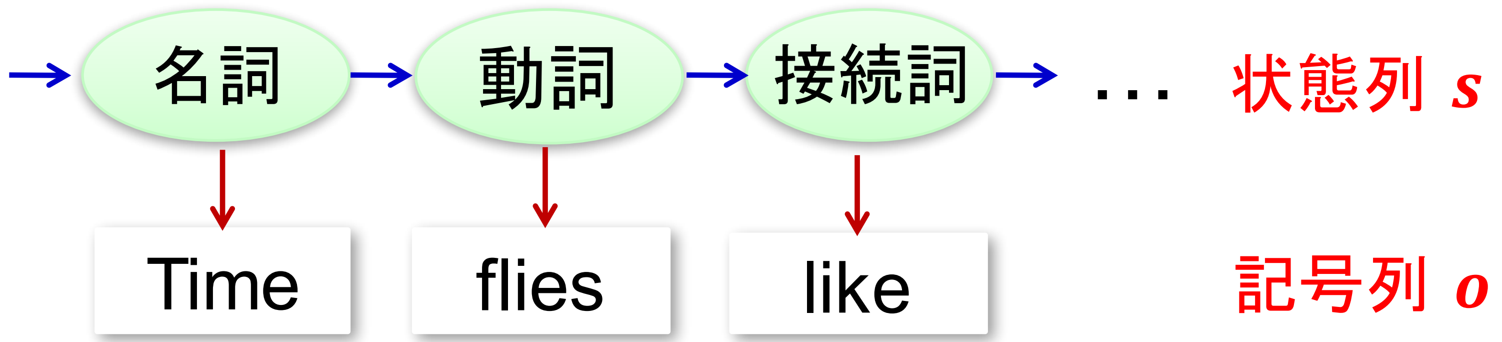
入力: $x_1, x_2, \ldots, x_T$

↓

出力: $y_1, y_2, \ldots, y_T$

隠れマルコフモデル (Hidden Markov Model; HMM)

- 系列データの確率モデル
- 状態を確率的に移動しながら，各状態から記号を出力する
  - 状態列：$\boldsymbol{s} = s_1, \cdots, s_T$
  - 記号列：$\boldsymbol{o} = o_1, \cdots, o_T$



隠れマルコフモデル (Hidden Markov Model; HMM)

S: 隠れ状態の集合 $\{1, 2, \ldots, K\}$

Σ: 出力記号の集合, e.g., $\{\mathrm{A}, \mathrm{B}\}$

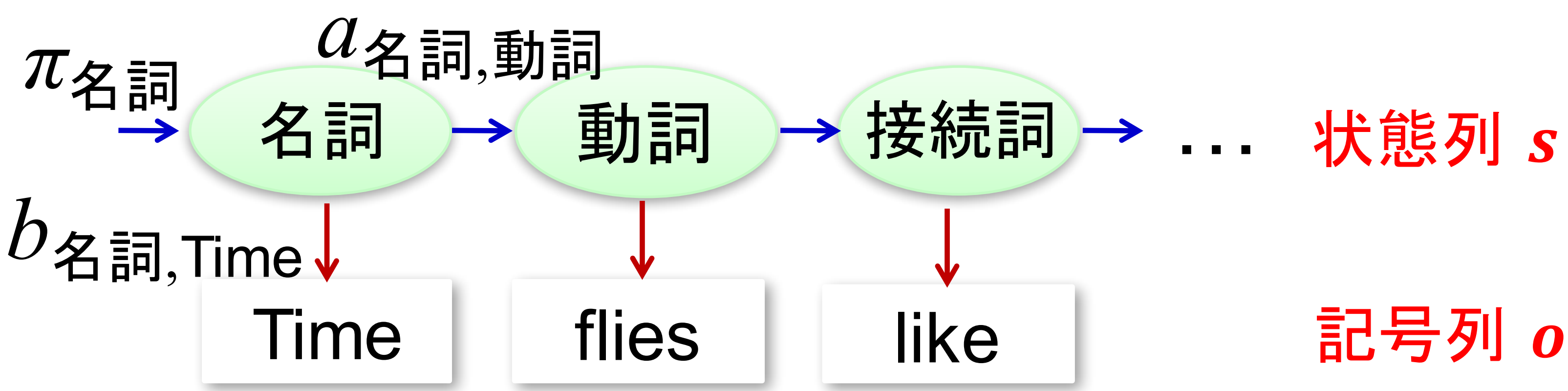
π_k: 文頭が状態 k になる確率

$a_{j,k}$: 状態 j から状態 k への遷移確率 $p(k|j)$

i.e., $\Sigma_{k \in S}\, a_{j,k} = 1$

$b_{k,o}$: 状態 k における記号 o の出力確率 $p(o|k)$

i.e., $\Sigma_{o \in \Sigma}\, b_{k,o} = 1$



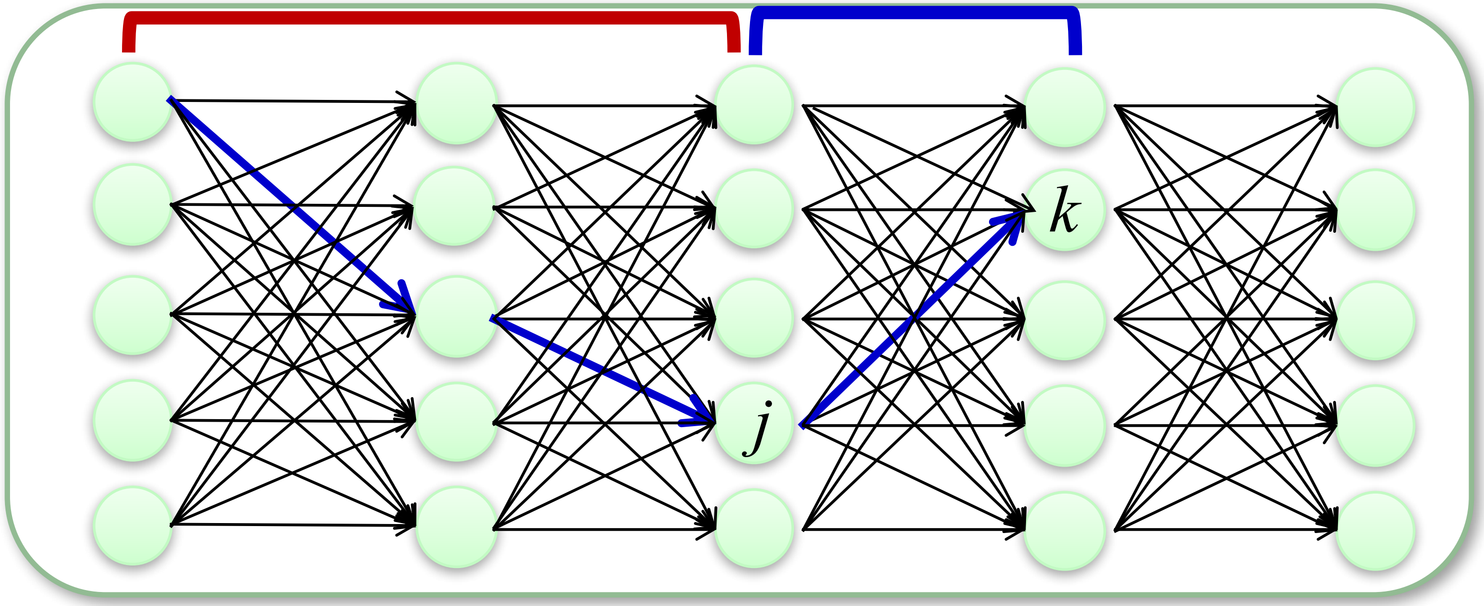
ビタビ(Viterbi)アルゴリズム

$t+1$ までの観測系列 $\boldsymbol{o}_{1:T+1}$ において、$s_{t+1}=k$ に至る状態系列の最大確率を

$$q_{t+1}(k) = \max_{\boldsymbol{s}_{1:t}} p(\boldsymbol{o}_{1:t+1}, s_{t+1}=k, \boldsymbol{s}_{1:t})$$

とすると, 以下のように再帰的に書ける

$$q_{t+1}(k) = \max_{j \in S}\left[q_t(j) a_{j,k}\right] b_{k,o_{t+1}}$$



HMMの諸問題

- 学習データを用意する
  - この作業を半自動化することも重要なテーマ

- π, a, b をデータから推定する
  - 最尤(さいゆう)推定, 事後確率最大化推定, ベイズ推定など様々
  - 今日の内容

- π, a, b から(生成)確率が最も高くなる品詞タグを推定する
  - 実用的には高速に推定することが重要
  - 先週の内容

HMM のパラメータ

S: 隠れ状態の集合 $\{1, 2, \ldots, K\}$

Σ: 出力記号の集合, e.g., $\{\mathrm{A}, \mathrm{B}\}$

π_k: 文頭が状態 k になる確率

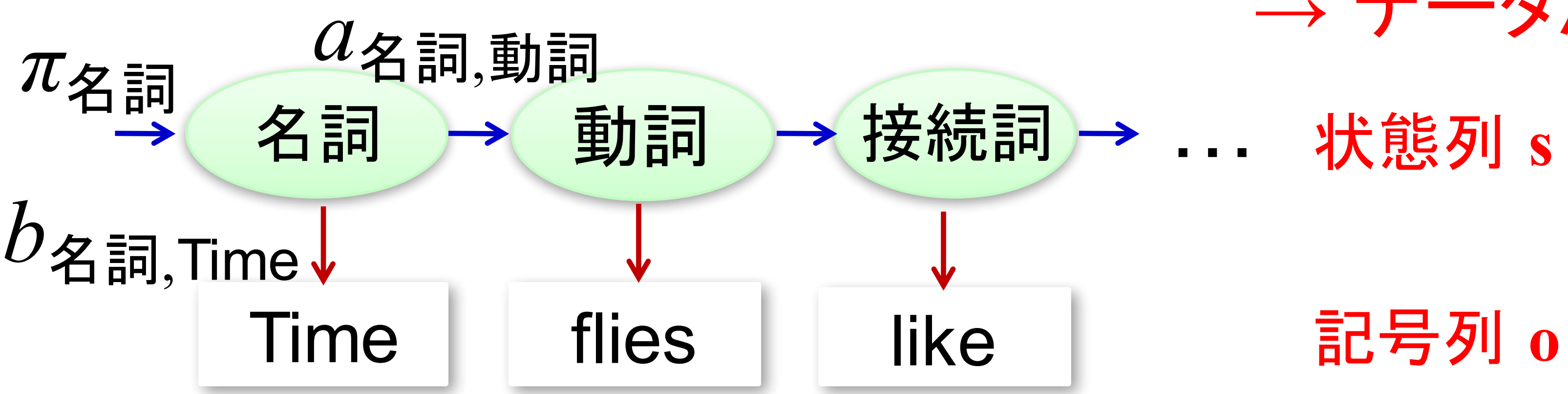
$a_{j,k}$: 状態 j から状態 k への遷移確率 $p(k|j)$

i.e., $\Sigma_{k \in S}\, a_{j,k} = 1$

$b_{k,o}$: 状態 k における記号 o の出力確率 $p(o|k)$

i.e., $\Sigma_{o \in \Sigma}\, b_{k,o} = 1$

パラメータ
→ データから学習



π

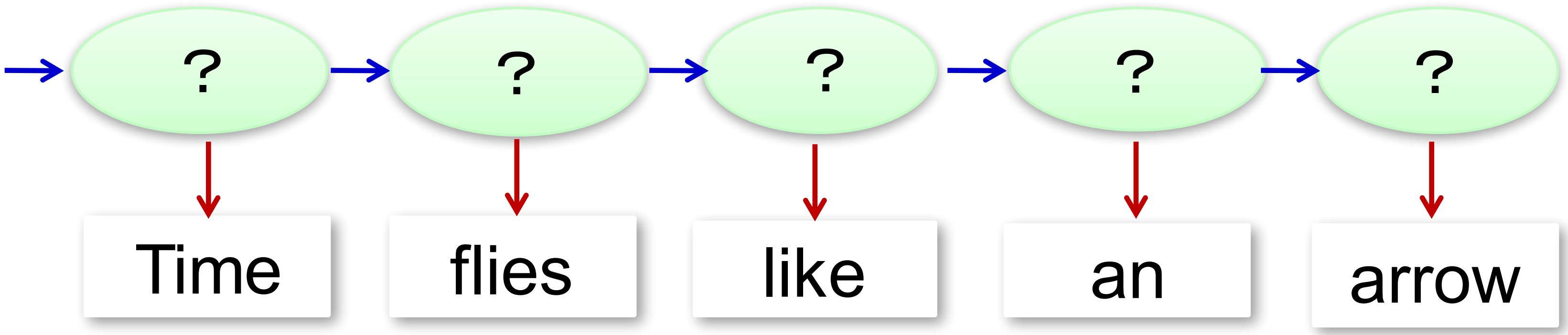
名詞	動詞	形容詞	冠詞	前置詞
0.6	0.0	0.0	0.4	0.0

a

遷移元＼遷移先	名詞	動詞	形容詞	冠詞	前置詞
名詞	0.3	0.4	0.1	0.0	0.2
動詞	0.1	0.0	0.5	0.2	0.2
形容詞	0.5	0.0	0.4	0.1	0.0
冠詞	0.7	0.0	0.0	0.0	0.3
前置詞	0.6	0.0	0.1	0.0	0.3

b

状態＼出力	an	…	like	…	time	…	arrow	…	flies	…
名詞	0	…	0.0	…	0.6	…	0.3	…	0.1	…
動詞	0.0	…	0.7	…	0.0	…	0.1	…	0.2	…
形容詞	0.0	…	1.0	…	0.0	…	0.0	…	0.0	…
冠詞	1.0	…	0.0	…	0.0	…	0.0	…	0.0	…
前置詞	0.0	…	1.0	…	0.0	…	0.0	…	0.0	…



パラメータの学習

- どうやって遷移確率、出力確率を決めるのか？
- 教師付き学習：状態列が既知のデータからパラメータを学習する
- 教師なし学習：状態列が未知のデータからパラメータを学習する

教師付き学習

- タグ付きコーパス：人手で正解（e.g. 品詞タグ）を付与したデータ
  - 人間は、理由がわからなくても正解を与えることはできる

```
Ms./NNP Haag/NNP plays/VBZ Elianti/NNP ./.
The/DT luxury/NN auto/NN maker/NN last/JJ year/NN sold/VBD
1,214/CD cars/NNS in/IN the/DT U.S./NNP
The/DT new/JJ rate/NN will/MD be/VB payable/JJ Feb./NNP
15/CD ./.
```

Penn Treebank 2 より引用

- 教師付き学習＝タグ付きコーパスからパラメータを学習する

最尤推定法

- 学習データの（対数）尤度を最大化するようにパラメータを決定する
  - 尤度：学習データ $D=\{(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)})\}_{i=1}^{N}$ の生成確率

$$L(\theta|D)=\log\prod_{i=1}^{N}p(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)}|\theta)=\sum_{i=1}^{N}\log p(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)}|\theta)$$

← 目的関数

$$\theta^{*}=\underset{\theta}{\operatorname{argmax}}\sum_{i=1}^{N}\log p(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)}|\theta)$$

演習1：HMM のパラメータの最尤推定

- HMM のパラメータ π_k, $a_{j,k}$, $b_{k,o}$ の最尤推定量を求めよ

$$p(\boldsymbol{o}_{1:T}, \boldsymbol{s}_{1:T}) = p(s_1)\prod_{t=1}^{T} p(s_{t+1}|s_t)p(o_t|s_t) = \pi_{s_1}\prod_{t=1}^{T} a_{s_t,s_{t+1}} b_{s_t,o_t}$$

$$\text{s.t.} \quad \sum_{k\in S}\pi_k = 1, \quad \sum_{k\in S} a_{j,k} = 1, \quad \sum_{o\in\Sigma} b_{k,o} = 1$$

$$\log p(\boldsymbol{o}_{1:T}, \boldsymbol{s}_{1:T}) = \log \pi_{s_1}\prod_{t=1}^{T} a_{s_t,s_{t+1}} b_{s_t,o_t}$$

$$= \log \pi_{s_1} + \sum_{t=1}^{T}\log a_{s_t,s_{t+1}} + \sum_{t=1}^{T}\log b_{s_t,o_t}$$

HMM の最尤推定

$$F(\pi_k, a_{j,k}, b_{k,o}, \lambda_\pi, \lambda_{a_{j\cdot}}, \lambda_{b_{k\cdot}})$$

$$= L\left(\pi_k, a_{j,k}, b_{k,o} \middle| \left\{\left(\boldsymbol{o}^{(i)}, \boldsymbol{s}^{(i)}\right)\right\}\right) + \lambda_\pi\left(\sum_{k\in S} 1 - \pi_k\right)$$

$$+\sum_{j\in S}\lambda_{a_{j\cdot}}\left(\sum_{k\in S} 1 - a_{j,k}\right) + \sum_{k\in S}\lambda_{b_{k\cdot}}\left(\sum_{o\in\Sigma} 1 - b_{k,o}\right)$$

$$\frac{\partial}{\partial\,\cdot}F\left(\pi_k, a_{t,k}, b_{k,o}, \lambda_\pi, \lambda_{a_{j\cdot}}, \lambda_{b_{k\cdot}}\right) = 0$$ とすると、

HMM の最尤推定

$$\frac{\partial}{\partial a_{j,k}} F(\pi_k, a_{t,k}, b_{k,o}, \lambda_\pi, \lambda_{a_{j\cdot}}, \lambda_{b_{k\cdot}})$$

$$= \sum_{i=1}^{N} \sum_{t=1}^{T} \frac{\delta(s_t^{(i)} = j, s_{t+1}^{(i)} = k)}{a_{j,k}} - \lambda_{a_{j\cdot}}$$

$$= \frac{C(j,k)}{a_{j,k}} - \lambda_{a_{j\cdot}}$$

$C(j,k)$: D 中の $\langle j,k \rangle$ の出現回数

$$\frac{C(j,k)}{a_{j,k}} - \lambda_{a_{j\cdot}} = 0, \qquad 1 - \sum_{k \in S} a_{j,k} = 0$$

$$a_{j,k} = \frac{C(j,k)}{\sum_{k \in S} C(j,k)} \qquad \leftarrow \; p(k|j)$$

HMM の最尤推定

- 遷移確率:品詞の並び $\langle t, k \rangle$ を数える

$$\pi_k = \frac{C(s_1 = k)}{\sum_{k \in S} C(s_1 = k)} \qquad a_{j,k} = \frac{C(j,k)}{\sum_{k \in S} C(j,k)}$$

- 出力確率:品詞と単語のペア $\langle k, o \rangle$ を数える

$$b_{k,o} = \frac{C(k,o)}{\sum_{o \in \Sigma} C(k,o)}$$

教師なし学習

- タグなしコーパス（ただの単語列）からパラメータを学習する
  - 学習データ $D = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$
- 最尤推定法

$$\theta^* = \operatorname*{argmax}_{\theta} \sum_{i=1}^{N} \log p(\boldsymbol{x}^{(i)}|\theta)$$

$$= \operatorname*{argmax}_{\theta} \sum_{i=1}^{N} \underline{\log \sum_{\boldsymbol{z}} p(\boldsymbol{x}^{(i)}, \boldsymbol{z}|\theta)}$$

隠れ変数
（潜在変数）

これは解析的に最適化できない

Baum-Welch アルゴリズム

- 記号列 $D = \{\boldsymbol{o}^{(i)}\}_{i=1}^{N}$ のみから HMM のパラメータを推定する手法
  - EM (Expectation-Maximization) アルゴリズムの一種
- アルゴリズム:
  - パラメータ π_k, $a_{j,k}$, $b_{k,o}$ をランダムに初期化
  - **Eステップ**: 現在のパラメータで隠れ変数 $\boldsymbol{s}$ の期待値を計算
  - **Mステップ**: $\boldsymbol{s}$ の期待値を使ってパラメータを更新
  - 収束するまでEステップとMステップを繰り返す

Baum-Welch アルゴリズム

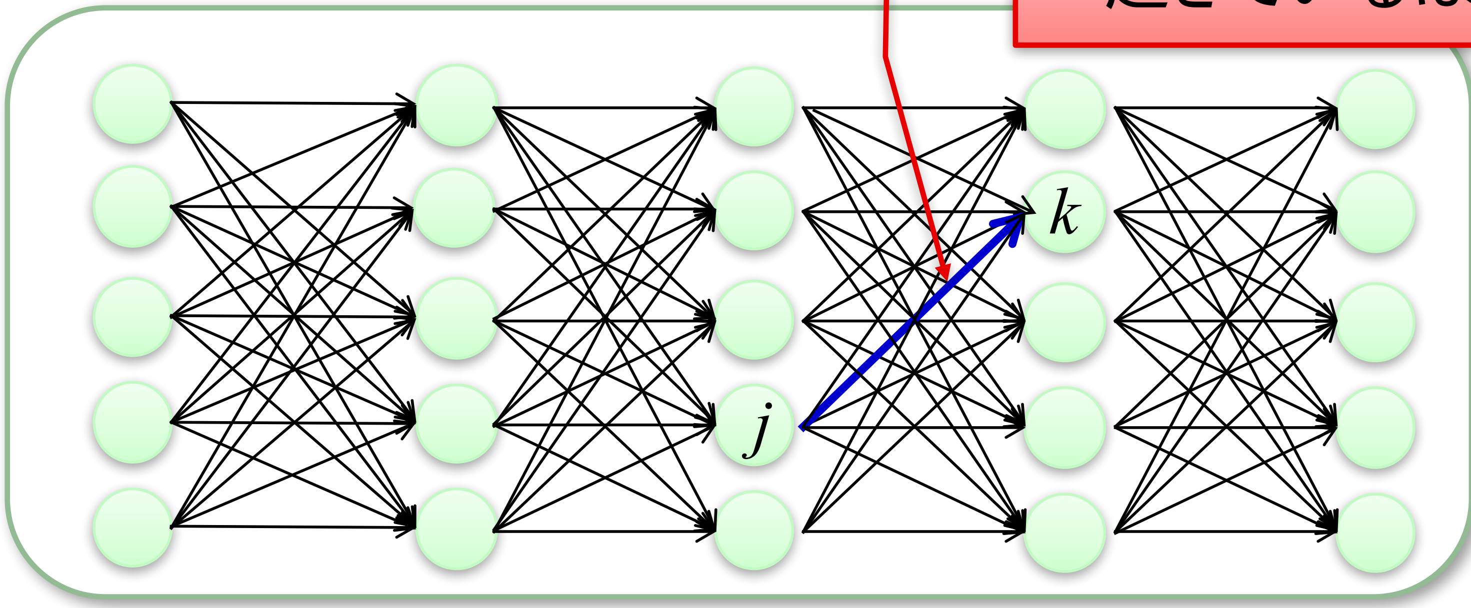
- Eステップ：現在のパラメータ $\theta^{\mathrm{old}} = \langle \pi_k^{\mathrm{old}}, a_{j,k}^{\mathrm{old}}, b_{k,o}^{\mathrm{old}} \rangle$ で、状態遷移の期待値を計算

$$\gamma_t(k) = p(s_t = k | \boldsymbol{o}_{1:T}, \theta^{\mathrm{old}})$$

$$\xi_t(j,k) = p(s_t = j, s_{t+1} = k | \boldsymbol{o}_{1:T}, \theta^{\mathrm{old}})$$

- これの計算方法は後述

この遷移がどれくらい起きているはずか？



Baum-Welch アルゴリズム

- Mステップ：γ_t, ξ_t を使って、パラメータを更新

教師あり学習の場合

$$\pi_k^{\text{new}} = \gamma_1(k) \qquad \pi_k = \frac{C(s_1 = k)}{\sum_{k \in S} C(s_1 = k)}$$

$$a_{j,k}^{\text{new}} = \frac{\sum_t \xi_t(j,k)}{\sum_{k \in S} \sum_t \xi_t(j,k)} \qquad a_{j,k} = \frac{C(j,k)}{\sum_{k \in S} C(j,k)}$$

$$b_{k,o}^{\text{new}} = \frac{\sum_{t \text{ s.t. } o_t = o} \gamma_t(k)}{\sum_t \gamma_t(k)} \qquad b_{k,o} = \frac{C(k,o)}{\sum_{o \in \Sigma} C(k,o)}$$

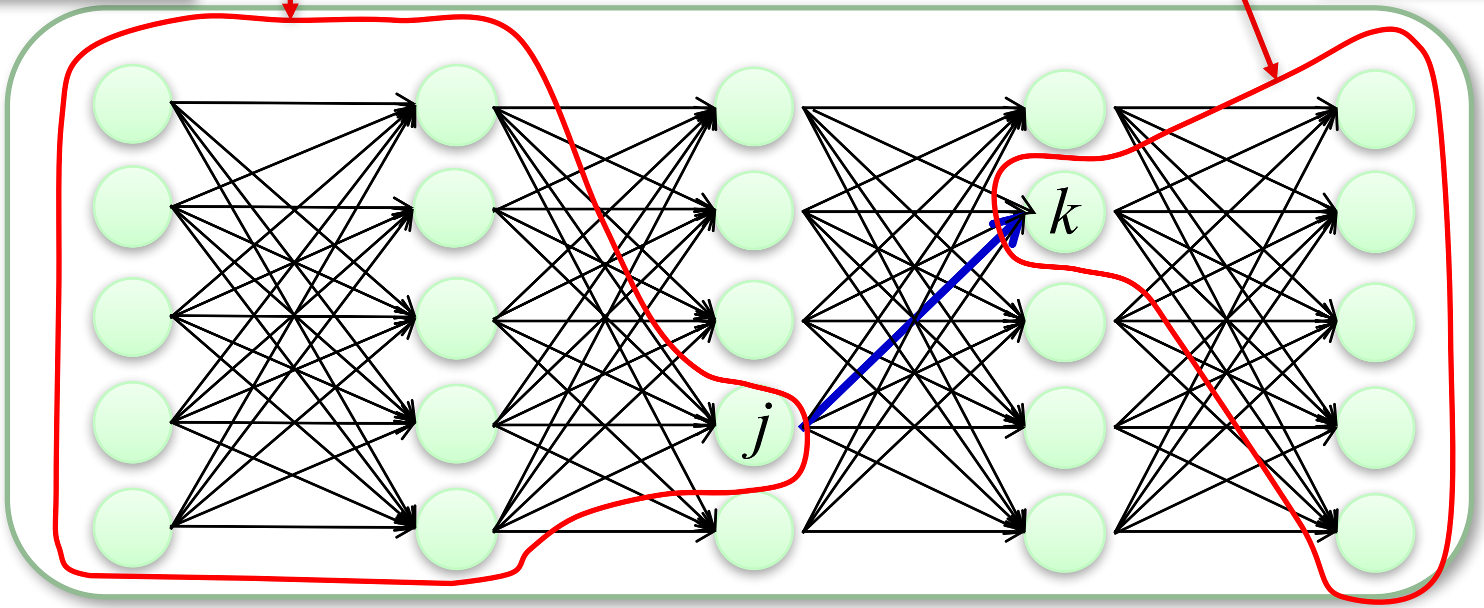
期待値の計算

- 全ての状態遷移のうち、$s_t = j, s_{t+1} = k$ を通る系列について確率の和をとる → 指数爆発

$$\xi_t(j,k) = p\left(s_t = j, s_{t+1} = k \middle| \boldsymbol{o}_{1:T}, \theta^{\mathrm{old}}\right)$$
$$= \frac{\sum_{\boldsymbol{s}\ \mathrm{s.t.}\ s_t=j, s_{t+1}=k} p(\boldsymbol{o}_{1:T}, \boldsymbol{s} | \theta^{\mathrm{old}})}{p(\boldsymbol{o}_{1:T} | \theta^{\mathrm{old}})}$$

j までの全ての系列

k から先の全ての系列



演習2:期待値の計算

- $\xi_t(j,k)$ を計算する方法を考えよ

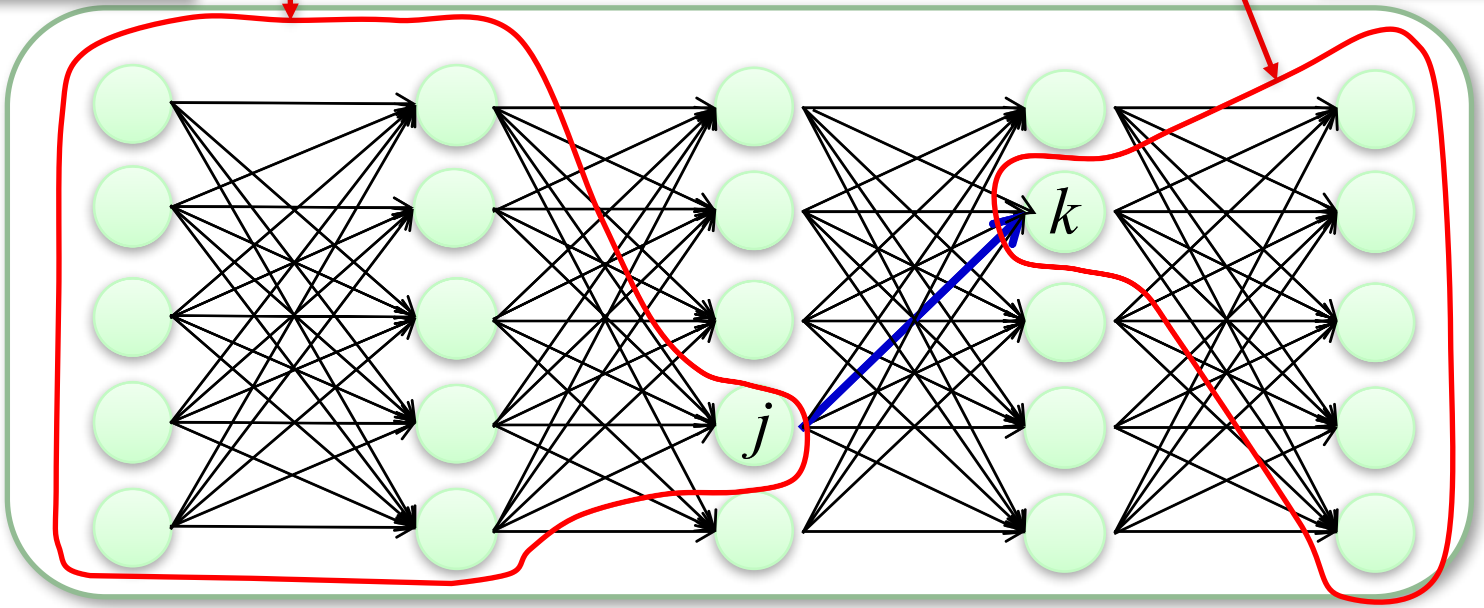
$$\xi_t(j,k) = p\left(s_t = j, s_{t+1} = k \middle| \boldsymbol{o}_{1:T}, \theta^{\mathrm{old}}\right)$$

$$\alpha_t(j) = p\left(\boldsymbol{o}_{1:t}, s_t = j \middle| \theta^{\mathrm{old}}\right)$$

$$\beta_t(j) = p\left(\boldsymbol{o}_{t+1:T} \middle| s_t = j, \theta^{\mathrm{old}}\right)$$

j までの全ての系列

k から先の全ての系列

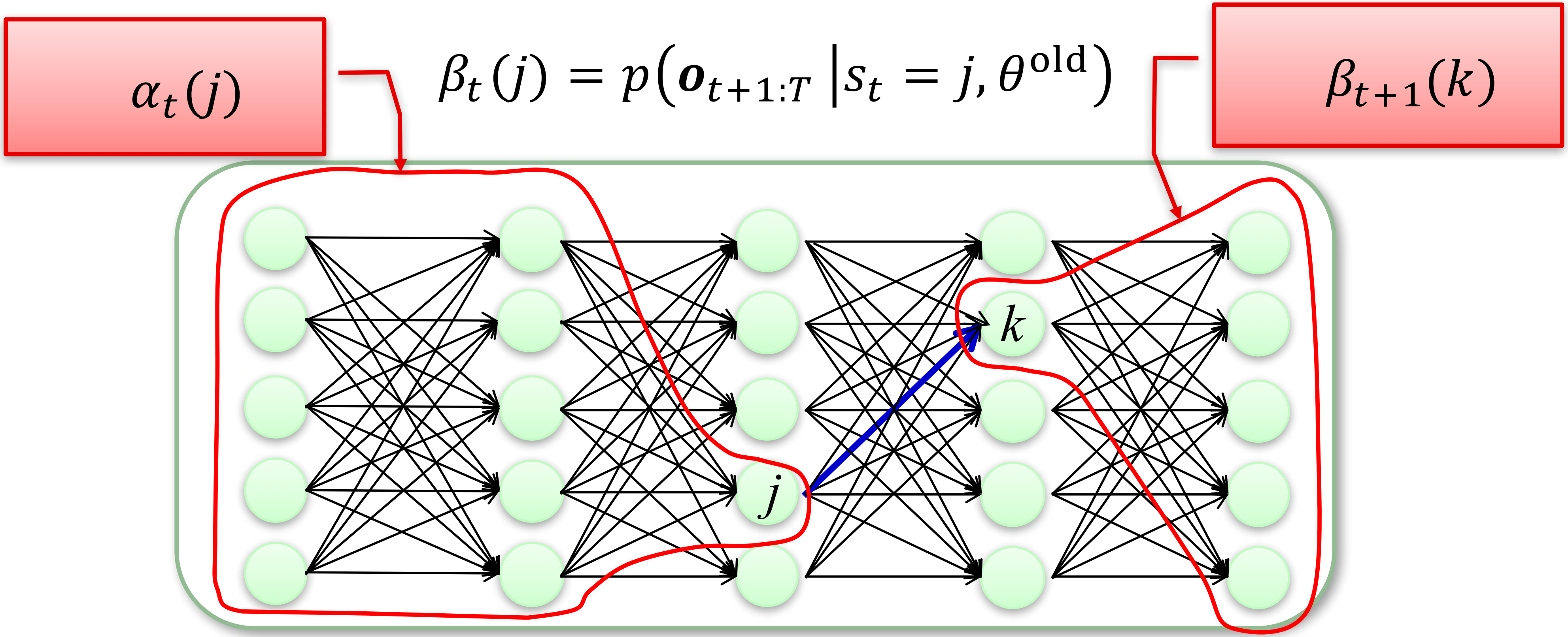


前向き・後向き確率

- 前向き確率：時刻 t に状態 j に至る全ての系列の確率の和
- 後向き確率：時刻 t の状態 j から最後に至る全ての系列の確率の和

$$\alpha_t(j) = p\left(\boldsymbol{o}_{1:t}, s_t = j \,\middle|\, \theta^{\text{old}}\right)$$

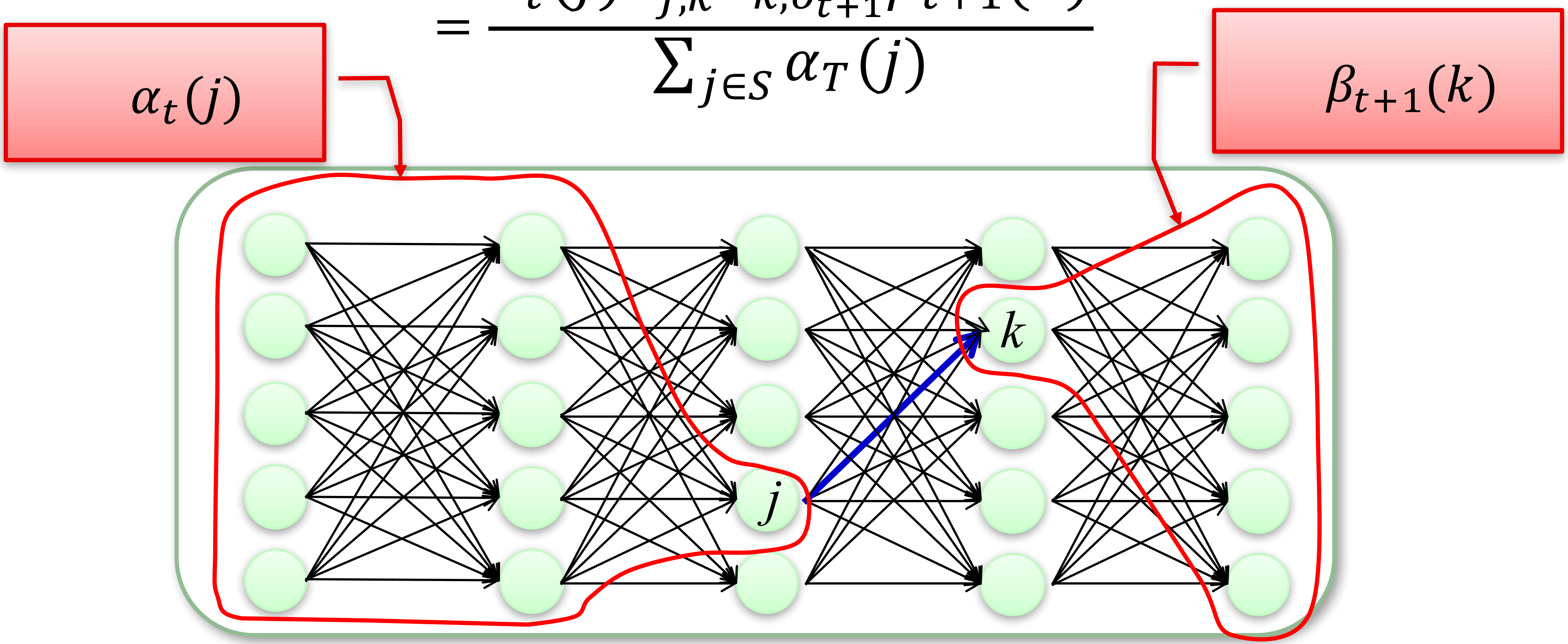
$$\beta_t(j) = p\left(\boldsymbol{o}_{t+1:T} \,\middle|\, s_t = j, \theta^{\text{old}}\right)$$



前向き・後向き確率

- $\gamma_t(k), \xi_t(j,k)$ は、前向き確率と後向き確率から計算できる

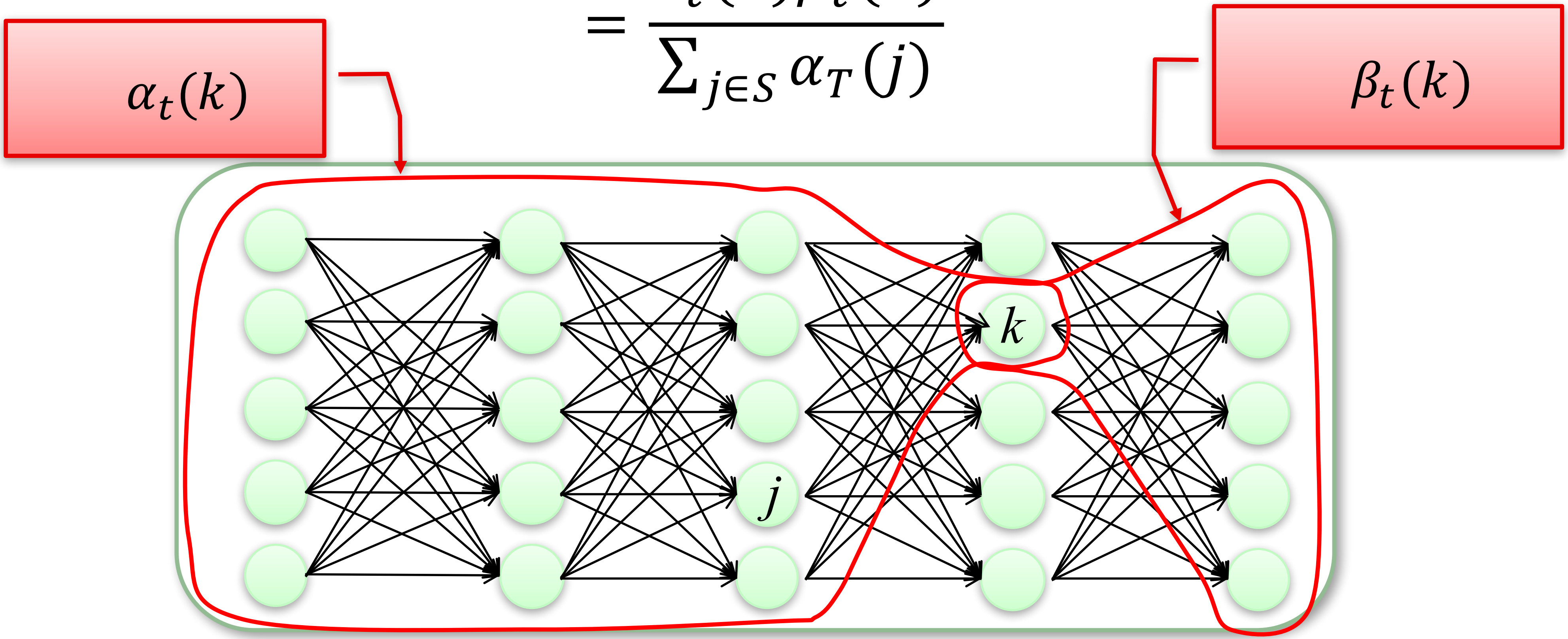
$$\xi_t(j,k) = p\left(s_t = j, s_{t+1} = k \middle| \boldsymbol{o}_{1:T}, \theta^{\mathrm{old}}\right)$$
$$= \frac{\alpha_t(j) a_{j,k} b_{k,o_{t+1}} \beta_{t+1}(k)}{\sum_{j \in S} \alpha_T(j)}$$



前向き･後向き確率

- $\gamma_t(k), \xi_t(j,k)$ は、前向き確率と後向き確率から計算できる

$$\gamma_t(k) = p\left(s_t = k \middle| \boldsymbol{o}_{1:T}, \theta^{\mathrm{old}}\right)$$
$$= \frac{\alpha_t(k)\beta_t(k)}{\sum_{j \in S} \alpha_T(j)}$$

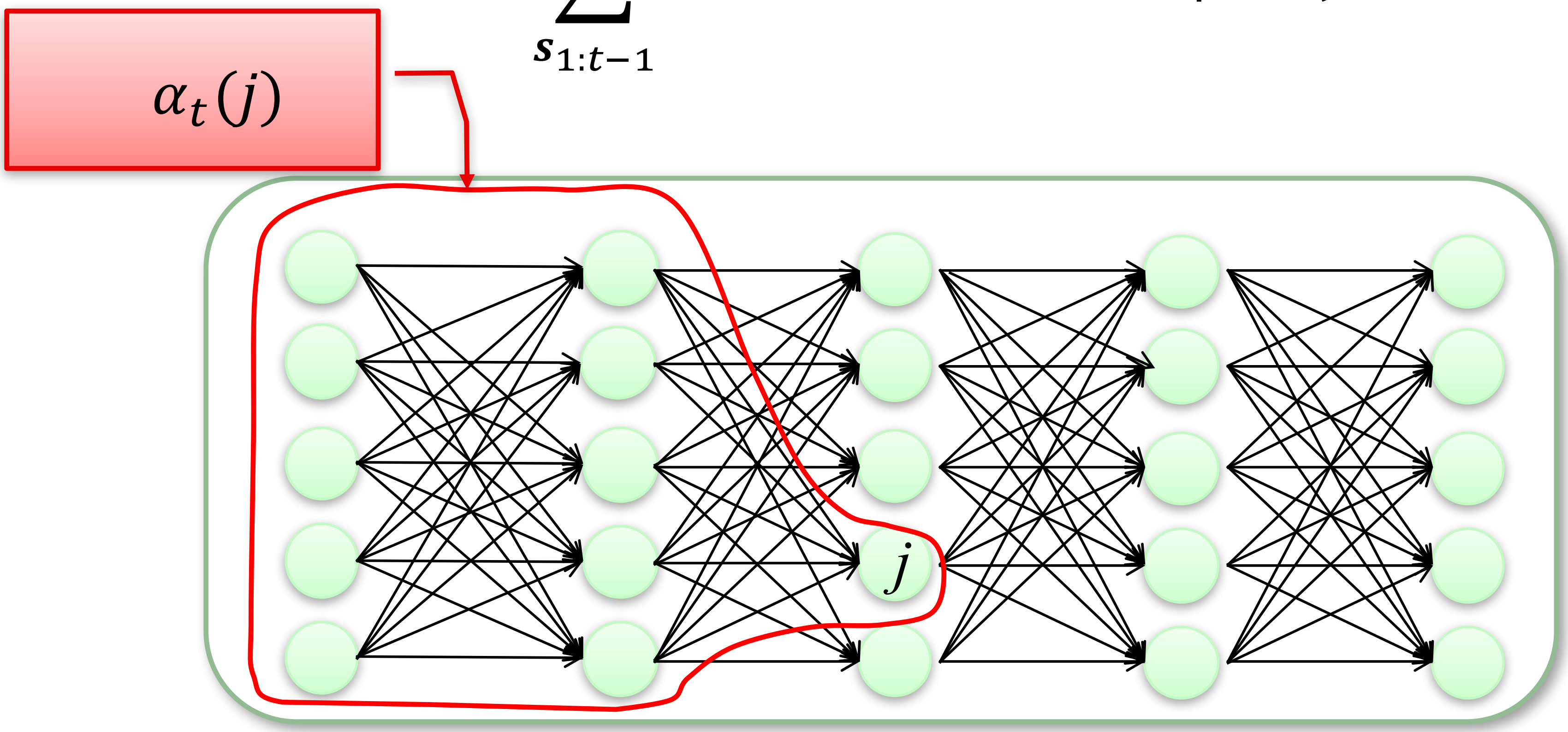


前向き確率の計算

- 前向き確率：時刻 t に状態 j に至る全ての系列の確率の和

$$\alpha_t(j) = p\left(\boldsymbol{o}_{1:t}, s_t = j \,\middle|\, \theta^{\mathrm{old}}\right)$$

$$= \sum_{\boldsymbol{s}_{1:t-1}} p\left(\boldsymbol{o}_{1:t}, s_t = j, \boldsymbol{s}_{1:t-1} \middle| \theta^{\mathrm{old}}\right) \rightarrow \text{指数爆発}$$



動的計画法

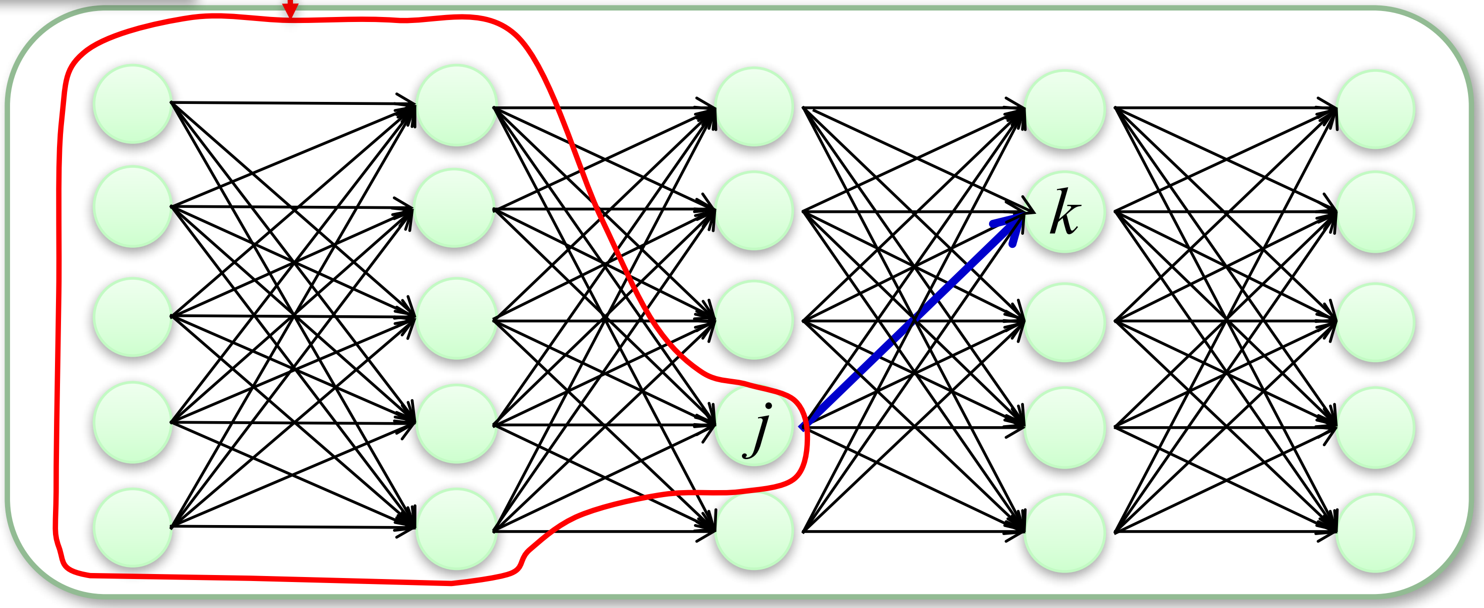
- 前向き確率 : 時刻 t に状態 j に至る全ての系列の確率の和

$$\alpha_{t+1}(k) = p\left(\boldsymbol{o}_{1:t+1}, s_{t+1} = k \,\middle|\, \theta^{\mathrm{old}}\right)$$

$$= \sum_{\boldsymbol{s}_{1:t}} p(\boldsymbol{o}_{1:t+1}, s_{t+1} = k, \boldsymbol{s}_{1:t} | \theta^{\mathrm{old}})$$

$$= \sum_{j \in S} \left[\alpha_t(j) a_{j,k}\right] b_{k,o_{t+1}}$$

$\alpha_t(j)$



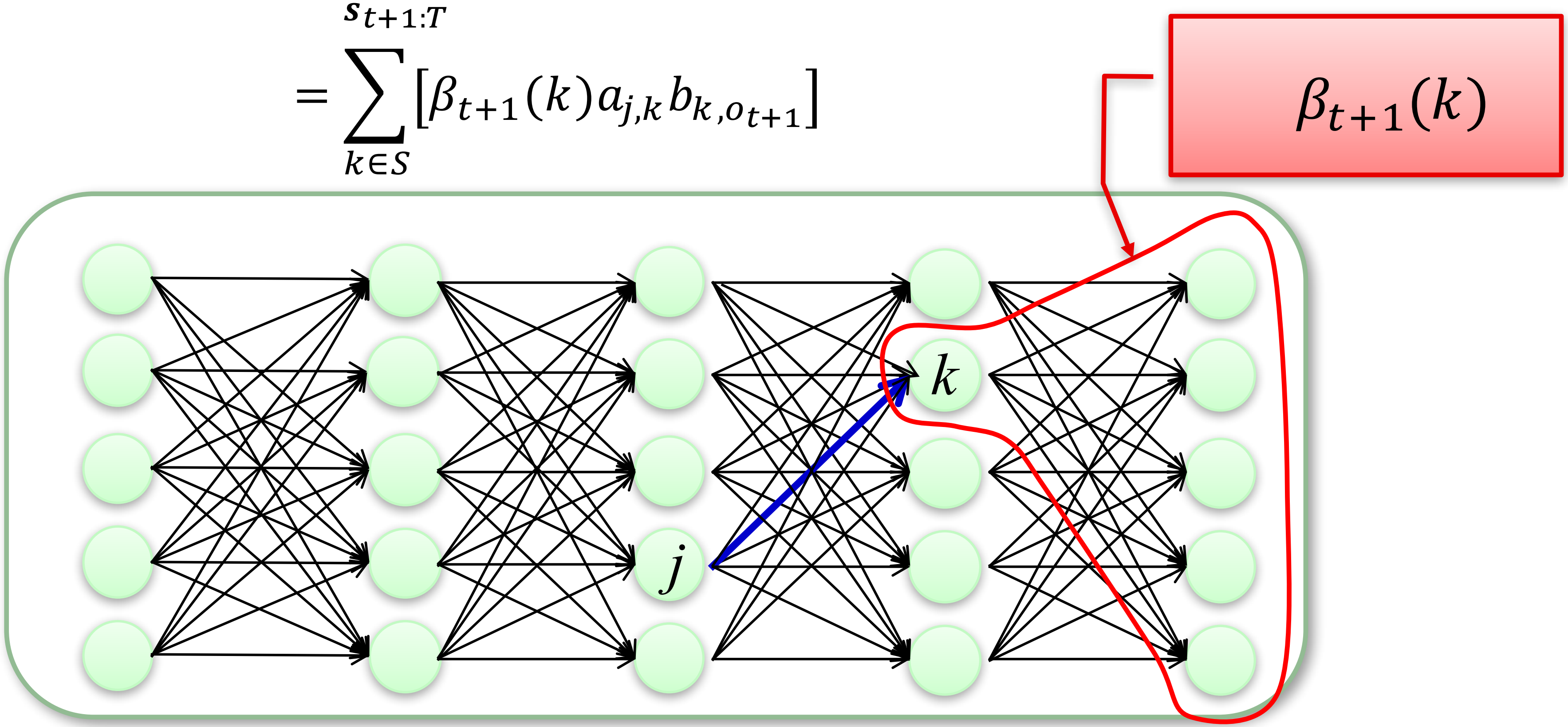
後向き確率の計算

- 後向き確率：時刻 t の状態 j から最後に至る全ての系列の確率の和

$$\beta_t(j) = p\left(\boldsymbol{o}_{t+1:T} \middle| s_t = j, \theta^{\text{old}}\right)$$

$$= \sum_{\boldsymbol{s}_{t+1:T}} p(\boldsymbol{o}_{t+1:T}, \boldsymbol{s}_{t+1:T} | s_t = j, \theta^{\text{old}})$$

$$= \sum_{k \in S} [\beta_{t+1}(k) a_{j,k} b_{k,o_{t+1}}]$$



動的計画法のポイント

$$
\begin{aligned}
\alpha_{t+1}(k) &= p\left(\boldsymbol{o}_{1:t+1}, s_{t+1} = k \,\middle|\, \theta^{\text{old}}\right) \\
&= \sum_{\boldsymbol{s}_{1:t}} p\left(\boldsymbol{o}_{1:t+1}, s_{t+1} = k, \boldsymbol{s}_{1:t} \,\middle|\, \theta^{\text{old}}\right) \\
&= \sum_{j \in S} \left[\alpha_t(j)\, a_{j,k}\right] b_{k,o_{t+1}}
\end{aligned}
$$

- 分配法則によって，積の和を和の積にする
  - $X_1A + X_2A + X_3A = (X_1 + X_2 + X_3)A$
  - $X_1AY_1 + X_2AY_1 + X_3AY_1$
    $+X_1AY_2 + X_2AY_2 + X_3AY_2$
    $+X_1AY_3 + X_2AY_3 + X_3AY_3$
    $=(X_1 + X_2 + X_3)A(Y_1 + Y_2 + Y_3)$

ビタビアルゴリズムと前向き・後向きアルゴリズム

- アルゴリズムはほとんど同じ
  - ビタビ：積の最大値（あるいは和の最大値）

$$q_{t+1}(k) = \max_{j \in S} \left[q_t(j)\, a_{j,k} \right] b_{k,o_{t+1}}$$

  - 前向き・後向き：積の和

$$\alpha_{t+1}(k) = \sum_{j \in S} \left[\alpha_t(j)\, a_{j,k} \right] b_{k,o_{t+1}}$$

- 両方とも分配法則を利用している
  → 半環 (semi-ring) なら同じアルゴリズムが適用可
  - $\max(ax, ay) = a \max(x, y)$
  - $\mathrm{sum}(ax, ay) = a\, \mathrm{sum}(x, y)$
- 構造予測における動的計画法はだいたいこの形

構造予測 (Structured Prediction)

- 出力がスカラー(数値、ラベル)ではなく、離散構造
  - 系列構造、木構造、…
  - $\boldsymbol{y}^* = f(\boldsymbol{x}) = \operatorname{argmax}_{\boldsymbol{y}}\, g(\boldsymbol{x}, \boldsymbol{y})$

音声認識: 系列→系列

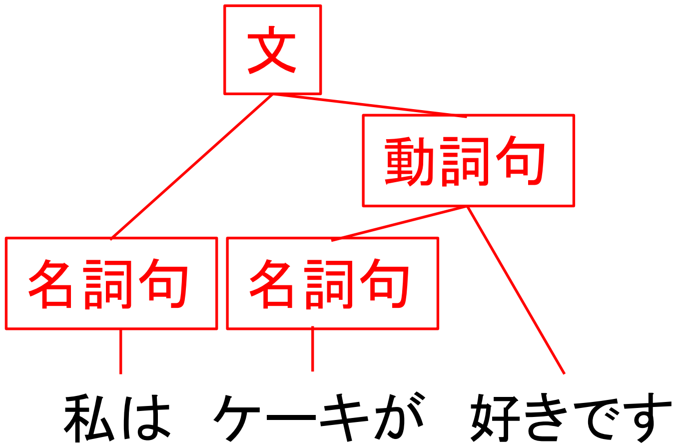
→ ワ レ ワ レ ハ …

機械翻訳: 系列→系列

今朝はおもちを食べました。

→ I ate rice cakes this morning.

構文解析: 系列→木

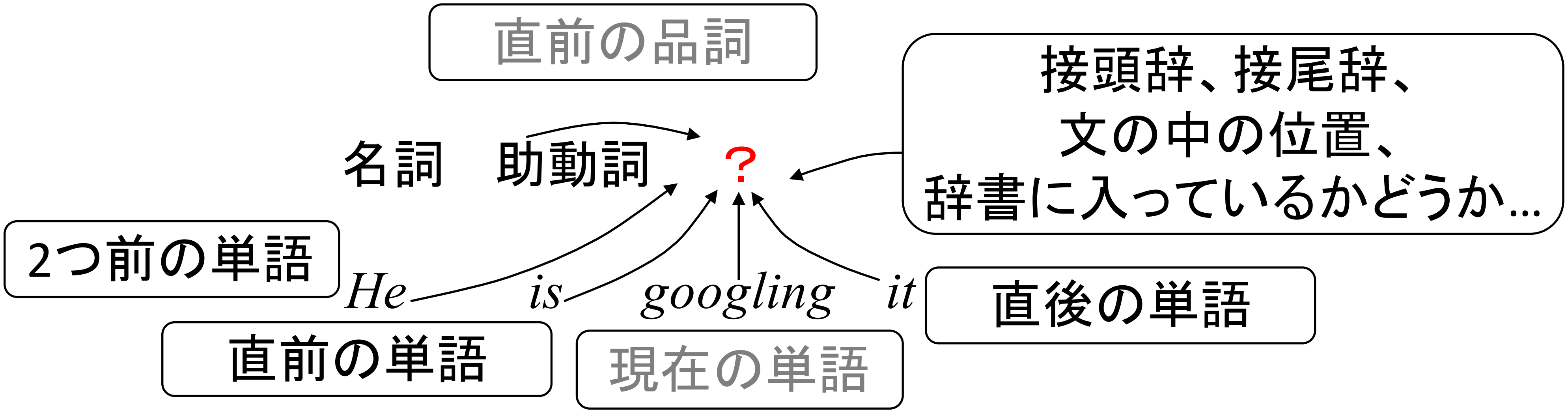


画像説明文生成: 二次元配列→系列

→ 海辺におしゃれな灯台が建っています。

機械学習を使いたい

- 品詞を当てるのに有用な特徴がいろいろある
  - 接尾辞が *-ing*
  - 直前の単語が “*is*”
- いろいろな特徴を利用して、より高精度に品詞を予測したい
  → 機械学習（SVM、etc.）

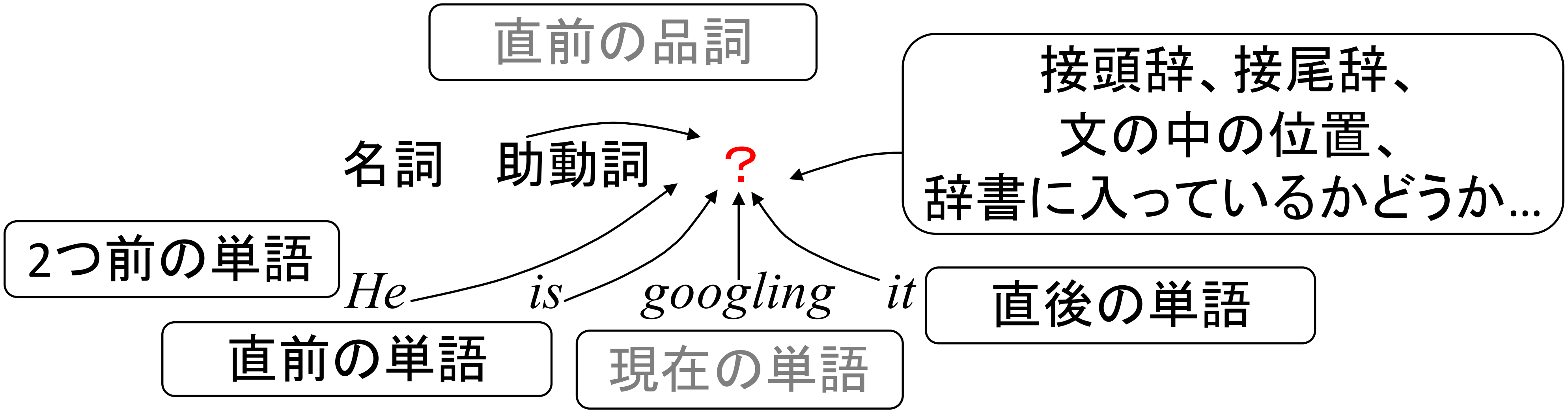


機械学習を使った品詞解析

- 入力 : 単語列 $\boldsymbol{o}$、出力 : 品詞列 $\boldsymbol{s}$
- 分類器で $\boldsymbol{s}$ を予測する＝スコアが最大の $\mathbf{s}$ を求める → 指数爆発

$$\boldsymbol{s}^* = \underset{\boldsymbol{s}}{\operatorname{argmax}}\, g_{\boldsymbol{w}}(\boldsymbol{o}, \boldsymbol{s}) = \underset{\boldsymbol{s}}{\operatorname{argmax}}\, \boldsymbol{w}^{\mathrm{T}} \underline{\boldsymbol{\varphi}(\boldsymbol{o}, \boldsymbol{s})}$$

特徴ベクトル



特徴ベクトル（素性ベクトル）

- x, y の「様々な特徴」を特徴ベクトル $\boldsymbol{\varphi}(x, y)$ で表す
  - 各要素が「特徴」の有無を表す
  - 二値特徴：x, y が特徴 m を持っているなら $\varphi_m(x, y) = 1$
- 現実世界の様々なモノ（テキスト，画像，etc.）を特徴空間に写像する

$\boldsymbol{\varphi}$(He is googling it, 名詞 助動詞 動詞 名詞)
$= \langle 0,0,0,1,0,0,1,0, \dots \dots \dots, 0,1, \dots \dots, 0,1,0 \rangle$

3単語目は直前が *is* で動詞

1単語目は *He* で名詞

1単語目は名詞で2単語目は助動詞

3単語目は –*ing* で動詞

ログ線形モデル (Log-linear Model)

- ロジスティック回帰、最大エントロピーモデルとも呼ばれる
- 線形モデルで確率を定義

$$p(y|x) = \frac{1}{Z(x)} \exp(\boldsymbol{w}^{\mathrm{T}} \underline{\boldsymbol{\varphi}(x,y)})$$

特徴ベクトル

- パラメータ $\boldsymbol{w}$ を最尤推定

$$\boldsymbol{w}^* = \operatorname*{argmax}_{\boldsymbol{w}} L(\boldsymbol{w}|D)$$

$$L(\boldsymbol{w}|D) = \sum_{i=1}^{N} \log p\left(y^{(i)} \middle| x^{(i)}\right)$$

$$= \sum_{i=1}^{N} \log \frac{\exp(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}(x^{(i)}, y^{(i)}))}{\sum_y \exp(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}(x^{(i)}, y))}$$

$$= \sum_{i=1}^{N} \left[\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}\left(x^{(i)}, y^{(i)}\right) - \log \sum_{y} \exp(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}(x^{(i)}, y)) \right]$$

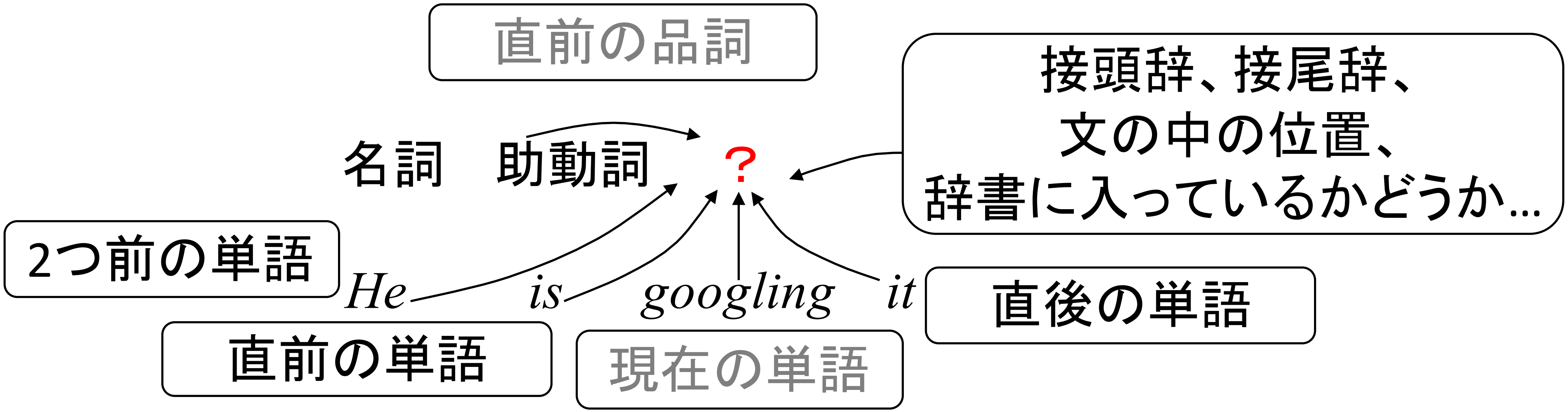
目的関数

ログ線形モデルによる品詞タグ付け

- 品詞タグ付け＝入力単語列 $\boldsymbol{o}_{1:T}$ に対し、確率が最大となる品詞列 $\boldsymbol{s}_{1:T}$ を求める

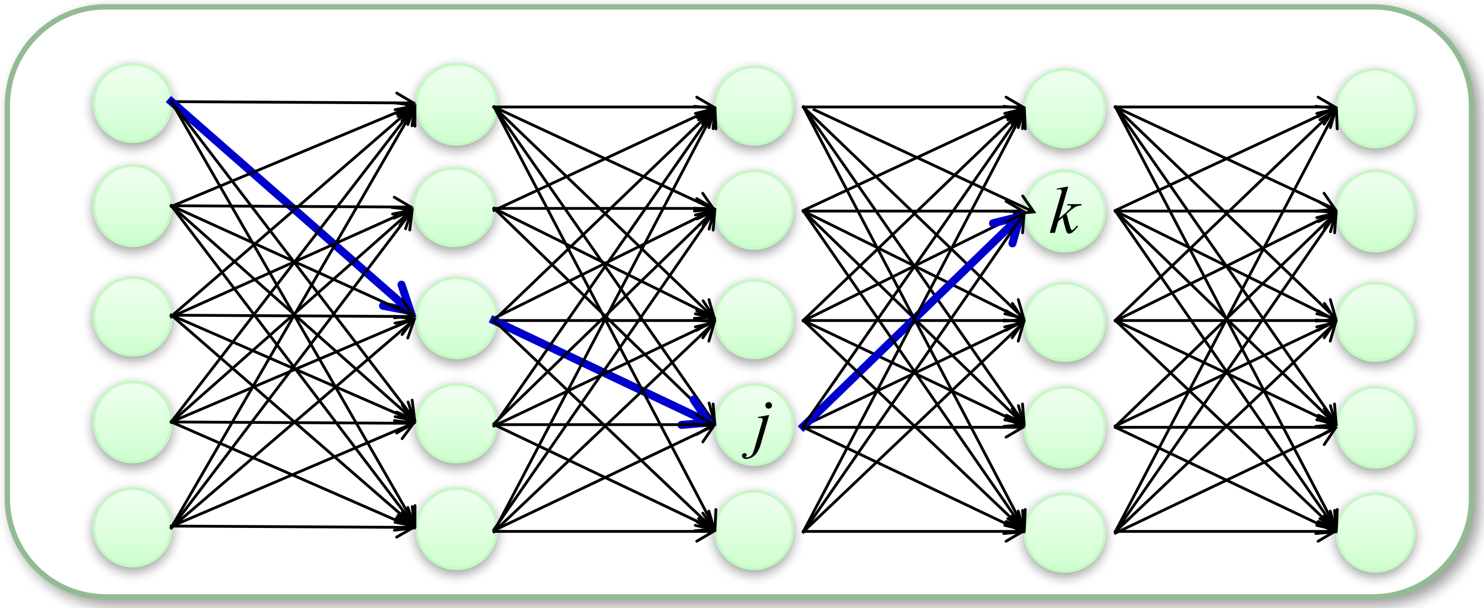
$$
\begin{aligned}
\boldsymbol{s}^* &= \underset{\boldsymbol{s}}{\operatorname{argmax}}\, p(\boldsymbol{s}_{1:T} | \boldsymbol{o}_{1:T}) \\
&= \underset{\boldsymbol{s}}{\operatorname{argmax}}\, \exp(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}(\boldsymbol{o}_{1:T}, \boldsymbol{s}_{1:T})) \\
&= \underset{\boldsymbol{s}}{\operatorname{argmax}}\, \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}(\boldsymbol{o}_{1:T}, \boldsymbol{s}_{1:T})
\end{aligned}
$$

← どうやって求める？



条件付き確率場 (Conditional Random Fields; CRF)

- ≒ログ線形モデル＋隠れマルコフモデル
- ログ線形モデル(後述)による構造予測
- ビタビアルゴリズムやパラメータ推定はHMMと同様の動的計画法が使える



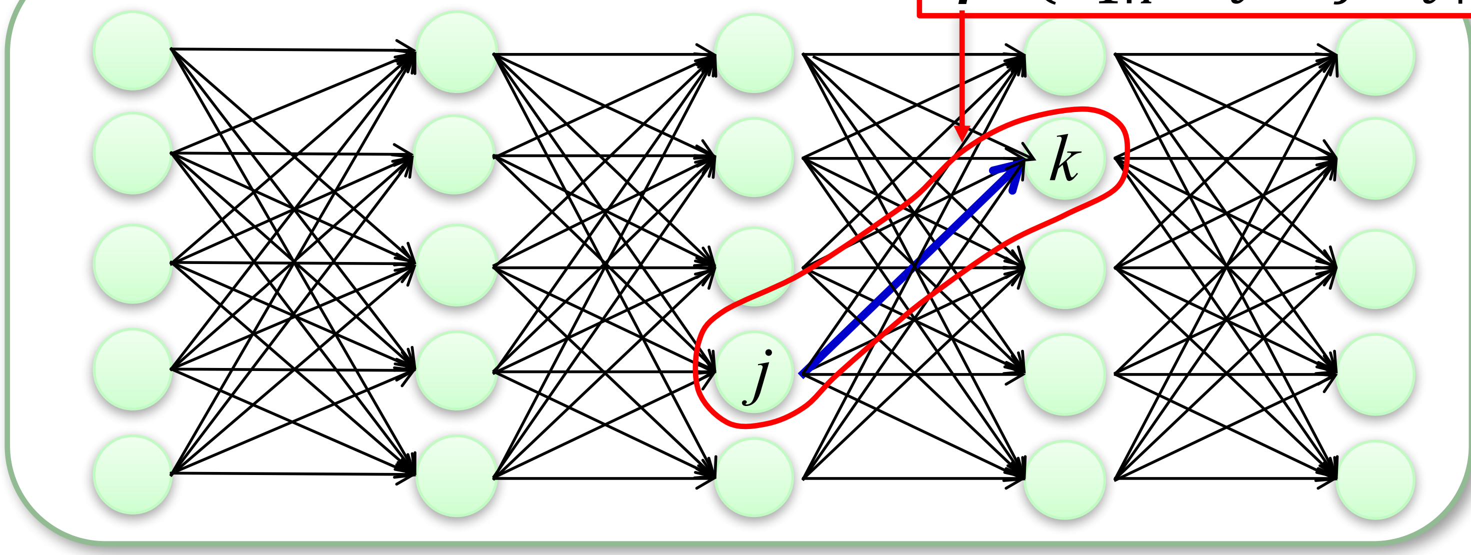
条件付き確率場

- 特徴ベクトルが、状態と遷移の特徴ベクトルに分解できると仮定

$$\boldsymbol{\varphi}(\boldsymbol{o}_{1:T}, \boldsymbol{s}_{1:T}) = \sum_{t=1}^{T} \boldsymbol{\varphi}^a(\boldsymbol{o}_{1:T}, s_t, s_{t+1}) + \boldsymbol{\varphi}^b(\boldsymbol{o}_{1:T}, s_t)$$

$$p(y|x) \propto \exp\left(\sum_{t=1}^{T} \boldsymbol{\varphi}^a(\boldsymbol{o}_{1:T}, s_t, s_{t+1}) + \boldsymbol{\varphi}^b(\boldsymbol{o}_{1:T}, s_t)\right)$$

$$\boxed{\boldsymbol{\varphi}^a(\boldsymbol{o}_{1:T}, s_t = j, s_{t+1} = k)}$$



ビタビアルゴリズム

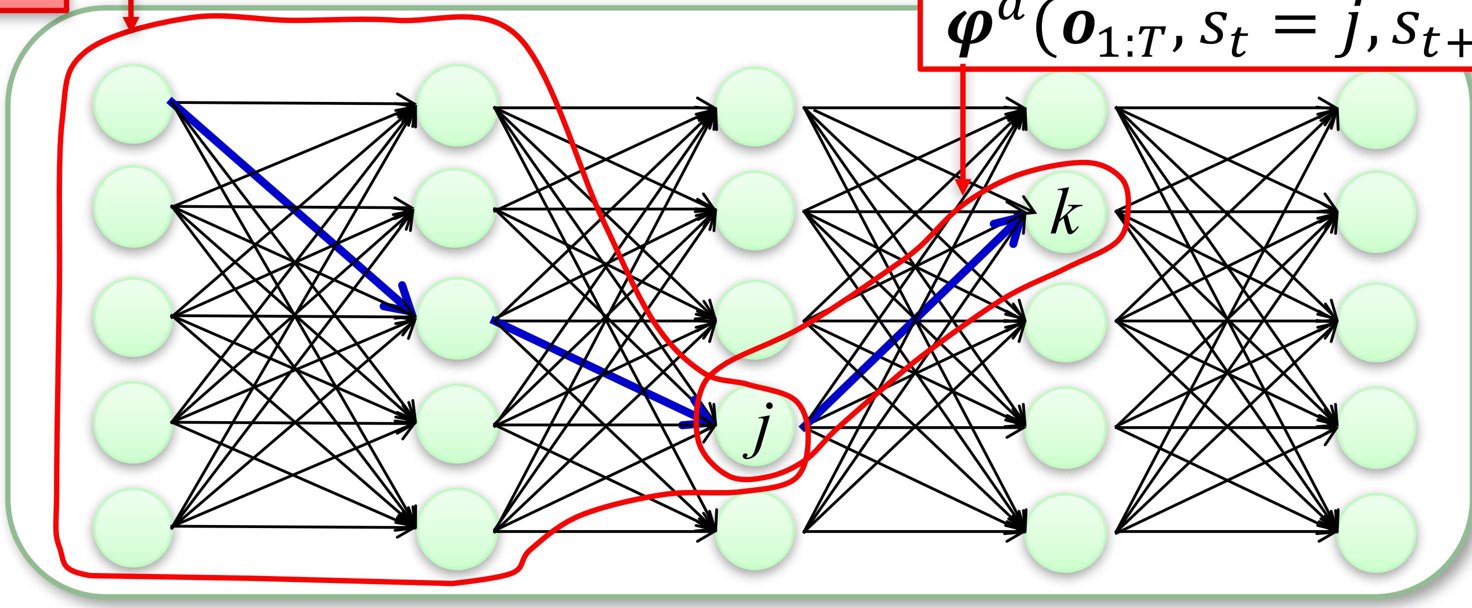
- HMMのビタビアルゴリズムと同じ手法が使える

$$q_{t+1}(k) = \max_{\boldsymbol{s}_{1:t}} \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}(\boldsymbol{o}_{1:T}, s_{t+1} = k, \boldsymbol{s}_{1:t})$$
$$= \max_{j \in S} [q_t(j) + \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}^a(\boldsymbol{o}_{1:T}, s_t = j, s_{t+1} = k)]$$
$$+ \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}^b(\boldsymbol{o}_{1:T}, s_{t+1} = k)$$

$$\boldsymbol{s}^* = \operatorname*{argmax}_{\boldsymbol{s}_{1:T}} \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}(\boldsymbol{o}_{1:T}, \boldsymbol{s}_{1:T}) = \operatorname*{argmax}_{k \in S} q_T(k)$$

$q_t(j)$

$\boldsymbol{\varphi}^a(\boldsymbol{o}_{1:T}, s_t = j, s_{t+1} = k)$



ログ線形モデルの学習

$$L(\boldsymbol{w}|D) = \sum_{i=1}^{N} \left[\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}\left(x^{(i)}, y^{(i)}\right) - \log \sum_{y} \exp\left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}\left(x^{(i)}, y\right)\right) \right]$$

$$\frac{\partial}{\partial w_m} L(\boldsymbol{w}|D)$$

$$= \sum_{i=1}^{N} \varphi_m\left(x^{(i)}, y^{(i)}\right) - \sum_{i=1}^{N} \frac{\sum_y \varphi_m\left(x^{(i)}, y\right) \exp\left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}\left(x^{(i)}, y\right)\right)}{\sum_{y'} \exp\left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}\left(x^{(i)}, y'\right)\right)}$$

$$= \underbrace{\sum_{i=1}^{N} \varphi_m\left(x^{(i)}, y^{(i)}\right)}_{\text{学習データ中の出現回数}} - \underbrace{\sum_{i=1}^{N} \sum_{y} \varphi_m\left(x^{(i)}, y\right) p\left(y|x^{(i)}\right)}_{\text{期待値}}$$

CRFの学習

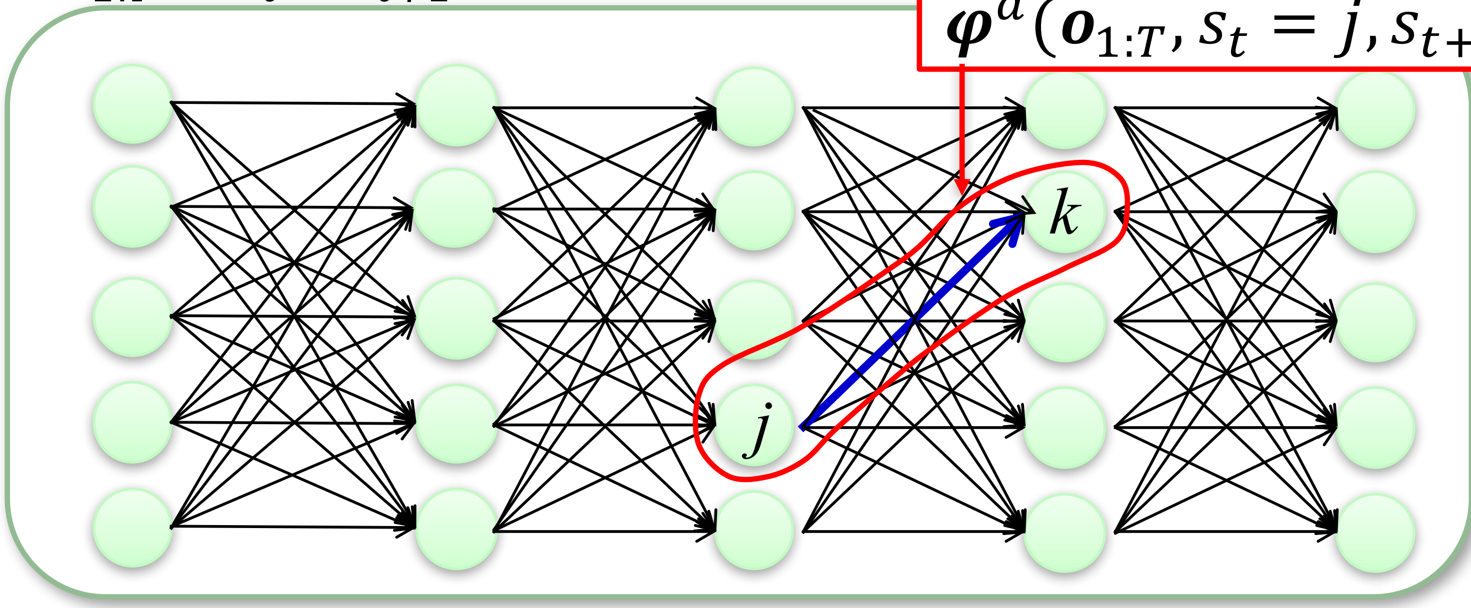
■ 特徴量 $\boldsymbol{\varphi}^a, \boldsymbol{\varphi}^b$ の期待値を計算する

$$E\left[\varphi_m\left(\boldsymbol{o}_{1:T}^{(i)}, \boldsymbol{s}_{1:T}\right)\middle|\boldsymbol{o}_{1:T}^{(i)}\right] = \sum_{\boldsymbol{s}_{1:T}} \varphi_m\left(\boldsymbol{o}_{1:T}^{(i)}, \boldsymbol{s}_{1:T}\right) p\left(\boldsymbol{s}_{1:T}\middle|\boldsymbol{o}_{1:T}^{(i)}\right)$$

$$E\left[\varphi_m^a\left(\boldsymbol{o}_{1:T}^{(i)}, s_t = j, s_{t+1} = k\right)\middle|\boldsymbol{o}_{1:T}^{(i)}\right]$$

$$= \sum_{\boldsymbol{s}_{1:T} \text{ s.t. } s_t = j, s_{t+1} = k} \varphi_m^a\left(\boldsymbol{o}_{1:T}^{(i)}, j, k\right) p\left(\boldsymbol{s}_{1:T}\middle|\boldsymbol{o}_{1:T}^{(i)}\right)$$

$\boldsymbol{\varphi}^a(\boldsymbol{o}_{1:T}, s_t = j, s_{t+1} = k)$

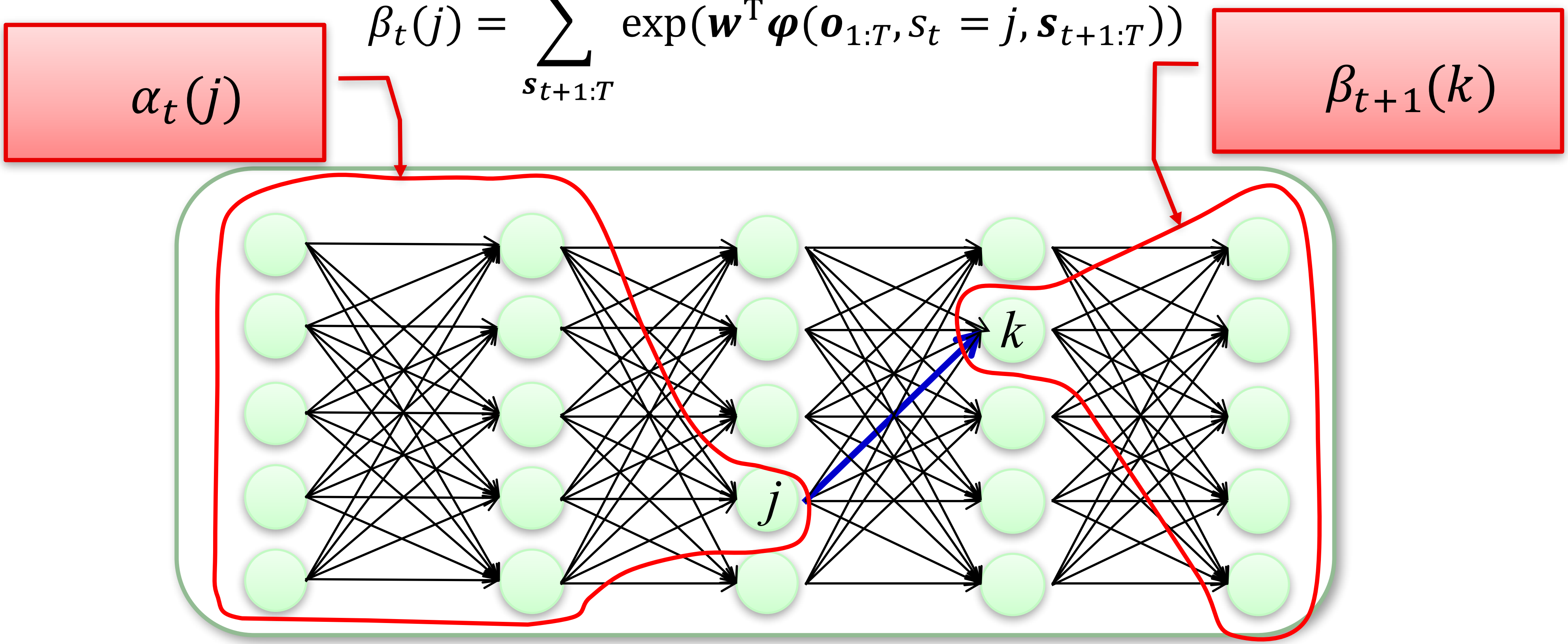


演習3:CRFの学習

- 特徴量 $\boldsymbol{\varphi}^a$, $\boldsymbol{\varphi}^b$ の期待値を計算する方法を考えよ
- HMM と同様に、$\alpha_t(j), \beta_t(j)$ を定める

$$\alpha_t(j) = \sum_{\boldsymbol{s}_{1:t-1}} \exp(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}(\boldsymbol{o}_{1:T}, \boldsymbol{s}_{1:t-1}, s_t = j))$$

$$\beta_t(j) = \sum_{\boldsymbol{s}_{t+1:T}} \exp(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}(\boldsymbol{o}_{1:T}, s_t = j, \boldsymbol{s}_{t+1:T}))$$



CRFの学習

- 期待値は、$\alpha_t(j), \beta_t(j)$ を使って計算できる

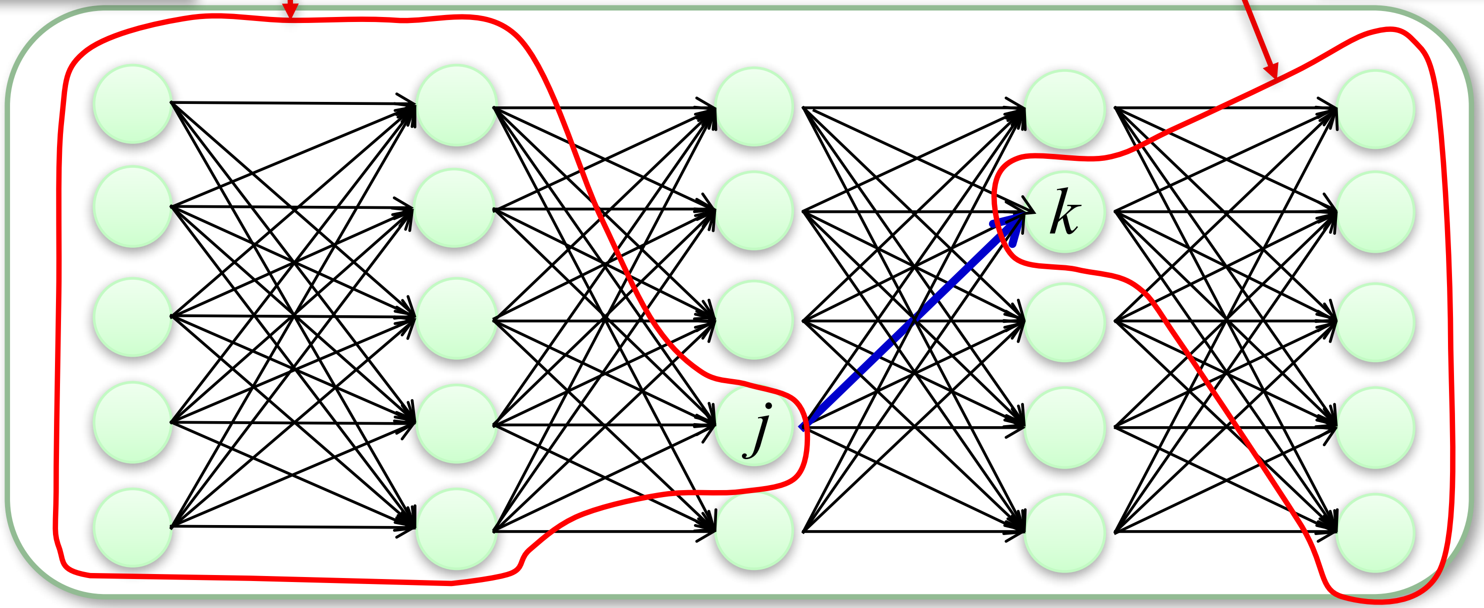
$$E\left[\varphi_m^a\left(\boldsymbol{o}_{1:T}^{(i)}, s_t=j, s_{t+1}=k\right)\middle|\boldsymbol{o}_{1:T}^{(i)}\right]$$

$$=\sum_{\boldsymbol{s}_{1:T}\ \text{s.t.}\ s_t=j, s_{t+1}=k} \varphi_m^a\left(\boldsymbol{o}_{1:T}^{(i)}, j, k\right) p\left(\boldsymbol{s}_{1:T}\middle|\boldsymbol{o}_{1:T}^{(i)}\right)$$

$$=\varphi_m^a\left(\boldsymbol{o}_{1:T}^{(i)}, j, k\right)\frac{1}{Z(\boldsymbol{o}_{1:T})}\alpha_t(j)\beta_{t+1}(k)\exp\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}^a\left(\boldsymbol{o}_{1:T}, j, k\right)\right)$$

$\alpha_t(j)$

$\beta_{t+1}(k)$



$\alpha_t(j)$ の計算

- 前向き確率の計算と同じ

$$\alpha_{t+1}(k) = \sum_{\boldsymbol{s}_{1:t}} \exp(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}(\boldsymbol{o}_{1:T}, \boldsymbol{s}_{1:t}, s_{t+1} = k))$$

$$= \sum_{j \in S} \sum_{\boldsymbol{s}_{1:t-1}} \exp(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}(\boldsymbol{o}_{1:T}, \boldsymbol{s}_{1:t-1}, s_t = j) + \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}^a(\boldsymbol{o}_{1:T}, j, k) + \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}^b(\boldsymbol{o}_{1:T}, k))$$

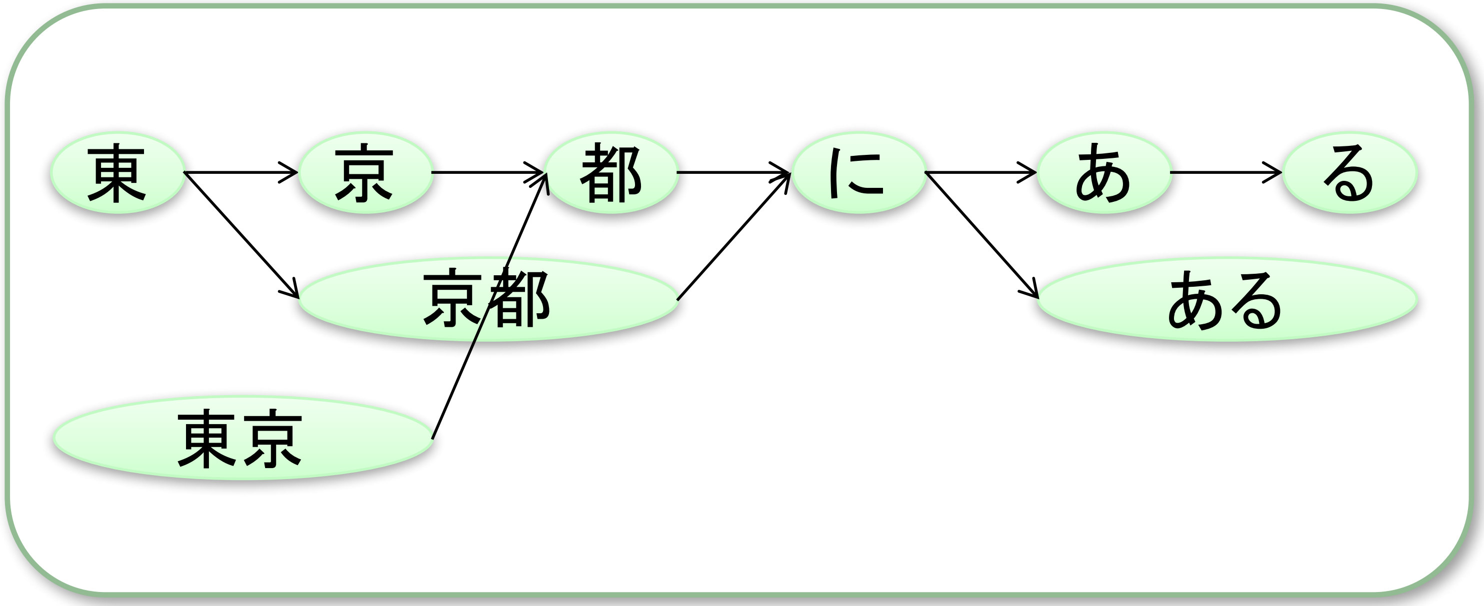
$$= \sum_{j \in S} \sum_{\boldsymbol{s}_{1:t-1}} \exp(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}(\boldsymbol{o}_{1:T}, \boldsymbol{s}_{1:t-1}, s_t = j)) \exp(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}^a(\boldsymbol{o}_{1:T}, j, k)) \exp(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}^b(\boldsymbol{o}_{1:T}, k))$$

$$= \exp\left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}^b(\boldsymbol{o}_{1:T}, k)\right) \sum_{j \in S} \exp\left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}^a(\boldsymbol{o}_{1:T}, j, k)\right) \sum_{\boldsymbol{s}_{1:t-1}} \exp\left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}(\boldsymbol{o}_{1:T}, \boldsymbol{s}_{1:t-1}, s_t = j)\right)$$

$$= \exp\left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}^b(\boldsymbol{o}_{1:T}, k)\right) \sum_{j \in S} \exp\left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}^a(\boldsymbol{o}_{1:T}, j, k)\right) \alpha_t(j)$$

構造予測いろいろ

- 最大マージンマルコフネットワーク
  - CRFと同じ構造＋SVM
- 構造化パーセプトロン
  - CRFと同じ構造＋パーセプトロン
- セミマルコフCRF
  - ラティス構造＋ログ線形モデル



機械学習＋離散構造

- あるデータ構造において、目的関数とその勾配が効率的に計算できれば、同様の手法が適用可
  - 特徴の期待値 → ログ線形モデル
  - マージン → サポートベクトルマシン
  - argmax → パーセプトロン
- 計算方法は一つではない。厳密解を求めなくても学習できる場合も多い
  - 期待値 → モンテカルロ法、変分法、etc.
  - argmax → 探索（最良優先探索、A*探索、etc.）、最小全域木、最小カット、線形計画法、etc.

まとめ

- HMM のパラメータ推定
  - 最尤推定
  - Baum-Welch アルゴリズム
  - 前向き・後向き確率 → ビタビアルゴリズムと同様の動的計画法で計算できる
- 条件付き確率場 (CRF)
  - HMM と同様に、動的計画法が適用できる
- 機械学習と離散構造