

13. 系列データの識別

13.1 ラベル系列に対する識別

- ラベル系列に対する識別問題の分類
 - 入力の系列長と出力の系列長が等しい
 - 例) 形態素解析、固有表現抽出
 - 系列ラベリング問題 \Rightarrow CRF
 - 入力の系列長に関わらず出力の系列長が 1
 - 例) 動画像の分類、文書分類
 - 系列識別問題 \Rightarrow HMM
 - 入力の系列長と出力の系列長に対応関係がない
 - 例) 連続音声認識、機械翻訳
 - 系列変換問題 \Rightarrow RNN, Seq2Seq+Attention, Transformer

13.2 系列ラベリング問題— CRF—

- 系列ラベリング問題とは
 - 入力系列の個々の要素に対してラベルを付与する
 - 系列の要素の出現確率は前後の要素に依存
 - 1 入力 1 出力の識別器を連続的に適用する方法では性能が上がらない
 - ⇒ 入力や出力の系列としての特徴を使う
 - 可能な出力系列の組合せは膨大な数になるので、単純な事後確率最大法は使えない
 - ⇒ 探索によって最適解を求める

13.2 系列ラベリング問題— CRF—

- 系列ラベリング問題の事例

- 形態素解析

入力	系列	で	入力	さ	れる	各	要素
出力	名詞	助詞	名詞	動詞	接尾辞	接頭辞	名詞

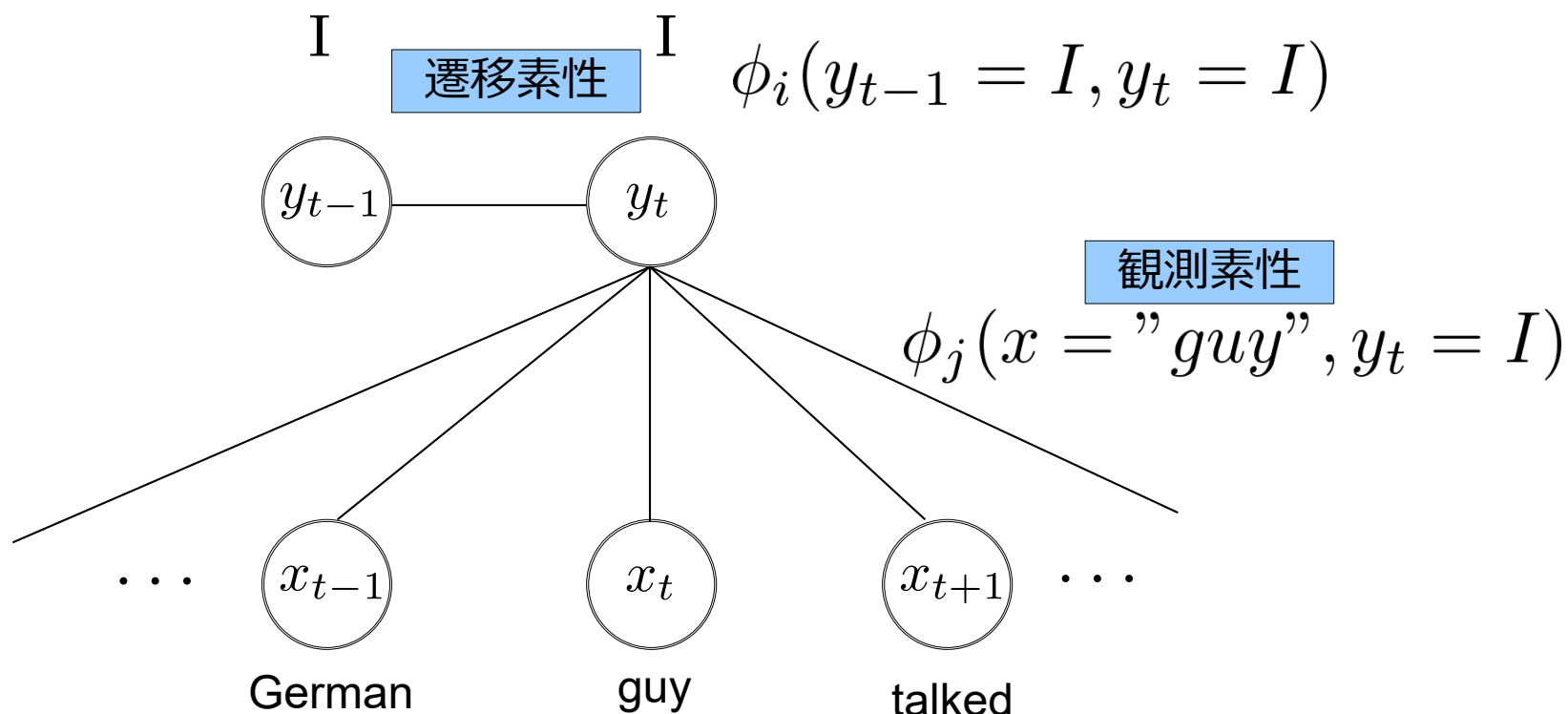
- 固有表現抽出（例：人を指す表現の抽出）

入力	Suddenly,	the	tall	German	guy	talked	to	me
出力	O	B	I	I	I	O	O	B

B: begin
I: inside
O: outside

13.2 系列ラベリング問題— CRF—

- 線型モデルによる系列ラベリング
 - 素性関数の導入
 - 入力系列 x と出力系列 y との間に定義される関数
 - 関係が成立すれば 1、不成立なら 0 を値とする



13.2 系列ラベリング問題— CRF—

- 対数線型モデル (softmax)

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_{\mathbf{x},\mathbf{w}}} \exp(\mathbf{w}^T \phi(\mathbf{x}, \mathbf{y}))$$

$$Z_{\mathbf{x},\mathbf{w}} = \sum_{\mathbf{y}} \exp(\mathbf{w}^T \phi(\mathbf{x}, \mathbf{y}))$$

識別モデル

あるクラスの事後確率が上がれば、他のクラスは下がる

- 出力の決定

$$\mathbf{y}^* = \arg \max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$$

$$= \arg \max_{\mathbf{y}} \frac{1}{Z_{\mathbf{x},\mathbf{w}}} \exp(\mathbf{w}^T \phi(\mathbf{x}, \mathbf{y}))$$

$$= \arg \max_{\mathbf{y}} \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$$

13.2 系列ラベリング問題— CRF—

- 素性関数の制限（含まれる出力系列を 2 に限定）

$$\mathbf{y}^* = \arg \max_{\mathbf{y}} \sum_t \mathbf{w}^T \phi(\mathbf{x}, y_t, y_{t-1})$$

- ビタビアルゴリズムによって探索が可能

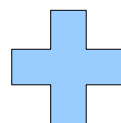
Algorithm	ビタビアルゴリズム
<hr/>	
for $t = 2$ to $ \mathbf{x} $ do	
for all y_t do	
$\alpha(t, y_t) = \max_{y_{t-1}} \{ \mathbf{w} \cdot \phi(\mathbf{x}, y_t, y_{t-1}) + \alpha(t-1, y_{t-1}) \}$	
$B(t, y_t) = \arg \max_{y_{t-1}} \{ \mathbf{w} \cdot \phi(\mathbf{x}, y_t, y_{t-1}) + \alpha(t-1, y_{t-1}) \}$	
end for	
end for	
$\mathbf{y}^* = \alpha$ の最大値に対応する B を逆に辿る	

13.2 系列ラベリング問題— CRF—

• CRF++ : CRF のツールキット

毎年 M1 名詞 - 副詞可能 B-DATE
1 2月 T5-1 名詞 - 副詞可能 I-DATE
中旬 M1 名詞 - 副詞可能 I-DATE
に S9 助詞 - 格助詞 - 一般 O
スピード M10 名詞 - 一般 O
スケート M10 名詞 - 一般 O
の S9 助詞 - 連体化 O
浅間 M1 名詞 - 固有名詞 - 地域 - 一般 B-LOCATION
選抜 M1 名詞 - サ変接続 O
大会 M1 名詞 - 一般 O
が S9 助詞 - 格助詞 - 一般 O
開か T1-9 動詞 - 自立 O
れる M9 動詞 - 接尾 O

学習データ



```
# Unigram
U00:%x[-2,0]
U01:%x[-1,0]
U02:%x[0,0]
U03:%x[1,0]
U04:%x[2,0]
U05:%x[-1,0]/%x[0,0]
U06:%x[0,0]/%x[1,0]

U10:%x[-2,1]
U11:%x[-1,1]
U12:%x[0,1]
U13:%x[1,1]
U14:%x[2,1]
...
```

X[-2,0]: 現在の単語から
見て 2 単語前の第 0 列
の値

テンプレート

crf_learn コマンド

- a 正則化法
- c 正則化項の重み
- f 語彙のカットオフ

モデル

Python でのコーディング : PyStruct

- 構造化学習の定式化

$$y^* = \arg \max_y w^T \phi(x, y)$$

- 線形識別モデルの分解

$$w^T \phi(x, y) = \sum_{i \in V} w_i^T \phi_i(x, y_i) + \sum_{(i,j) \in E} w_{ij}^T \phi_{ij}(x, y_i, y_j)$$

重みは learners
で学習

V は出力記号の集合

出力記号間の関係 E は
models で与える

```
model = ChainCRF()
ssvm = FrankWolfeSSVM(model=model, C=.1, max_iter=11)
ssvm.fit(X_train, y_train)
```


13.3 系列識別問題— HMM—

- 例題

- PC 操作系列による熟練度の判定

- k: キーボード、 g: マウス、 e: エラー
- 初心者の入力系列例

k e k g k e k g g k g k k e g e e k e e e g e

- 熟練者の入力系列例

k k e k g k k k e k g k g g g e g k g

- 判定したい入力系列

k g e k g k k g e k g e k e e k e g e k

13.3 系列識別問題—HMM—

- 生成モデルによるアプローチ
 - 系列識別問題ではクラスの事前確率を得られることが多い

$$\begin{aligned} y^* &= \arg \max_y P(y|\mathbf{x}) \\ &= \arg \max_y \frac{P(\mathbf{x}, y)}{P(\mathbf{x})} \\ &= \arg \max_y \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \\ &= \arg \max_y P(\mathbf{x}|y)P(y) \end{aligned}$$

生成モデル

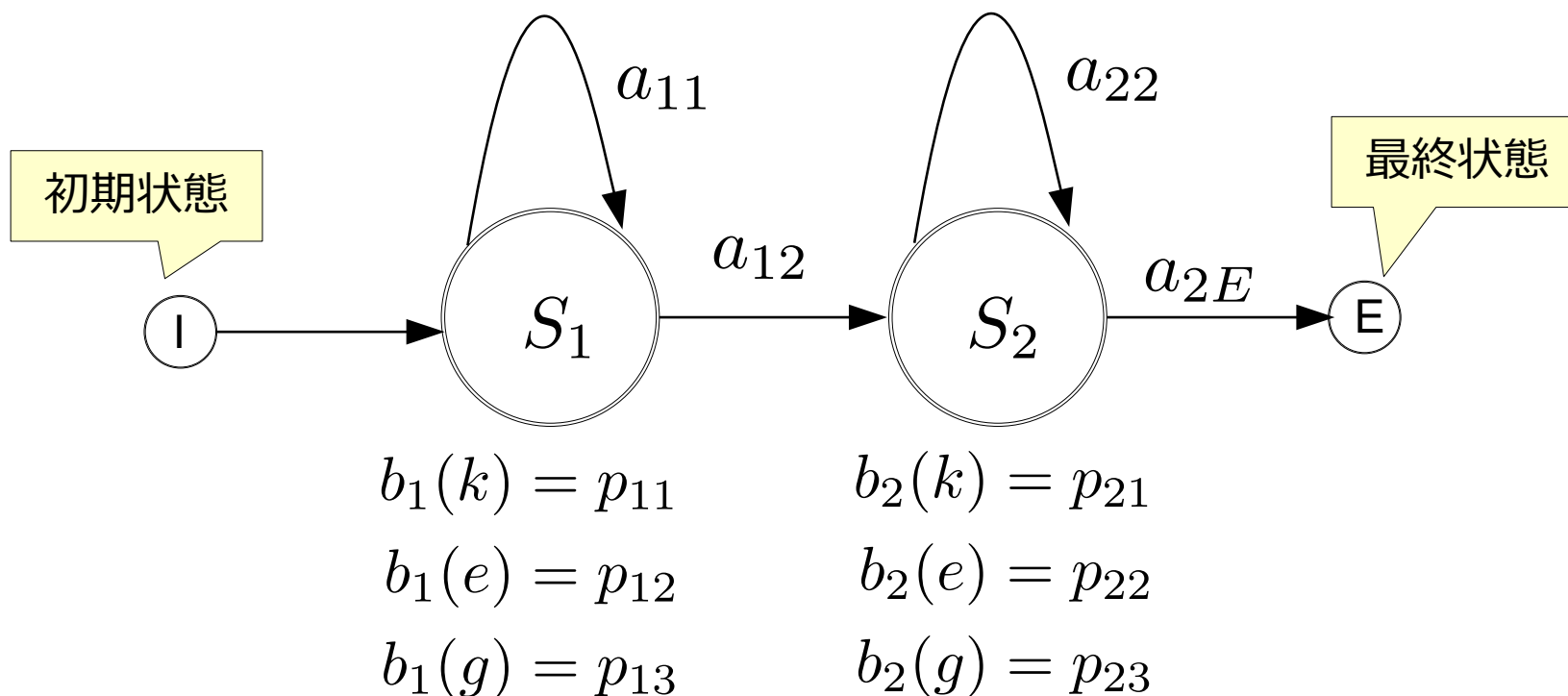
尤度

事前確率

尤度は、あるクラスの確率モデルを、他のクラスとは無関係に求めている

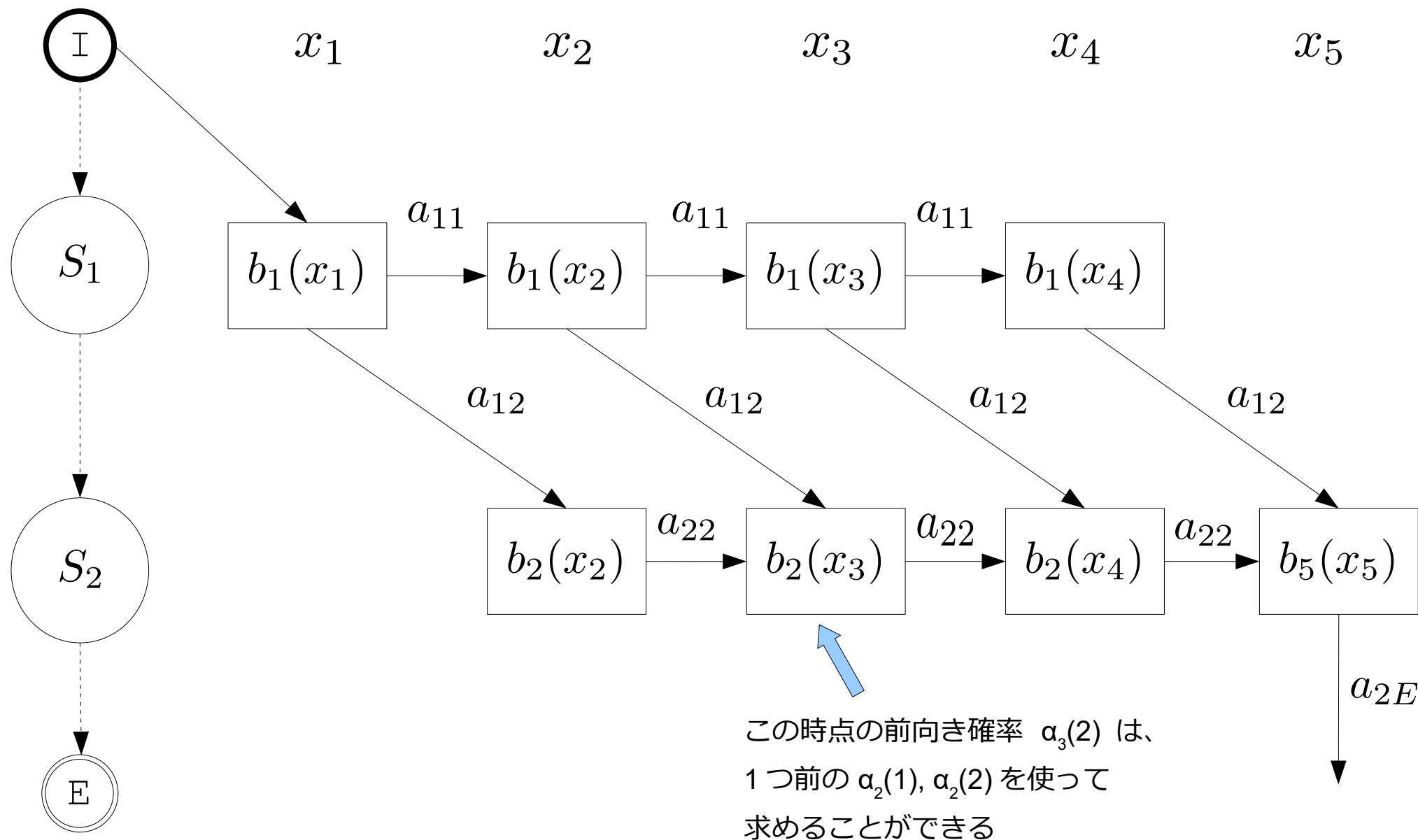
13.3 系列識別問題—HMM—

- 不定長入力に対する尤度計算法
 - 自己遷移を持つ確率オートマトンを用いる



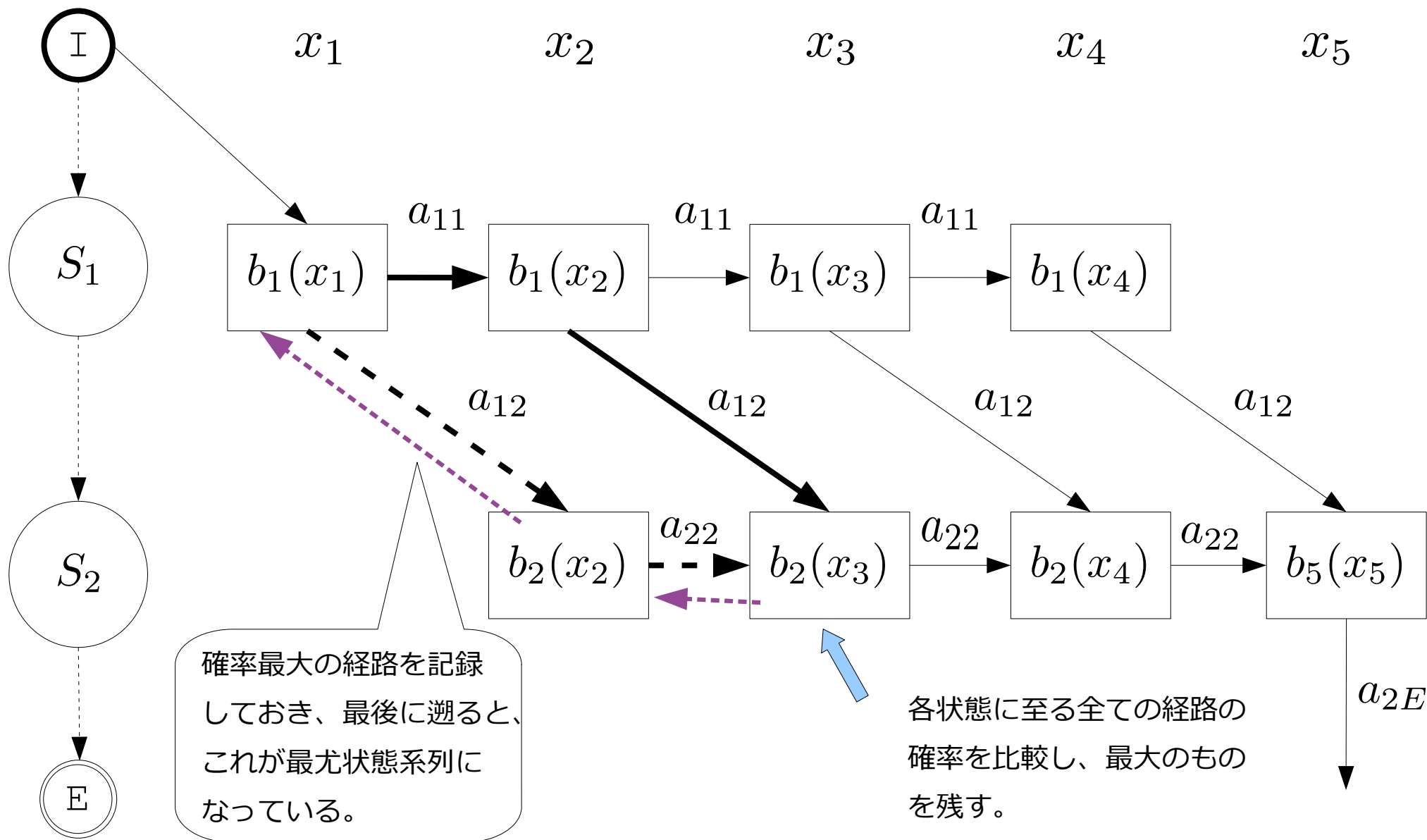
13.3 系列識別問題— HMM—

前向きアルゴリズム（正確な計算）



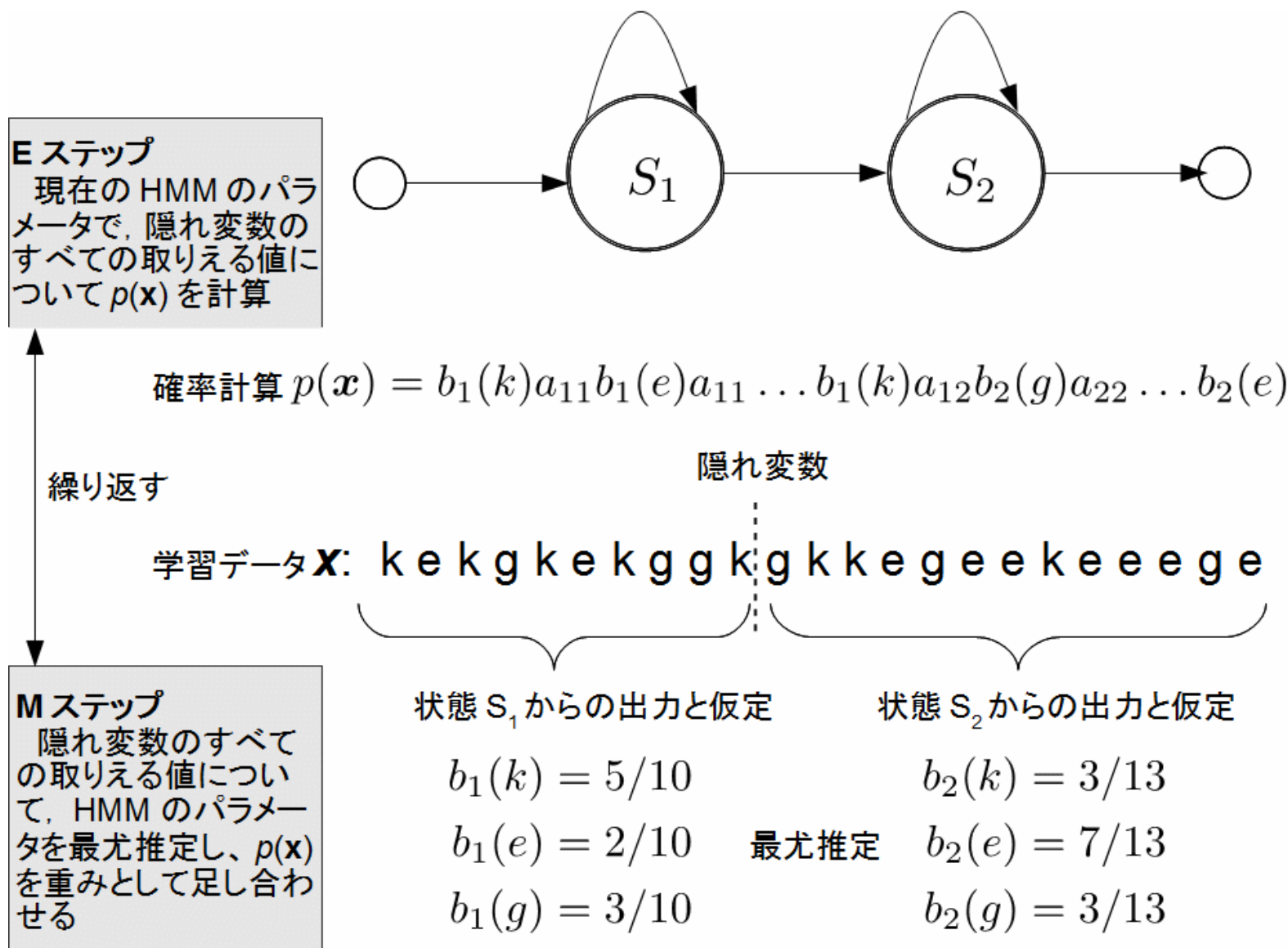
13.3 系列識別問題—HMM—

- ビタビアルゴリズムを用いた探索



13.3 系列識別問題— HMM—

• HMM の学習 : EM アルゴリズム



補足

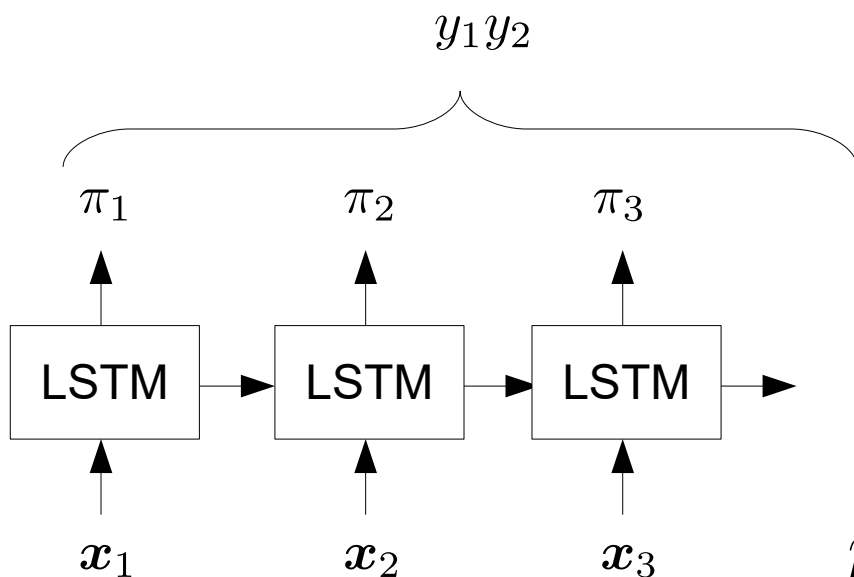
系列変換問題 — RNN—

- 系列変換問題の定式化
 - 入力系列 $\mathbf{x} = \mathbf{x}_1, \dots, \mathbf{x}_T$
 - 出力系列 $\mathbf{y} = y_1, \dots, y_L$
 - 一般に $T > L$
 - 系列識別と探索を組み合わせた複雑な処理が必要
- End-to-End アプローチ
 - 入力から出力への変換をニューラルネットワークで学習（音声認識の事例で説明）
 - Connectionist temporal classification
 - Attention モデル

Connectionist temporal classification

- アイデア

- 出力記号に blank 記号 `_` を加えて、入力長と出力長を合わせる
- 正解系列に変換可能な出力系列の確率の和を求める



- h_{ai} という正解系列に対して

`_h__a__i`
`hhh_aaaa_i`
`hh____ai__`

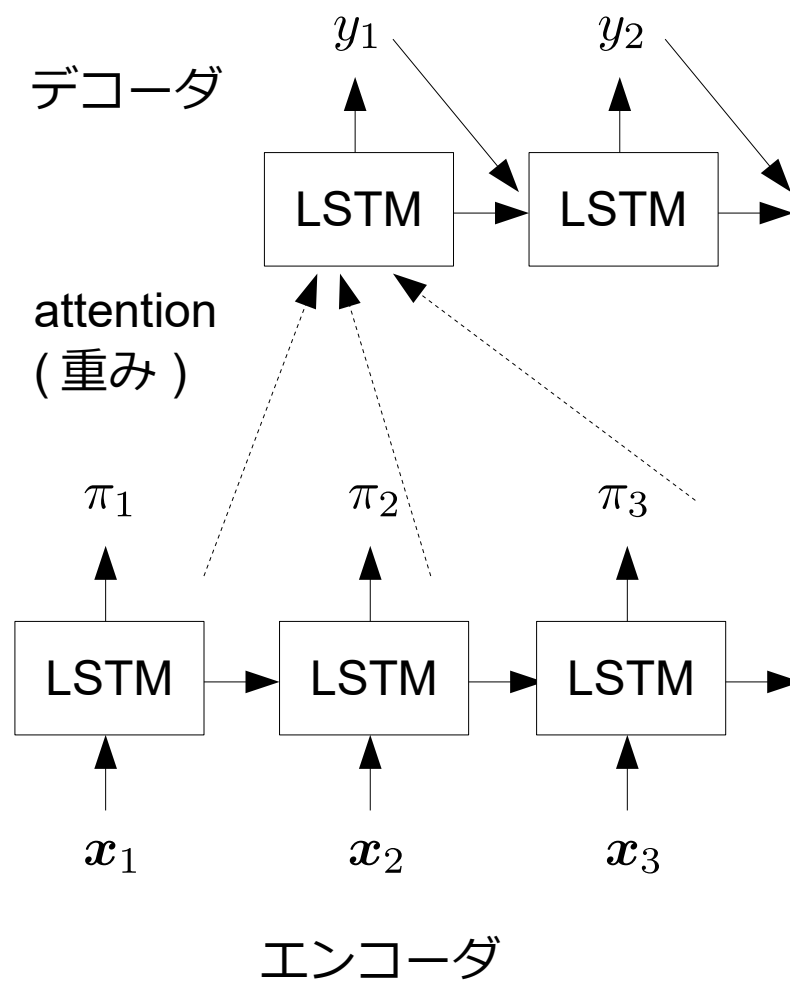
などの出力系列を正解とみなす

$$p(\mathbf{y}|\mathbf{x}) = \sum_{\boldsymbol{\pi} \rightarrow \mathbf{y}} p(\boldsymbol{\pi}|\mathbf{x}) = \sum_{\boldsymbol{\pi} \rightarrow \mathbf{y}} \prod_{i=1}^T p(\pi_i|\mathbf{x}_i)$$

Attention モデル

- アイディア

- 特徴ベクトルを分散表現に変換するエンコーダと、分散表現から出力を求めるデコーダの組み合わせ
- 一定範囲の分散表現から出力を計算するために注意機構 (attention) を用いる



Attention モデル

- Attention とは
 - 出力決定時に入力系列のどの部分の情報を重視するかを決めるメカニズム
 - 入力系列の隠れ状態ベクトルの重み付き和で表現
 - 重みは出力の学習時に調整
- 参考サイト
<https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/>