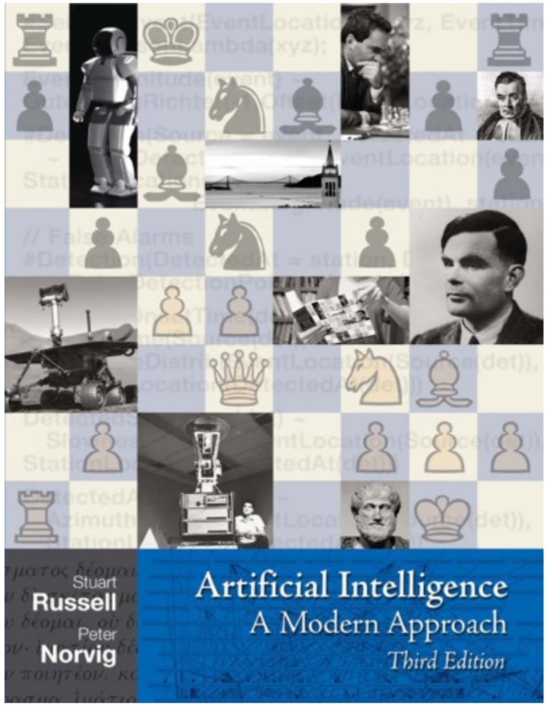
自然言語処理(1)
構造予測1

宮尾 祐介

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

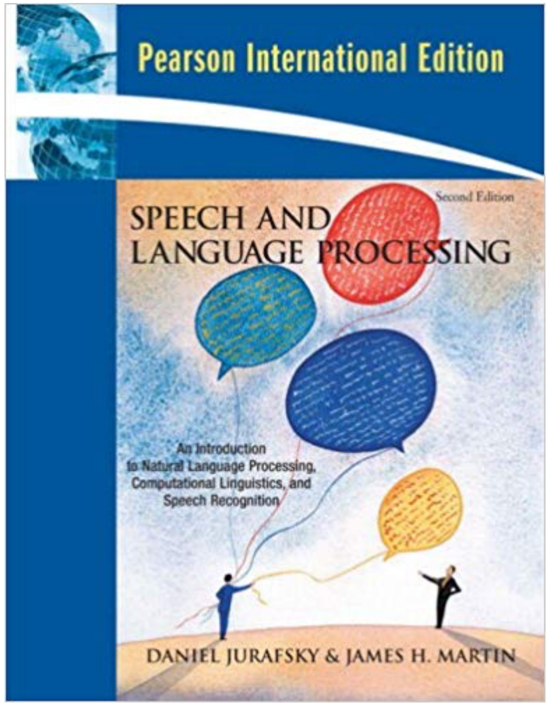
https://mynlp.github.io/

参考書



Russell, Norvig. Artificial Intelligence: A Modern Approach (Third Edition). Prentice Hall. 2009

ラッセル, ノーヴィグ. エージェントアプローチ 人工知能 第2版. 共立出版. 2008年



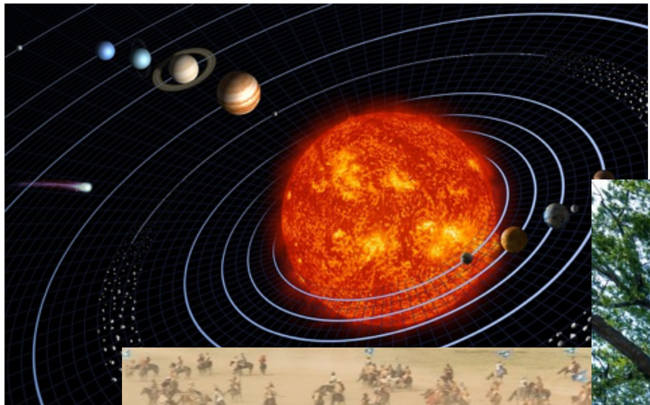
Jurafsky, Martin. Speech and Language Processing International Version. Pearson. 2008

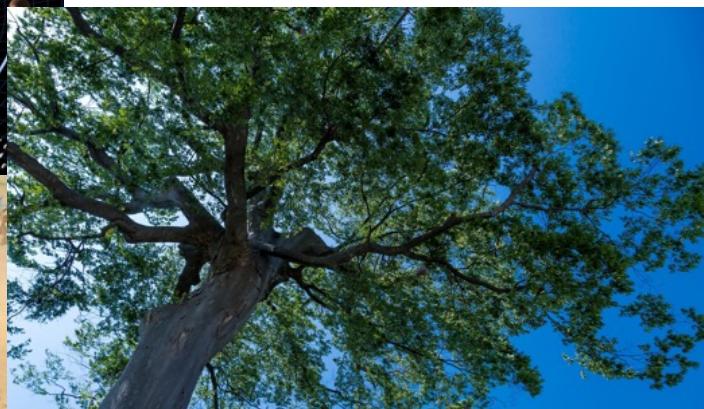
黒橋. 自然言語処理. 放送大学教育振興会. 2015

※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

Computational Science

- 森羅万象を「計算」であきらかにする
  - 宇宙、気象、物質、生命、経済、社会、歴史、心理、脳、etc.
- 計算科学のあらゆる理論・技術を使う
  - 確率・統計、機械学習、離散数学、形式言語、形式論理、数値計算、etc.





Computational Science の方法論

- 森羅万象を「計算」であきらかにする
  - 宇宙、気象、物質、生命、経済、社会、歴史、心理、脳、etc.
- 計算科学のあらゆる理論・技術を使う
  - 確率・統計、機械学習、離散数学、形式言語、形式論理、数値計算、etc.

情報科学

離散数学
形式言語
形式論理
計算モデル
機械学習

→

問題
↓
数理モデル
↓
観測データ

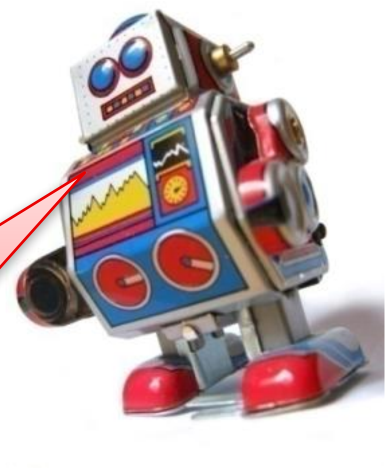
経済について
解明したい！

知能の Computational Science

- 知能:ヒトの情報処理・認知能力
- 人類が解明したいモノの一つ
  - 学術的興味
  - 人間の代わりに働くロボット
- どうやって？
- 何が違う？
  - 不確実性 → 確率・統計、機械学習
  - 離散構造

あれなに？

最近できたカレーのお店です。
激辛だけど評判いいですよ。



知能システムと離散構造

- ヒトが関わる多くのモノやできごとは、離散構造
  - リスト(系列構造)：自然言語(単語列)、購買履歴、楽譜、株価、遺伝子
  - 木構造：自然言語(構文木)、プログラミング言語、XML、証明木、データベースクエリ
  - グラフ：自然言語(意味構造)、インターネット、ソーシャルネットワーク、ウェブページ、遺伝子発現パスウェイ
- これらに適用できる数理モデルが必要

自然言語処理・計算言語学

- 人間の「ことばを操る能力」を科学する（言語学）
- 「ことばを操る能力」＝「計算」≒「知能」
  - 入力：自然言語 or 知覚など
  - 出力：行動 or 自然言語
- ことばを観測して、知能のしくみを探る

入力

自然言語処理ではどういう応用がありますか？

計算 ≒ 知能

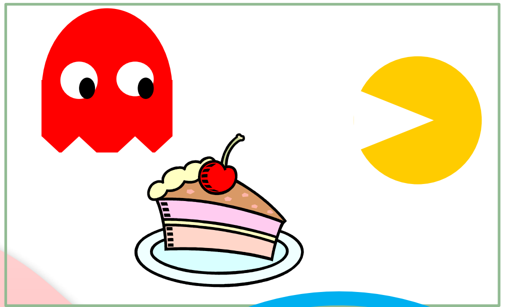
出力

昔はかな漢字変換，最近は自動翻訳や対話システムがあります．

自然言語処理・計算言語学

この講義の範囲

意味構造



意味解析
文脈解析

構文構造

友達 から ケーキ を もらった

構文解析

単語・品詞列

友達 から ケーキ を もらった
名詞 助詞 名詞 助詞 動詞

形態素解析・品詞解析

入力：文字列 友達からケーキをもらった

離散数学
形式言語
形式論理
計算モデル
機械学習

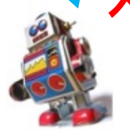
出力：文字列，etc.

機械翻訳

A friend gave me a cake.

検索，情報抽出，
質問応答

対話システム



紅茶を入れ
ましょうか

自然言語

- 社会において自然に発生して用いられている言語
- 主に人間同士で意志疎通を行うことを目的として、人間が日常、読み、書き、聞き、喋り、思考するための言語
- 例：日本語、英語、ドイツ語、フランス語、…

⇔人工言語

  - エスペラントなど人工的につくられた言語
  - 形式言語（記号論理学やコンピュータープログラミングのためにつくられた記号や式によって作られた人工言語）

[デジタル大辞泉]

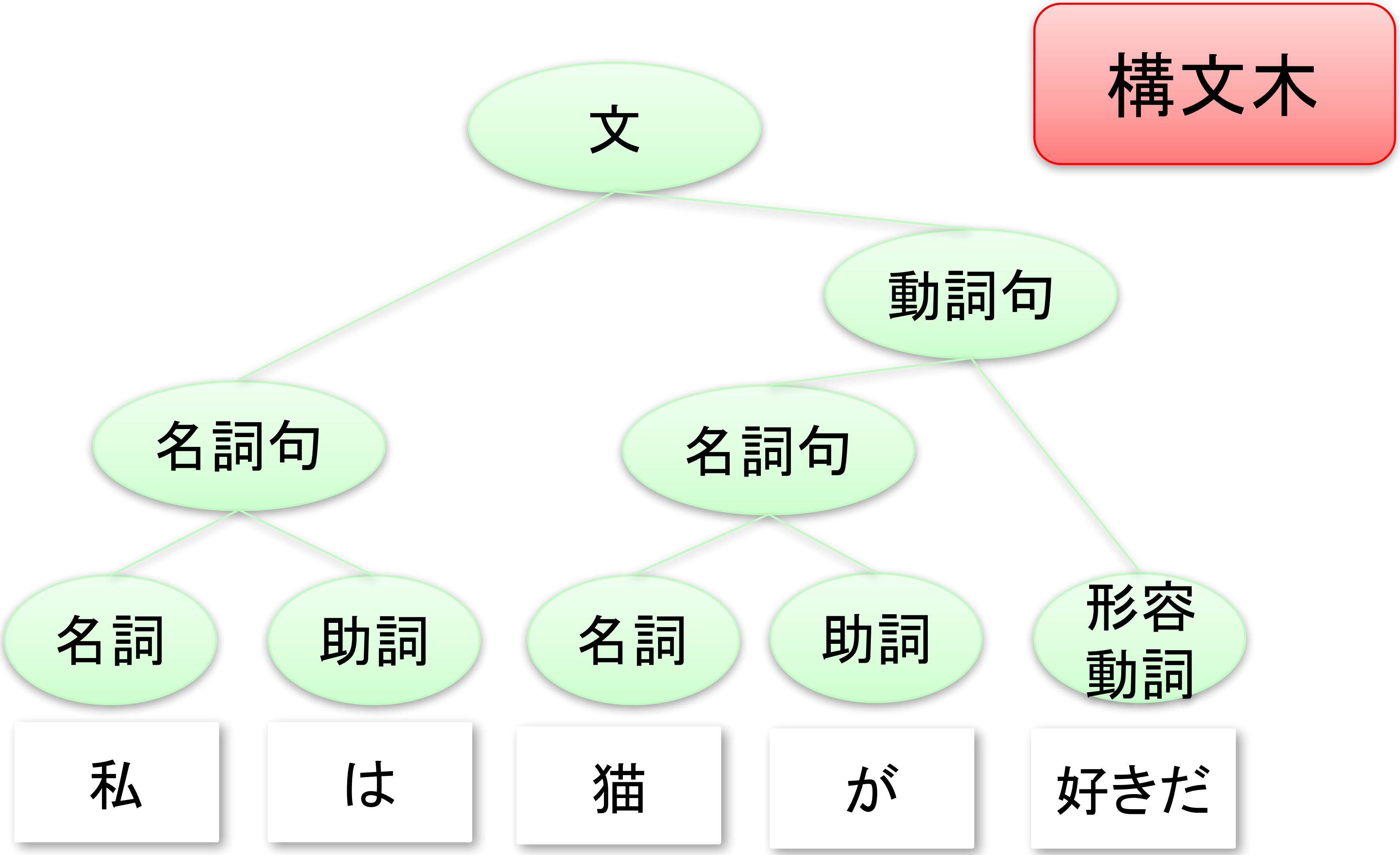
形態素解析

私は猫が好きだ

単語・品詞列

名詞	助詞	名詞	助詞	形容動詞
私	は	猫	が	好きだ

構文解析



構造学習・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、…
  - $\boldsymbol{y} = f(\boldsymbol{x}) = \mathrm{argmax}_{\boldsymbol{y}'}\, g(\boldsymbol{x}, \boldsymbol{y}')$

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

文

動詞句

名詞句 名詞句

私は ケーキが 好きです

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。

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: $x_1, x_2, \ldots, x_n$

⬇

出力: $y_1, y_2, \ldots, y_n$

形態素解析
＝単語分割＋系列ラベリング

私は猫が好きだ

単語分割（分かち書き）

私 | は | 猫 | が | 好きだ

系列ラベリング
（品詞タグ付け）

名詞	助詞	名詞	助詞	形容動詞
私	は	猫	が	好きだ

単語分割も系列ラベリング

東京都にある

入力:

東	京	都	に	あ	る

出力:

	0	1	1	1	0
東	京	都	に	あ	る

東京 | 都 | に | ある

何が難しい？

どれが正しい？
（人間の解釈に合う？）

東京　都　に　ある

東　京都　に　ある

すもももももももものうち

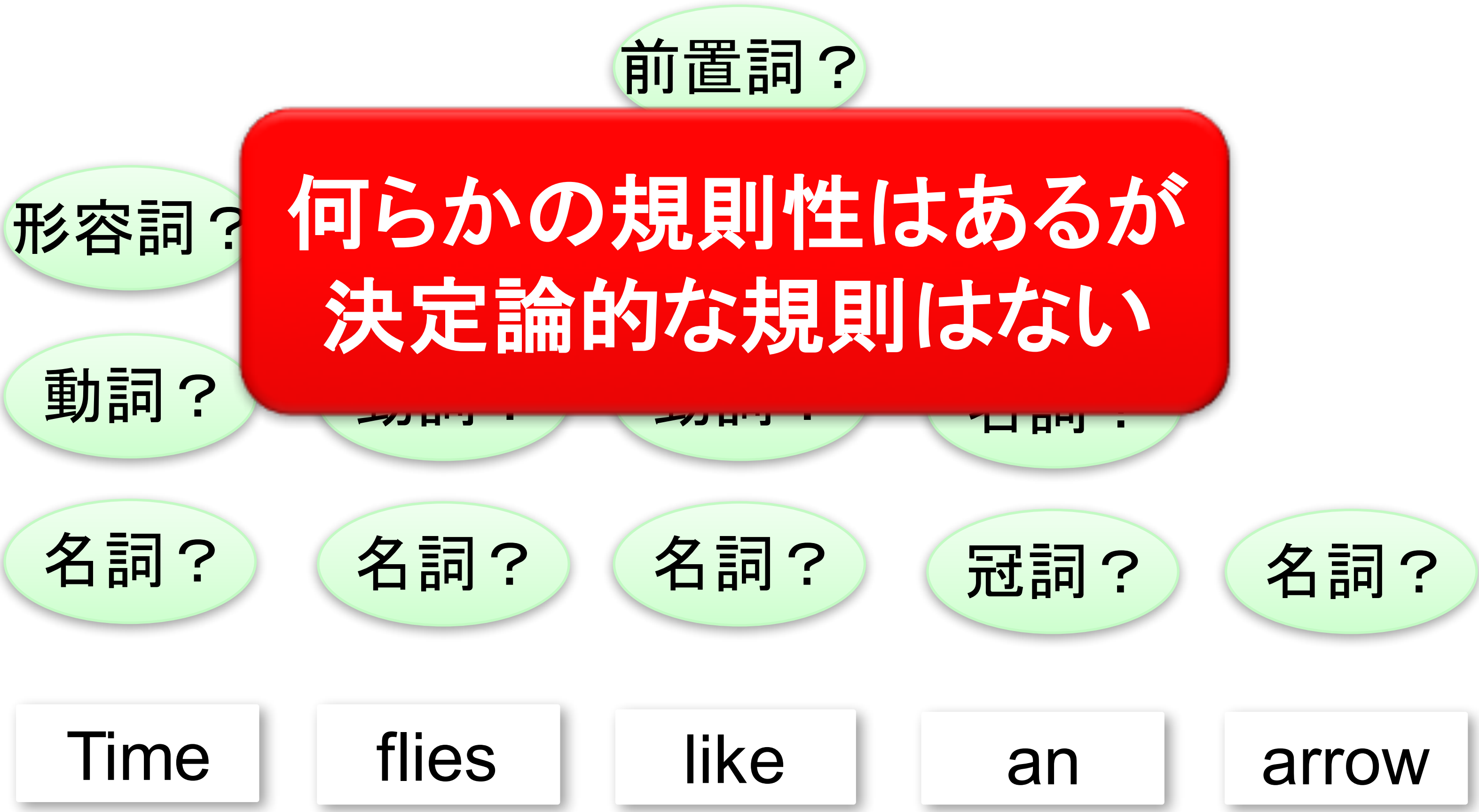
外国人参政権

- 辞書を作ればよいのでは？
  → 辞書にある単語をどう組み合わせるか
  にあいまい性（不確実性）がある

何が難しい？

どれが正しい？
（人間の解釈に合う？）

an はほとんど冠詞
like は動詞か前置詞が多い
名詞の後は名詞か動詞が多い
…



何が難しい？

どれが正しい？
（人間の解釈に合う？） → 確率が高いのはどれ？

確率論的に人間の知能を再現できる数理モデルを探す

形容詞？ 形容詞

動詞？ 動詞？ 動詞？ 名詞？

名詞？ 名詞？ 名詞？ 冠詞？ 名詞？

Time flies like an arrow

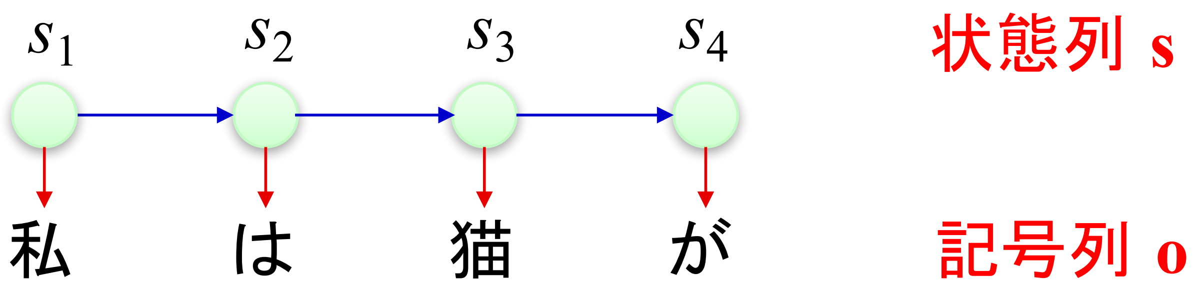
データ主導の方法論

どれが正しい？ → 確率が高いのはどっち？

問題	形態素解析 （単語分割＋品詞タグ付け）
↓	
数理モデル	隠れマルコフモデル, 条件付き確率, etc.
↕	
観測データ	品詞タグ付けコーパス

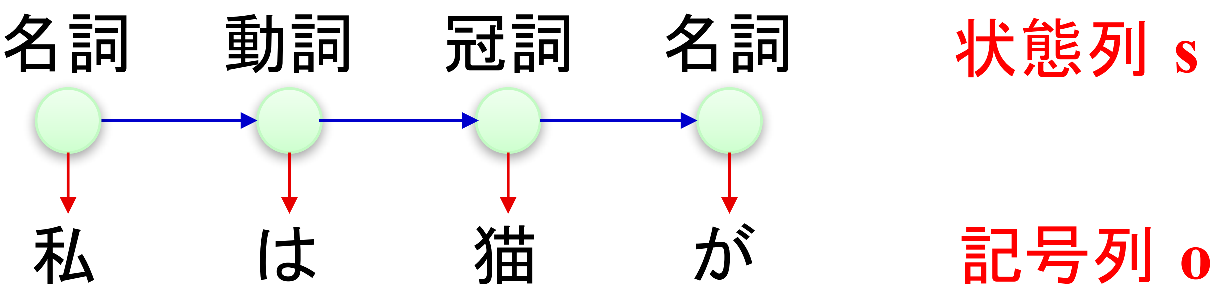
隠れマルコフモデル (Hidden Markov Model; HMM)

- 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 : $\mathbf{s} = s_1, \ldots, s_n$
  - 記号列 : $\mathbf{o} = o_1, \ldots, o_n$



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：$\mathbf{s} = s_1, \ldots, s_n$
  - 記号列：$\mathbf{o} = o_1, \ldots, o_n$



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., $\{A, B\}$

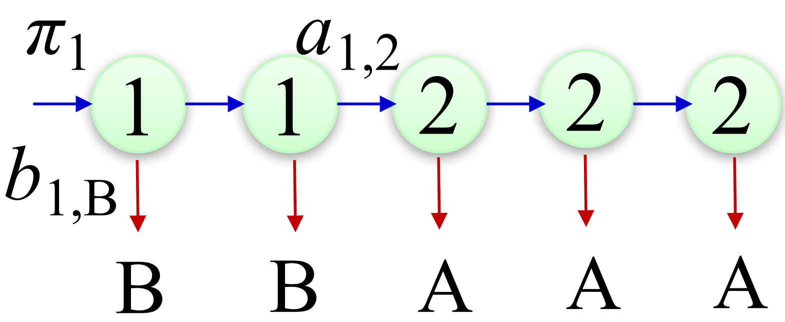
π_k: 文頭が状態 k になる確率

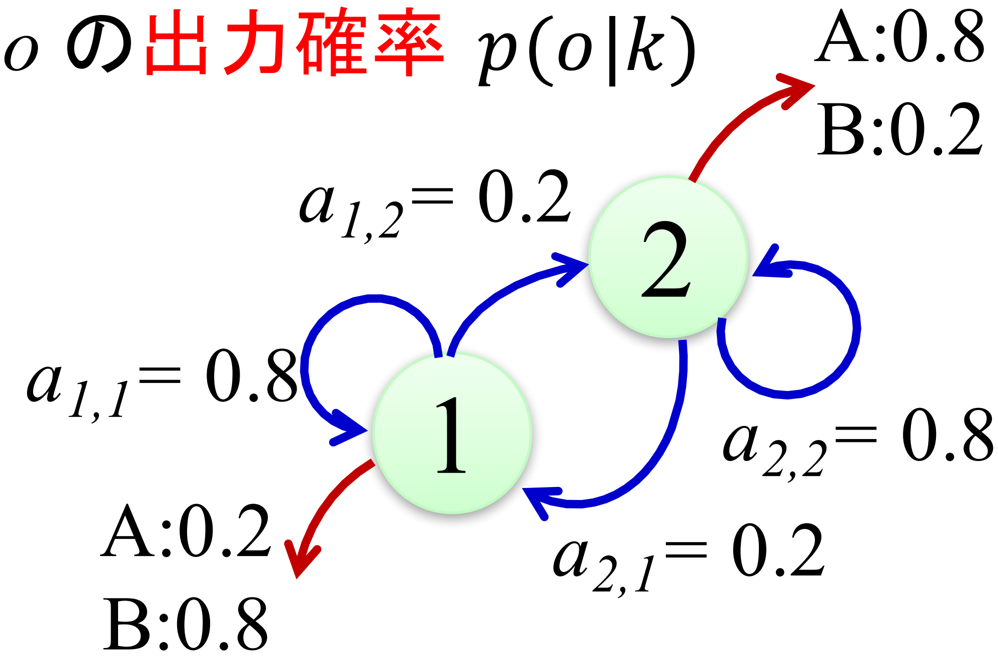
$a_{t,k}$: 状態 t から状態 k への遷移確率 $p(k|t)$

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

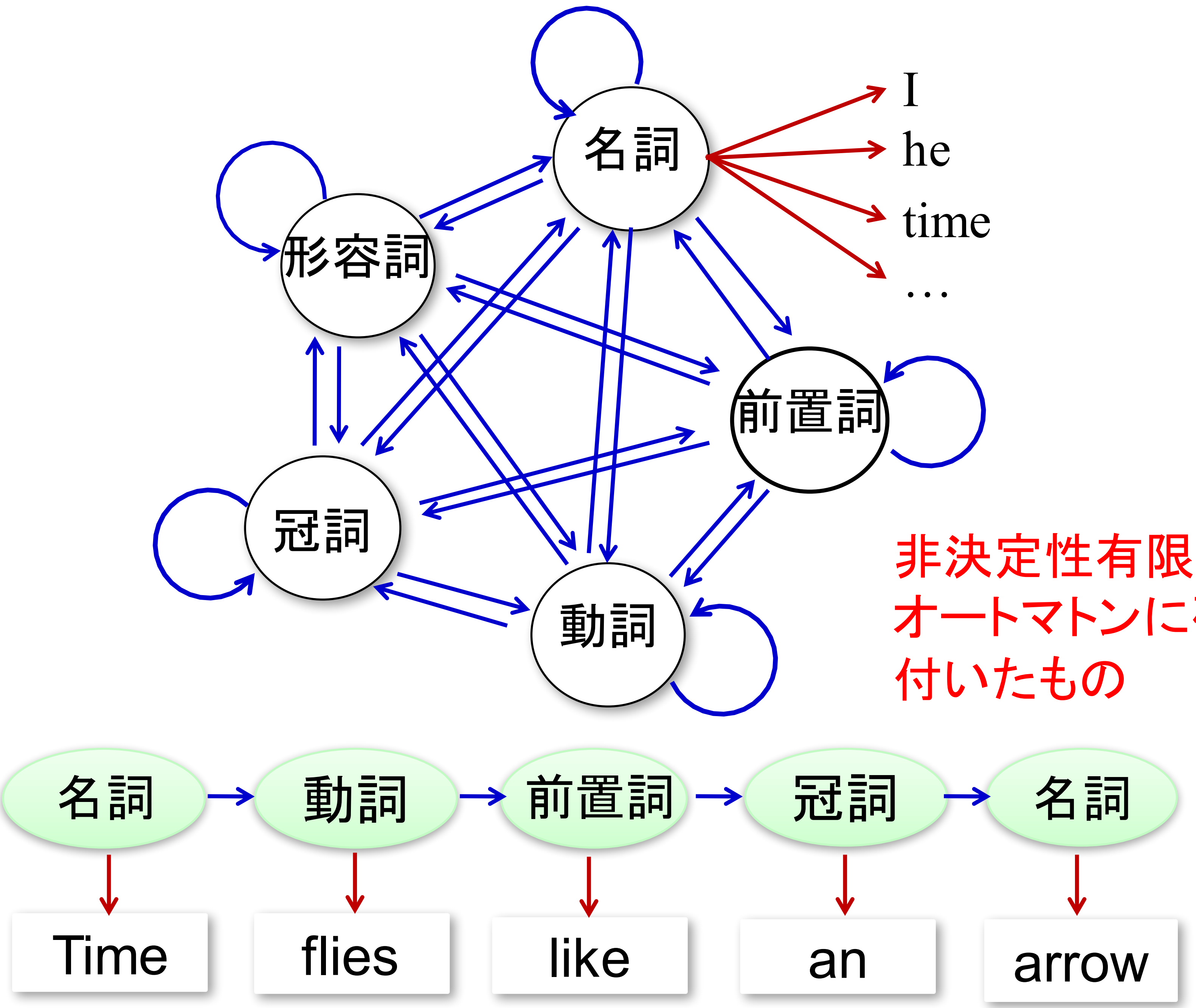
$b_{k,o}$: 状態 k における記号 o の出力確率 $p(o|k)$

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



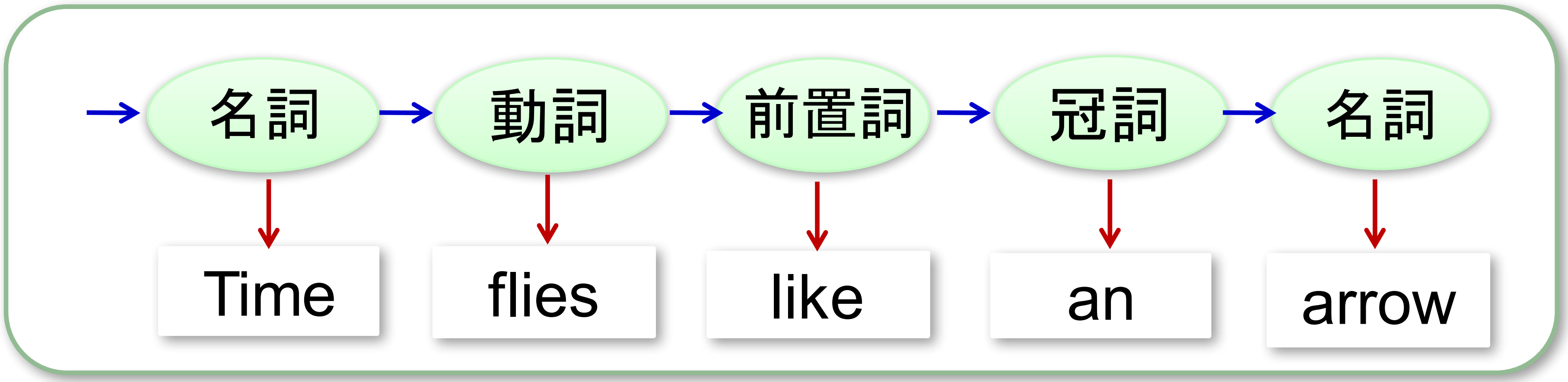


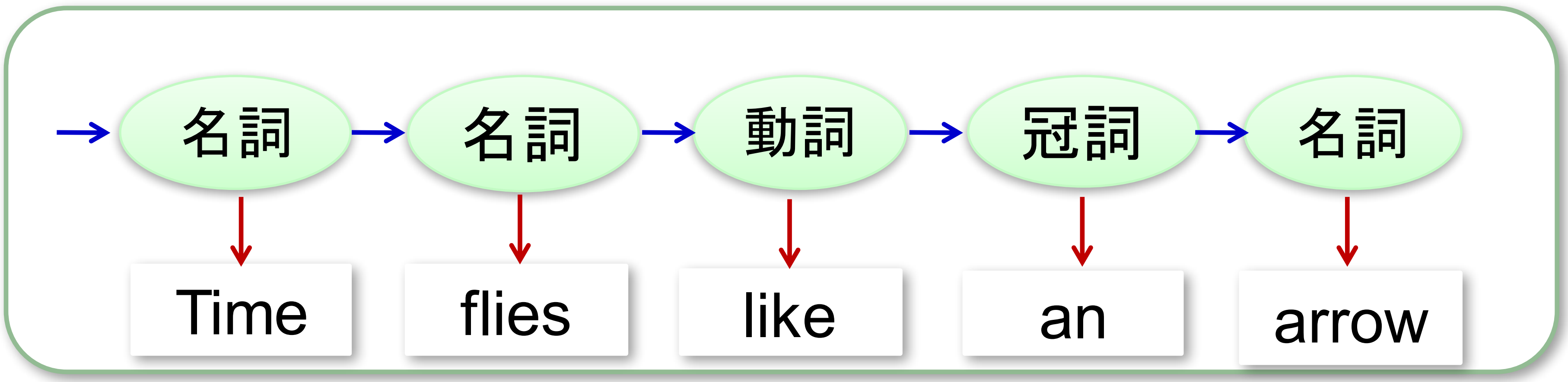
隠れマルコフモデル



非決定性有限状態オートマトンに確率が付いたもの

どっちが正しい？


名詞
動詞
前置詞
冠詞
名詞
Time
flies
like
an
arrow


名詞
名詞
動詞
冠詞
名詞
Time
flies
like
an
arrow

π

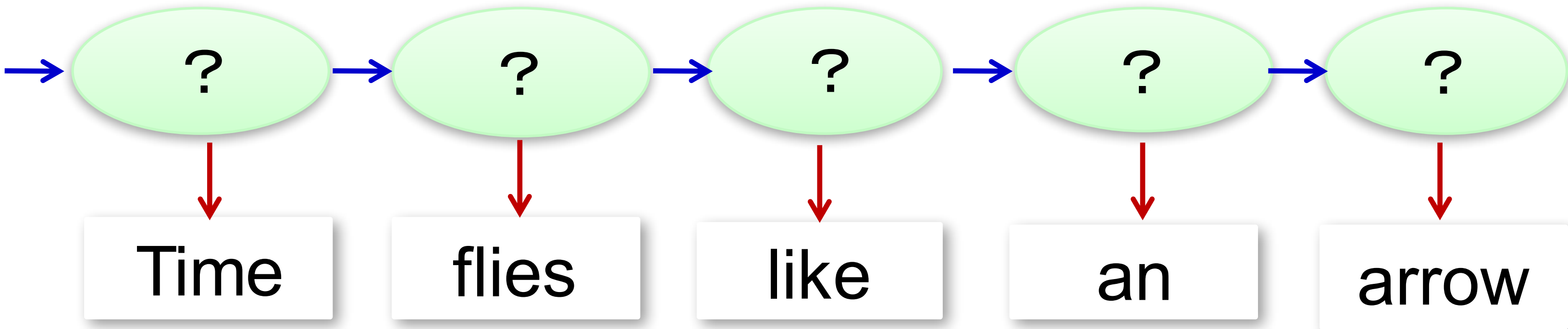
名詞	動詞	形容詞	冠詞	前置詞
0.6	0.0	0.0	0.4	0.0

a

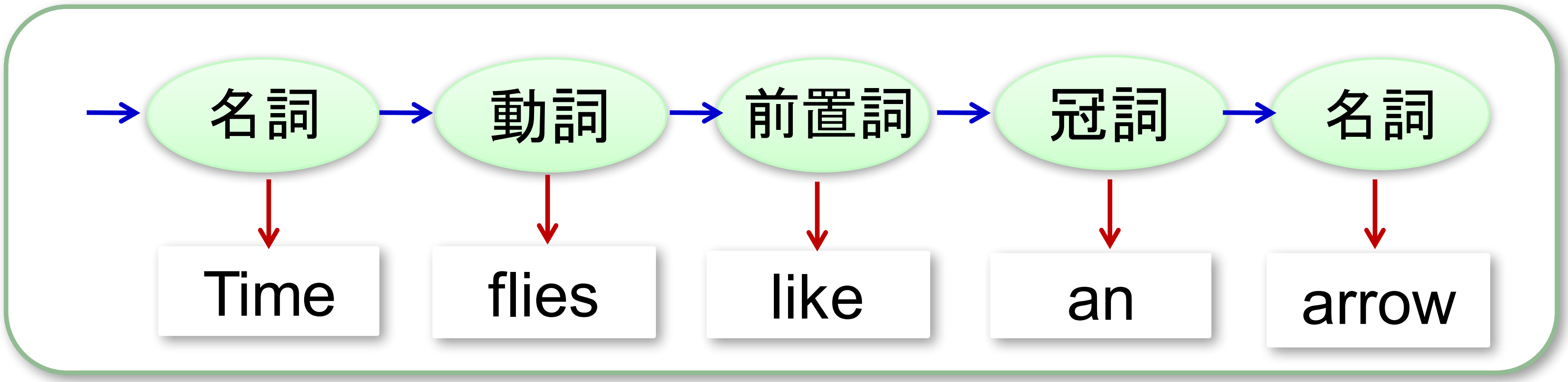
遷移元＼遷移先	名詞	動詞	形容詞	冠詞	前置詞
名詞	0.3	0.4	0.1	0.0	0.2
動詞	0.1	0.0	0.5	0.2	0.2
形容詞	0.5	0.0	0.4	0.1	0.0
冠詞	0.7	0.0	0.0	0.0	0.3
前置詞	0.6	0.0	0.1	0.0	0.3

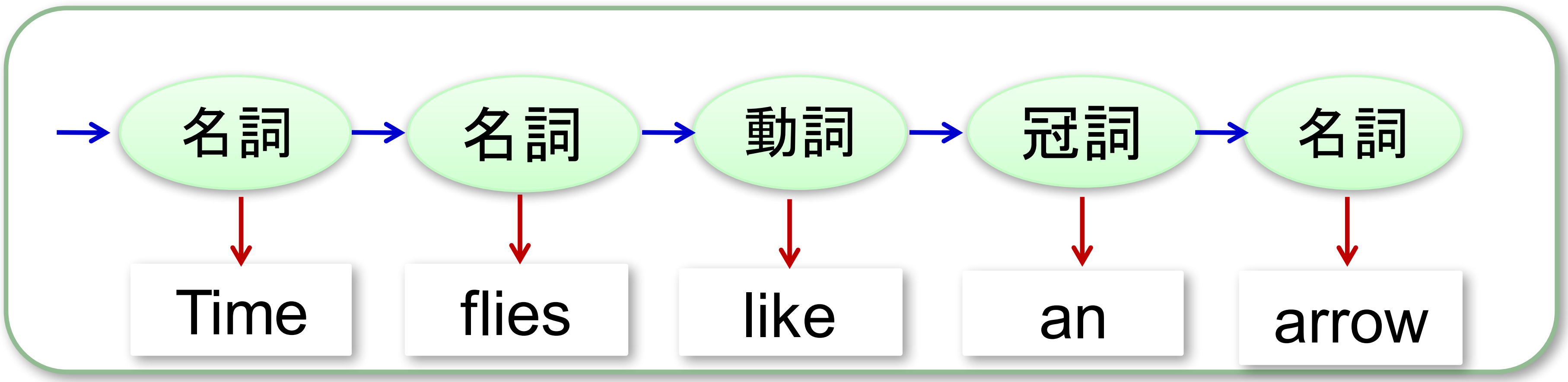
b

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	…

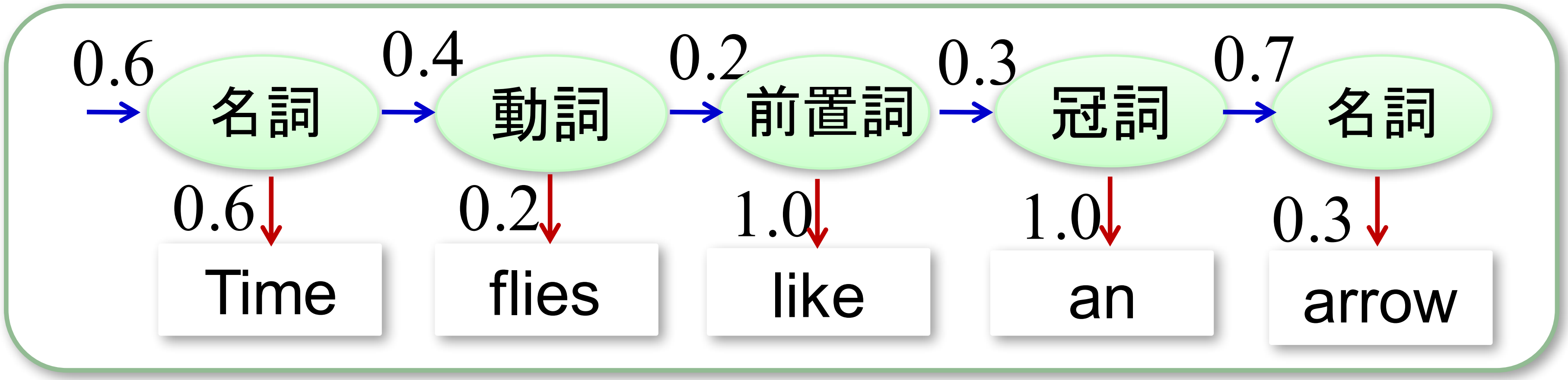


演習1: 確率が高いのはどっち？





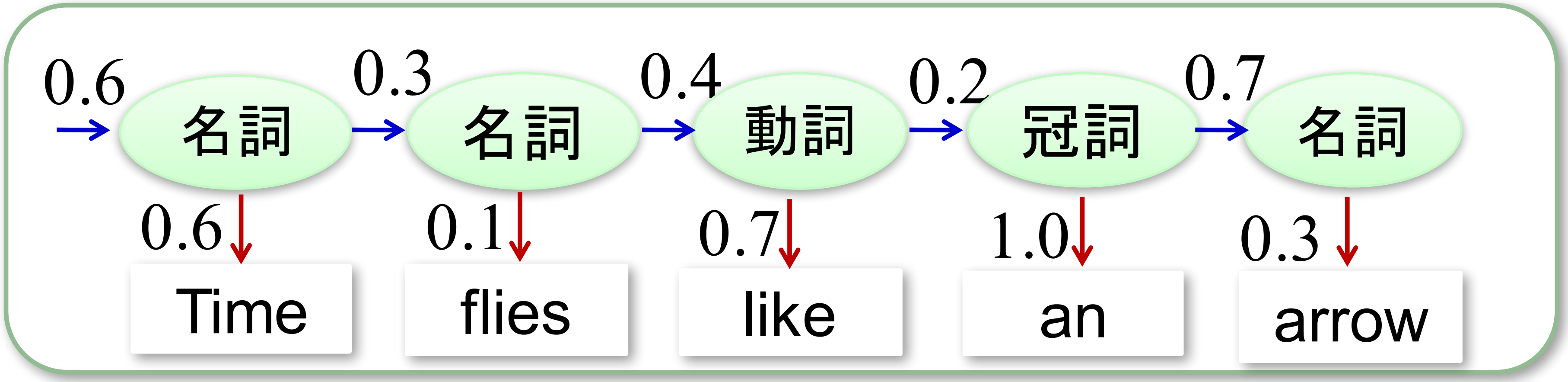
演習1: 確率が高いのはどっち？



p(名詞) × p(time|名詞) × p(動詞|名詞) × p(files|動詞) × p(前置詞|動詞) × p(like|前置詞) × p(冠詞|前置詞) × p(an|冠詞) × p(名詞|冠詞) × p(arrow|名詞)

$$=0.6\times 0.6\times 0.4\times 0.2\times 0.2\times 1.0\times 0.3\times 1.0\times 0.7\times 0.3$$

$$=0.0003628$$



$$=0.0001270$$

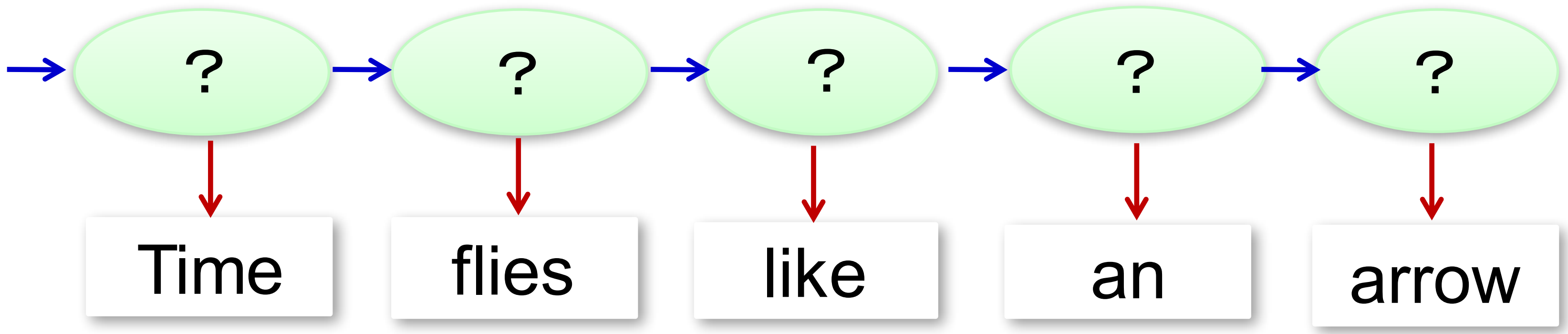
HMMによる品詞タグ付け

- 学習・評価データを用意する
  - 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
  - あとで扱う
- 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
  - 以下、ここから説明する

HMMにおける品詞タグ推定問題

問題:π, a, b が与えられたとき、(生成)確率が最も高くなる品詞タグを推定せよ

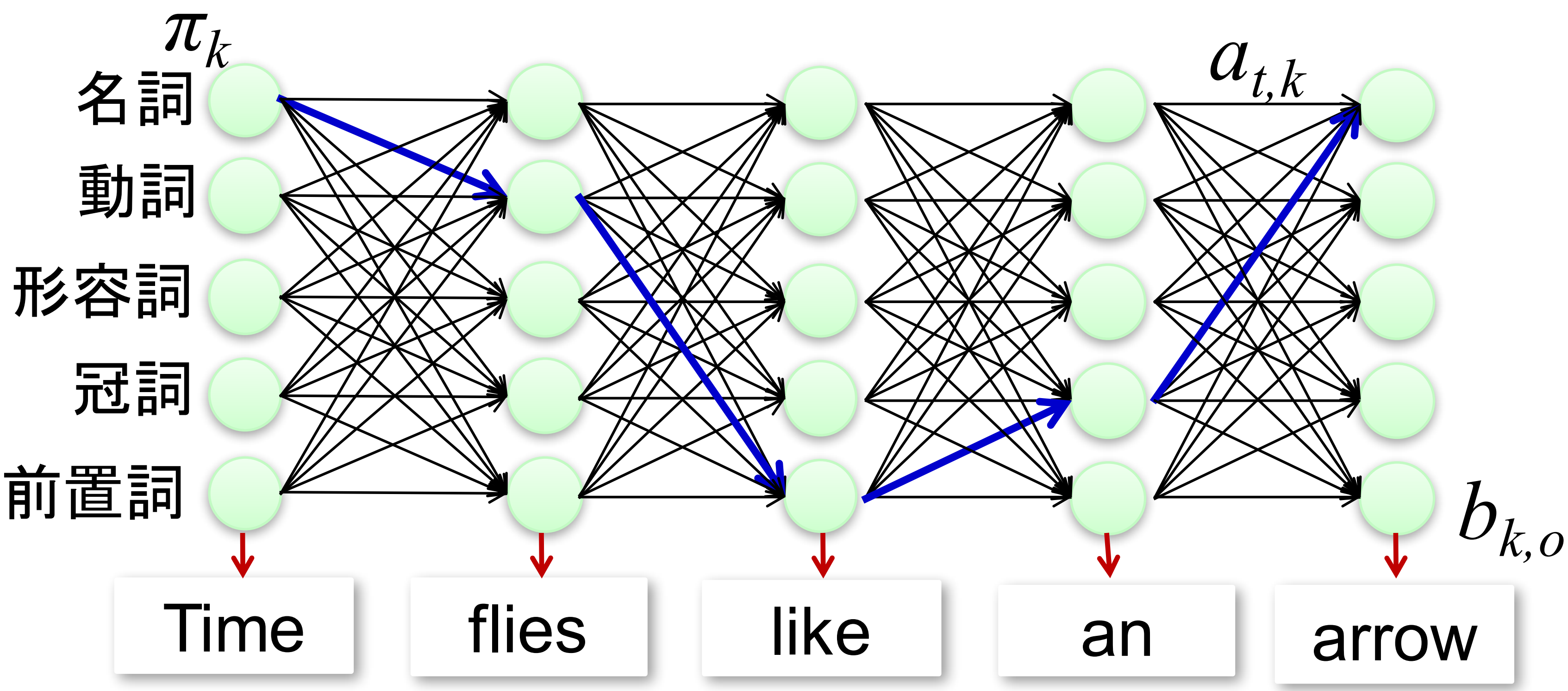
- 記号列 **o**: 既知
- 状態列 **s**: 未知
- パラメータ π, a, b: 既知



トレリス

問題：π, a, b が与えられたとき、（生成）確率が最も高くなる品詞タグを推定せよ

- 横軸：単語列、縦軸：とりうる状態
- トレリス上のパス＝状態列
- 問題＝5^5通りの中から1つのパスを見つける



演習2:トレリス構造における探索問題

- アルゴリズムを設計しその計算量について答えよ
- 確率の最も高いタグ付けを求めよ

演習後解説

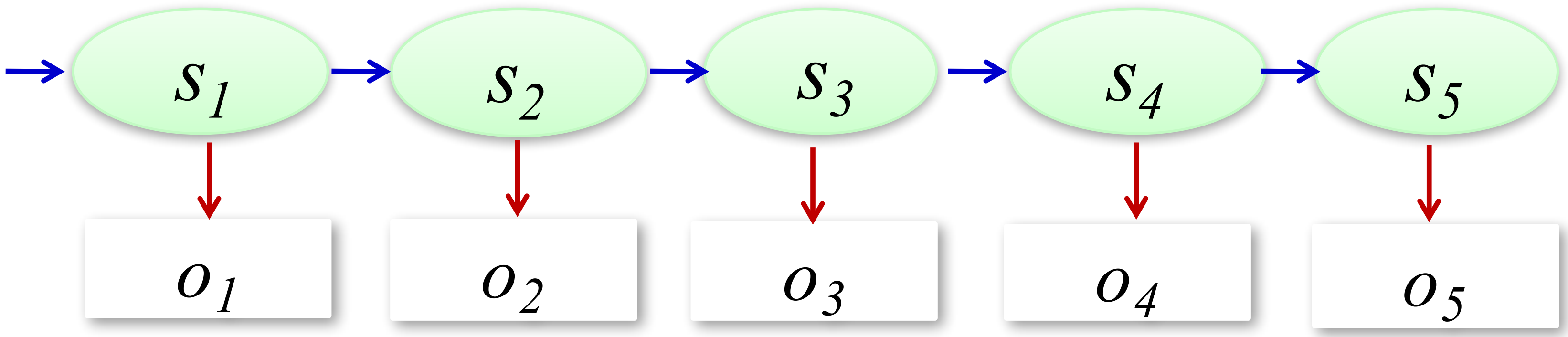
HMMによる系列ラベリング

記号列 $o_{1:T}$ と状態列 $s_{1:T}$ の同時確率分布は

単語　品詞

$$p(o_{1:\mathrm{T}}, s_{1:\mathrm{T}}) = p(s_1)\prod_{t=1}^{T} p(s_{t+1}|s_t)p(o_t|s_t) = \pi_1 \prod_{t=1}^{T} a_{s_t, s_{t+1}} b_{s_t, o_t}$$

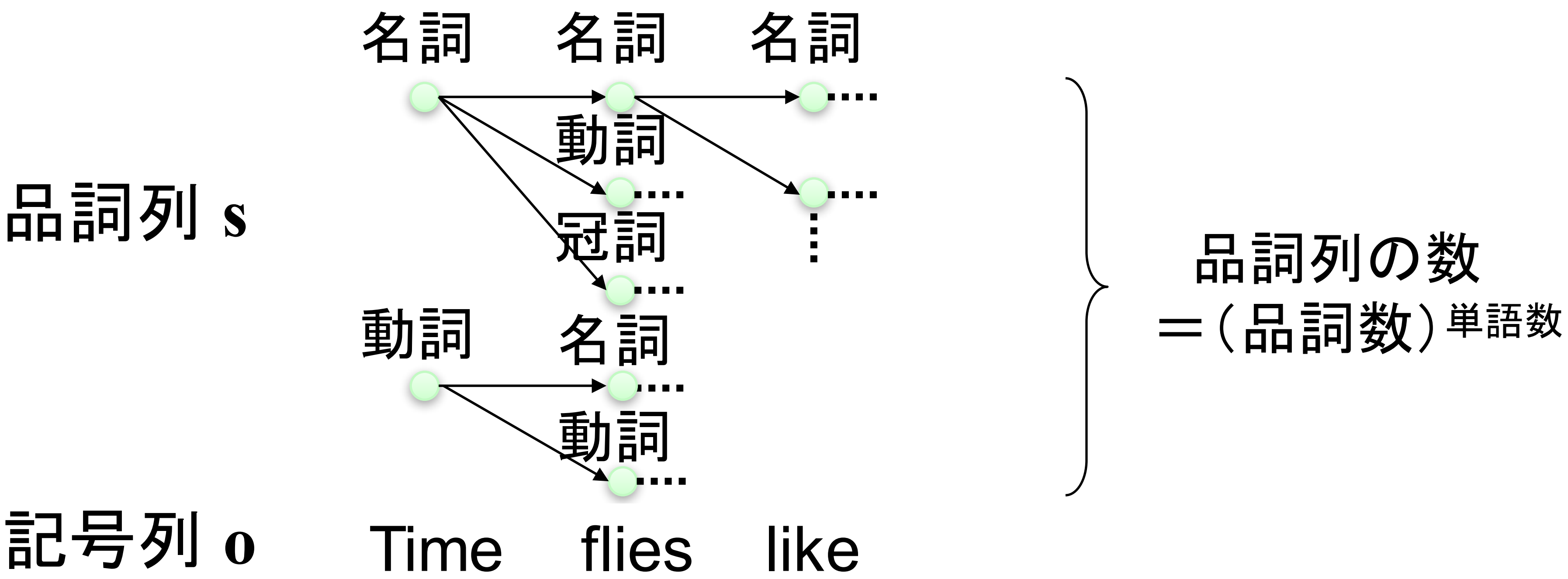
解きたい題： $\underset{s_{1:T}}{\mathrm{argmax}}\, p(o_{1:T}, s_{1:T})$



p(名詞) × p(time|名詞) × p(動詞|名詞) × p(files|動詞) × p(前置詞|動詞) × p(like|前置詞) × p(冠詞|前置詞) × p(an|冠詞) × p(名詞|冠詞) × p(arrow|名詞)

素朴な方法

- 状態列を列挙し、各状態列の確率を計算する
- 長さ T の系列に対して状態数 K のとき $O(K^T)$ の計算量が必要



- 品詞が10個で20単語だと 10^{20} = 1垓

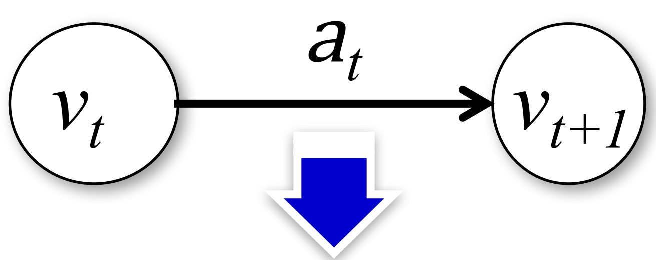
動的計画法(Dynamic Programing)

多段階意思決定問題に対する典型的な最適化手法

仮定

- $t+1$ 段階の状態 v_{t+1} は、t 段階の状態 v_t とその時の行動(選択) a_t によってのみ一意に決定される
- T 段階全体での評価値は、各段階の評価値の和として与えられる:

$$f(v_1, a_1) + \cdots + f(v_T, a_T)$$



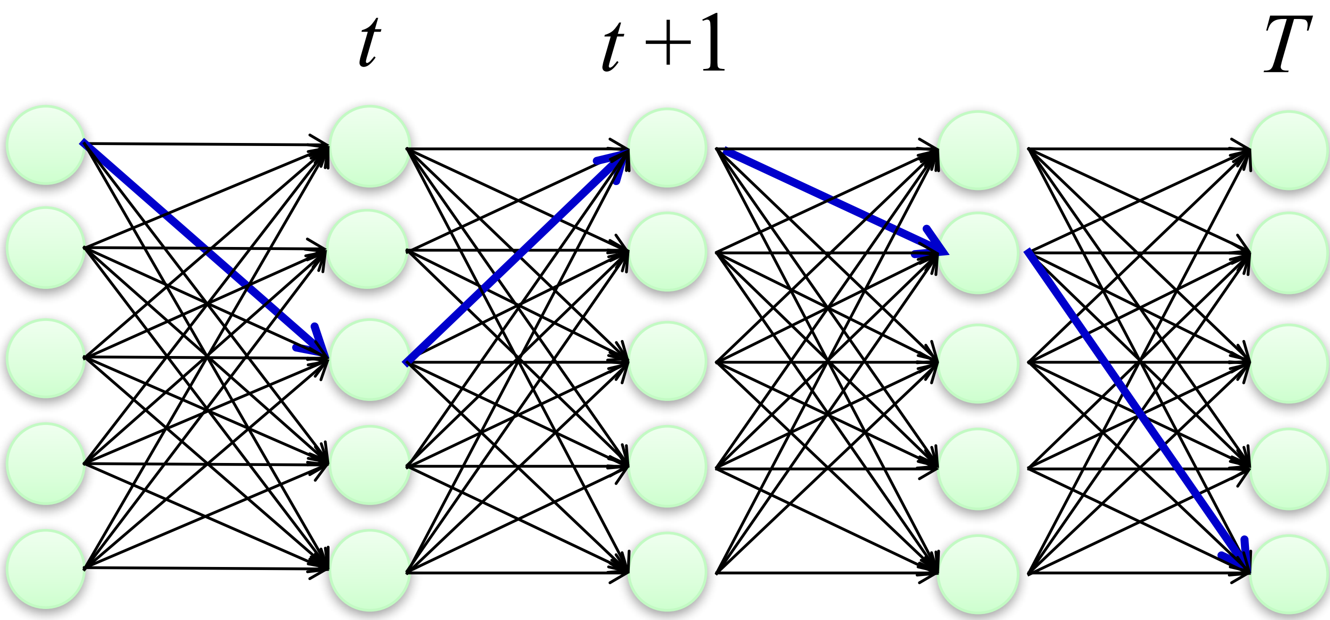
評価値: $f(v_t, a_t)$

最適性の原理(Principle of Optimality)

[R.E. Bellman,1957]

動的計画法の基本原理

t 段階の状態 v_t から T（最終）段階に至る最適な行動（選択）があったとき、それは $t+1$ 段階の状態 v_{t+1} から T（最終）段階に至る最適な行動（選択）になっている

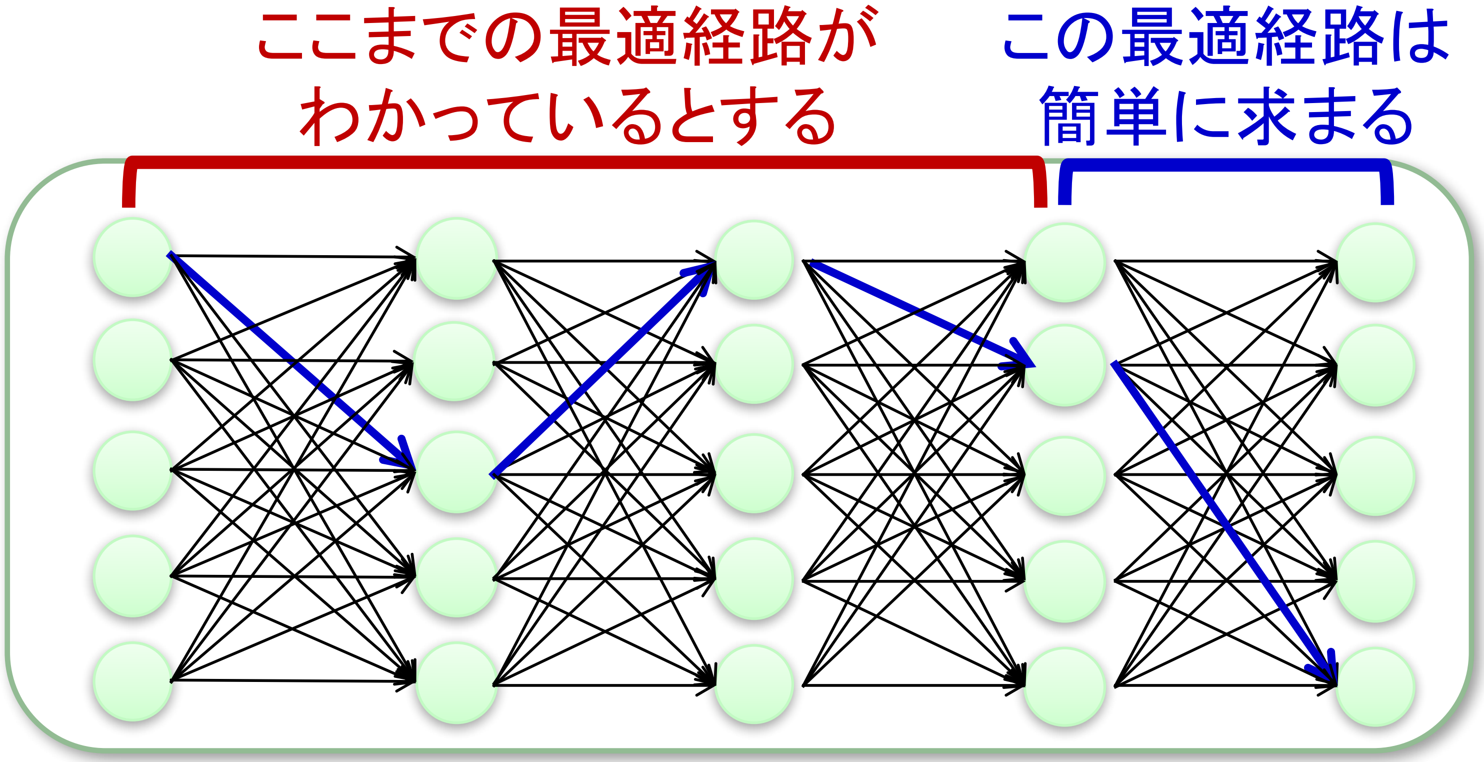


ビタビ(Viterbi)アルゴリズム

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

$$q_{t+1}(k) = \max_{s_{1:t}} p(o_{1:t+1}, s_{t+1}=k, s_{1:t})$$

とすると



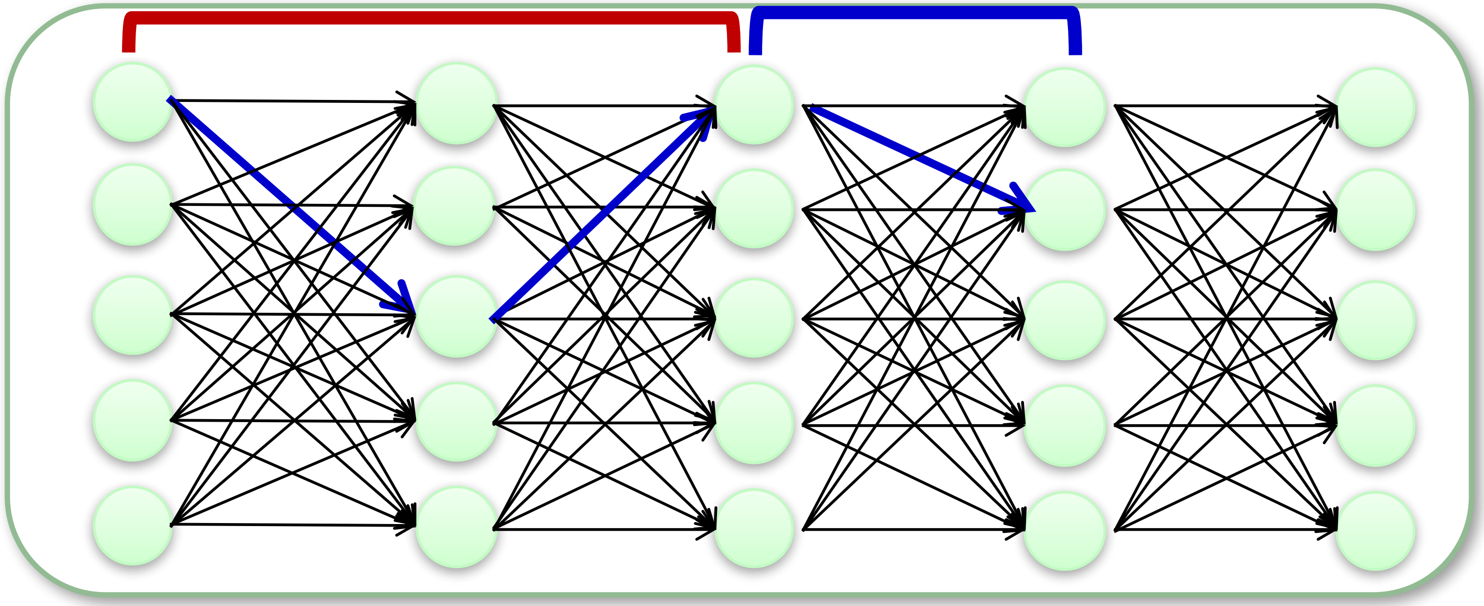
ビタビ(Viterbi)アルゴリズム

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

$$q_{t+1}(k) = \max_{s_{1:t}} p(o_{1:t+1}, s_{t+1}=k, s_{1:t})$$

とすると

ここまでの最適経路がわかっているとする　この最適経路は簡単に求まる



ビタビ(Viterbi)アルゴリズム

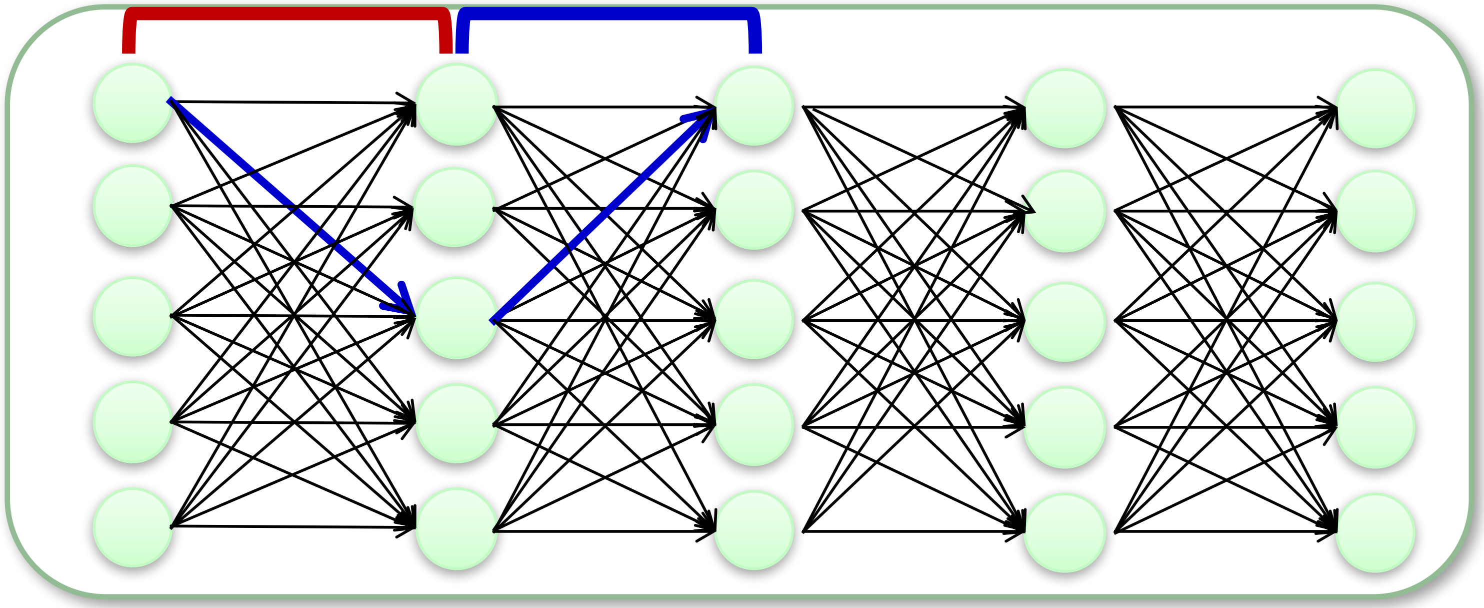
$t+1$ までの観測系列 $o_{1:t+1}$ において、$s_{t+1}=k$ に至る状態系列の最大確率を

$$q_{t+1}(k) = \max_{s_{1:t}} p(o_{1:t+1}, s_{t+1}=k, s_{1:t})$$

とすると

ここまでの最適経路がわかっているとする

この最適経路は簡単に求まる

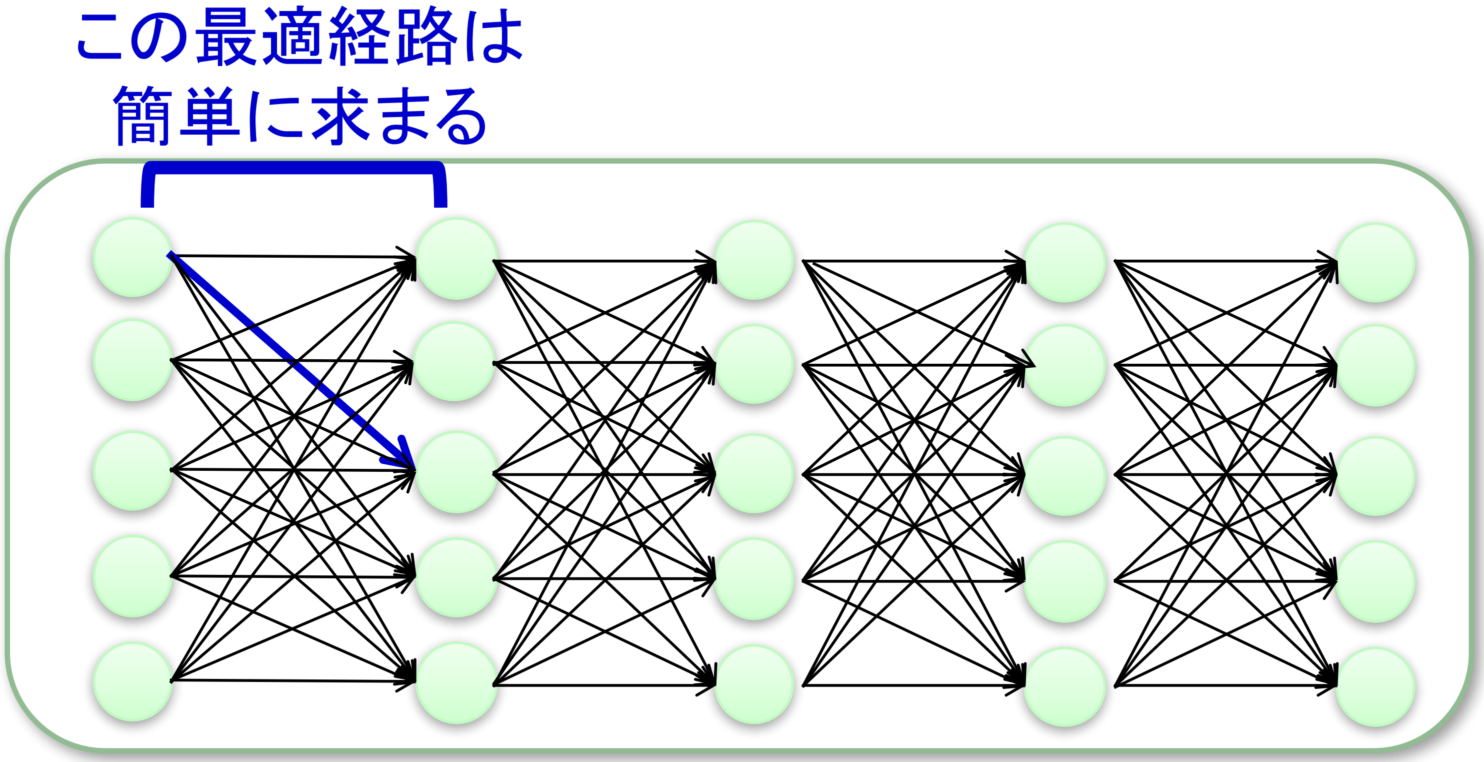


ビタビ(Viterbi)アルゴリズム

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

$$q_{t+1}(k) = \max_{s_{1:t}} p(o_{1:t+1}, s_{t+1}=k, s_{1:t})$$

とすると



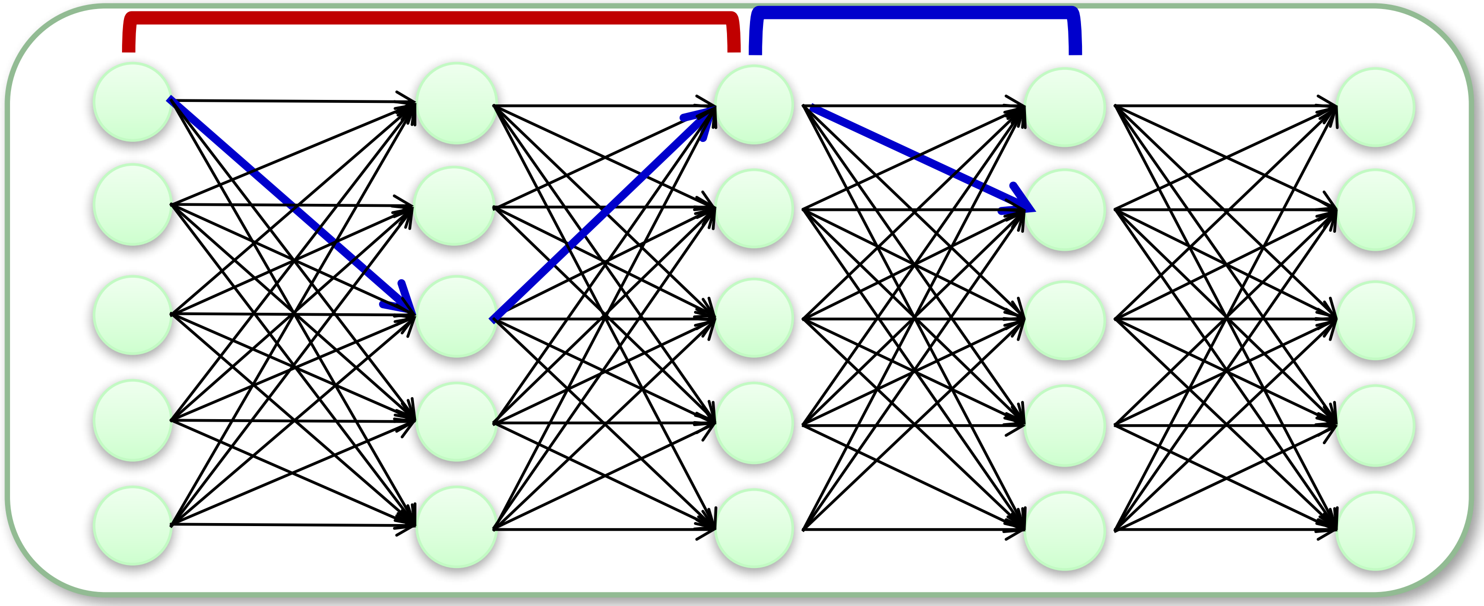
ビタビ(Viterbi)アルゴリズム

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

$$q_{t+1}(k) = \max_{s_{1:t}} p(o_{1:t+1}, s_{t+1}=k, s_{1:t})$$

とすると, 以下のように再帰的に書ける

$$q_{t+1}(k) = \max_{i} \left[\underline{q_t(i)}\, \underline{a_{i,k}} \right] \underline{b_k(o_{t+1})}$$



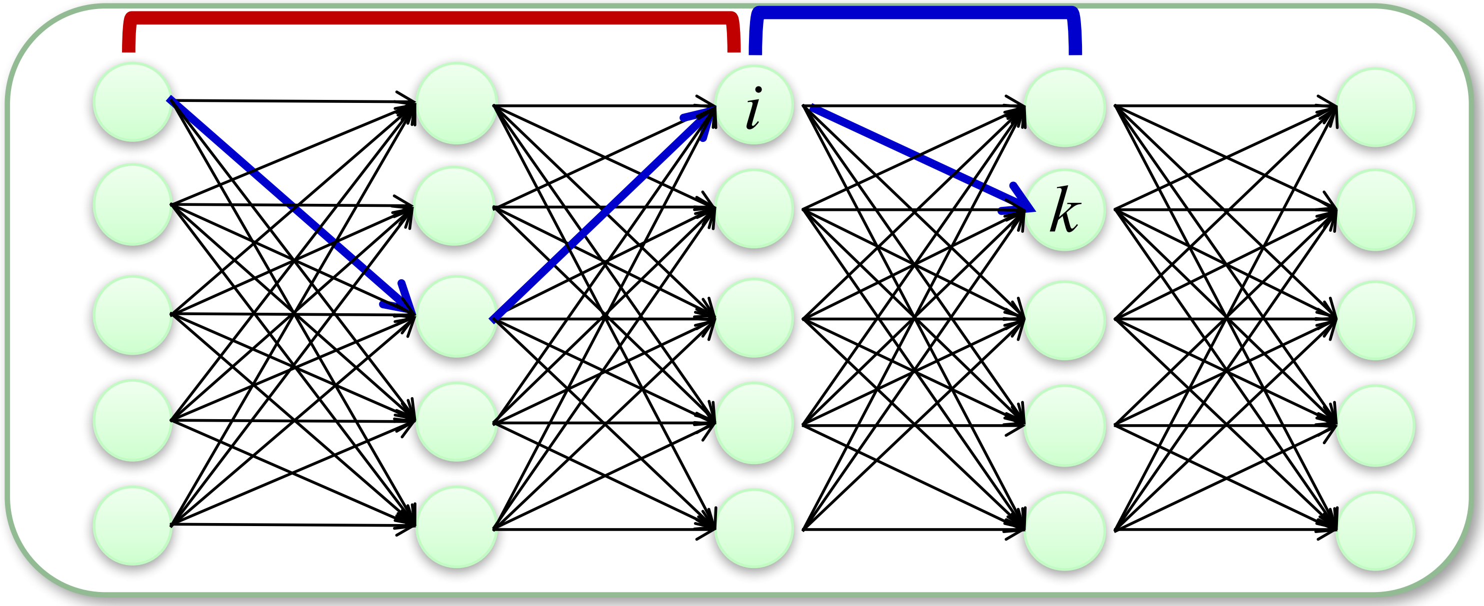
ビタビ(Viterbi)アルゴリズム

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

$$q_{t+1}(k) = \max_{s_{1:t}} p(o_{1:t+1}, s_{t+1}=k, s_{1:t})$$

とすると, 以下のように再帰的に書ける

$$q_{t+1}(k) = \max_i \left[q_t(i) a_{i,k} \right] b_k(o_{t+1})$$



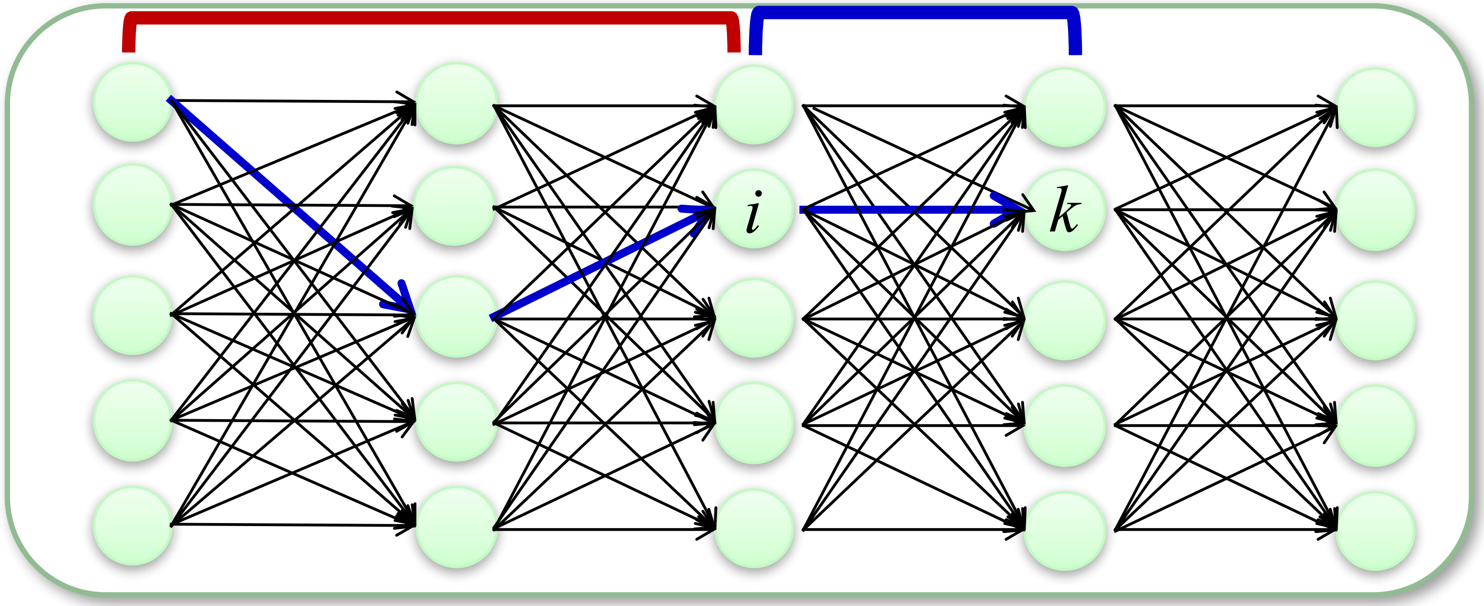
ビタビ(Viterbi)アルゴリズム

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

$$q_{t+1}(k) = \max_{s_{1:t}} p(o_{1:t+1}, s_{t+1}=k, s_{1:t})$$

とすると, 以下のように再帰的に書ける

$$q_{t+1}(k) = \max_{i} \left[q_t(i) a_{i,k} \right] b_k(o_{t+1})$$



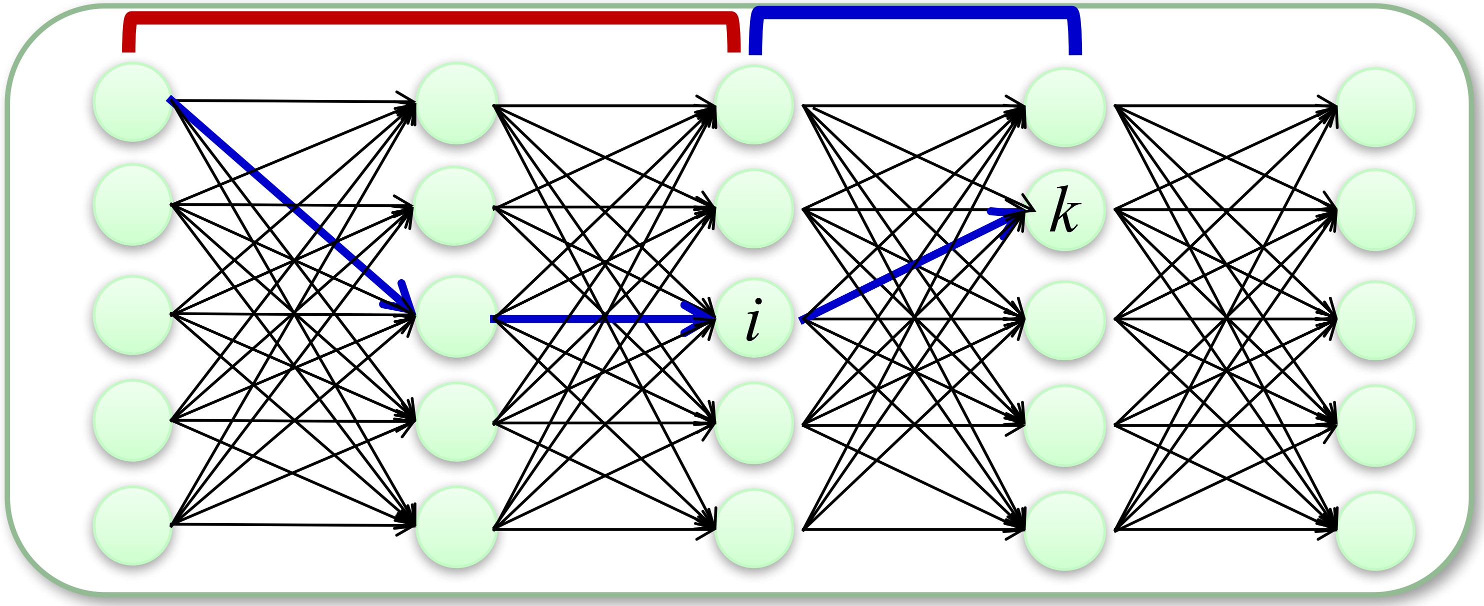
ビタビ(Viterbi)アルゴリズム

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

$$q_{t+1}(k) = \max_{s_{1:t}} p(o_{1:t+1}, s_{t+1}=k, s_{1:t})$$

とすると, 以下のように再帰的に書ける

$$q_{t+1}(k) = \max_{i}\left[q_t(i)a_{i,k}\right]b_k(o_{t+1})$$



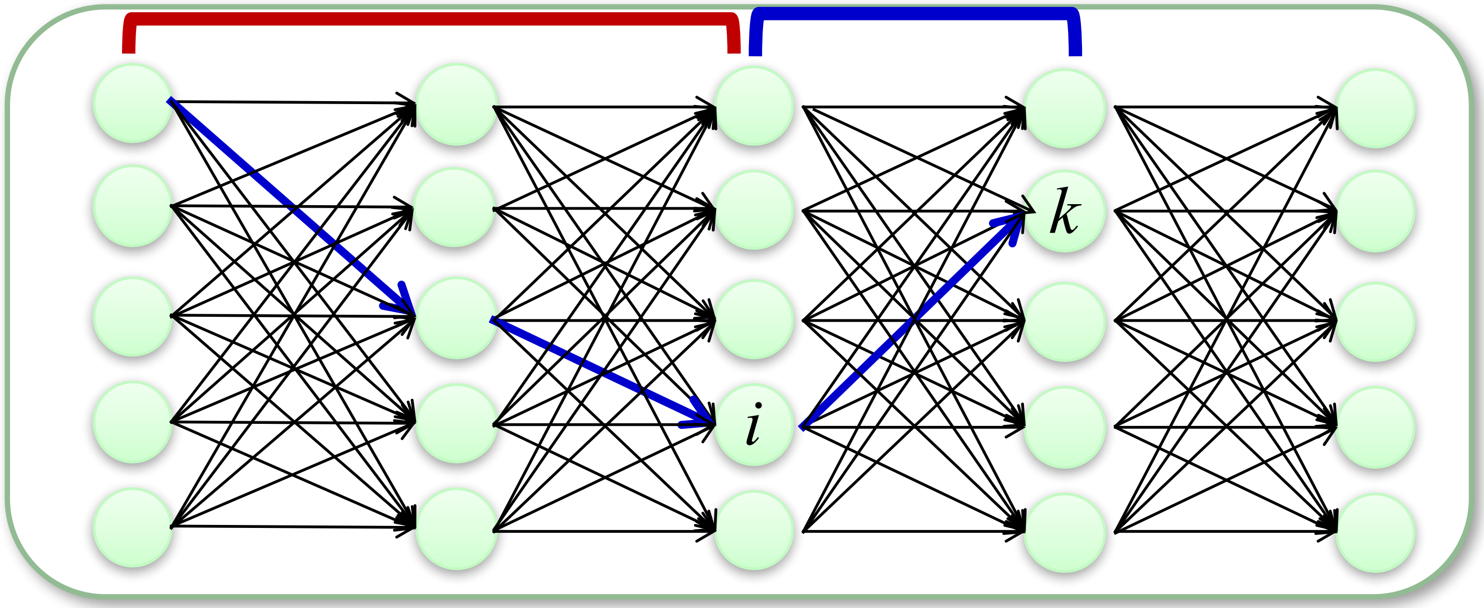
ビタビ(Viterbi)アルゴリズム

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

$$q_{t+1}(k) = \max_{s_{1:t}} p(o_{1:t+1}, s_{t+1}=k, s_{1:t})$$

とすると, 以下のように再帰的に書ける

$$q_{t+1}(k) = \max_{i} \left[q_t(i)\, a_{i,k} \right] b_k(o_{t+1})$$



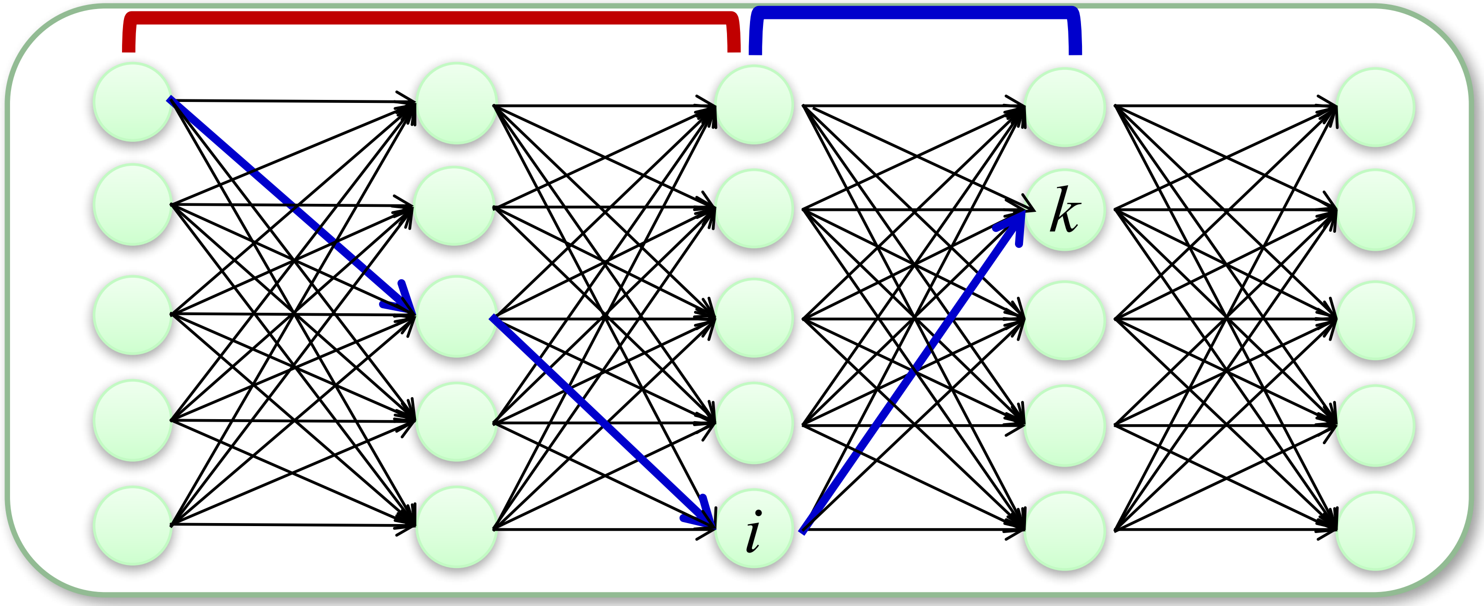
ビタビ(Viterbi)アルゴリズム

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

$$q_{t+1}(k) = \max_{s_{1:t}} p(o_{1:t+1}, s_{t+1}=k, s_{1:t})$$

とすると, 以下のように再帰的に書ける

$$q_{t+1}(k) = \max_{i} \left[\underline{q_t(i)}\, \underline{a_{i,k}} \right] \underline{b_k(o_{t+1})}$$



ビタビ(Viterbi)アルゴリズム

前向きステップ：経路の確率計算

後ろ向きステップ：最適経路の復元

ビタビ(Viterbi)アルゴリズム

1. 各状態 k =1, 2, …, K に対して初期化

$$q_1(k) = \pi_k b_k(o_1),\ \ \xi_1(k) = 0$$

$O(TK)$

2. t = 1, 2, …, T, k = 1, 2, …, K に対する再帰計算:

$$q_{t+1}(k) = \max_i \left[q_t(i) a_{i,k}\right] b_k(o_{t+1})$$

$$\xi_{t+1}(k) = \operatorname*{argmax}_i \left[q_t(i) a_{i,k}\right]$$

$O(K)$

3. 再帰計算終了時の確率

$$q^* = \max_k q_T(k)$$

$$s_T^* = \operatorname*{argmax}_k q_T(k)$$

$O(TK^2)$

4. バックトラック: $s_t^* = \xi_{t+1}(s_{t+1}^*)$

ビタビ(Viterbi)アルゴリズム

1. 各状態 k =1, 2, …, K に対して初期化:

$$q_1(k) = \pi_k b_k(o_1),\ \ \xi_1(k) = 0$$

2. t = 1, 2, …, T, k = 1, 2, …, K に対する再帰計算:

$$q_{t+1}(k) = \max_i \left[q_t(i) a_{i,k} \right] b_k(o_{t+1})$$

$$\xi_{t+1}(k) = \operatorname{argmax}_i \left[q_t(i) a_{i,k} \right]$$

3. 再

4. バッ

対数尤度で考えれば加算演算だけで
アルゴリズムを作ることができる

$$\log q_{t+1}(k) = \max_i \left[\log q_t(i) + \log a_{i,k} \right] + \log b_k(o_{t+1})$$

HMMによる品詞タグ付け

- 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
  - 授業の後半で扱う

- 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
  - 以下、ここから説明する

宿題

1. 遷移確率が直前 m 個の状態に依存する場合のビタビアルゴリズムを考え、その時間計算量を求めよ
   - 遷移確率 $p(k \mid s_{t\text{-}m+1}, \ldots, s_{t\text{-}1}, s_t) = a_{t\text{-}m+1,\ldots,t\text{-}1,t,k}$
2. HMM が適用できる独自の問題を考え、状態集合・記号集合を定義せよ。HMM でうまく問題が解ける場合と解けない場合について考察せよ。