知的情報処理システム特論Ⅰ　第7回

二宮　崇

今日の講義の予定

- **生成モデルと識別モデル**
- **確率的識別モデル**
  - 構造予測のための多クラスロジスティック回帰
  - 条件付き確率場 (CRF)
  - Linear-chain CRF

- **教科書**
  - 高村大也. **言語処理のための機械学習**. コロナ社.
  - Yusuke Miyao (2006) From Linguistic Theory to Syntactic Analysis: Corpus-Oriented Grammar Development and Feature Forest Model, Ph.D Thesis, University of Tokyo
  - Cristopher M. Bishop “PATTERN RECOGNITION AND MACHINE LEARNING” Springer, 2006

Generative Model and Discriminative Model

生成モデルと識別モデル

生成モデルから識別モデルへ

- **生成モデル**
  - HMM
    - 状態遷移による生成
    - 状態遷移の確率は一つ前の状態のみに依存
    - 任意の文xと状態列qに対する同時確率$p(x, q)$を計算できる

生成モデルの改良の先に

- **条件部を詳細化**
  - HMM: $p(q_j|q_{j-1}, q_{j-2})$

- **素性（特徴）という考え方**

  条件部をどんどんリッチにして線形補間をとればよいのでは？→HMMの状態xの確率$p(x) = p(x|x$周辺の状況$)$

生成モデルの問題点

- **生成モデルの問題点**
  - 独立性の仮定
    - HMMは一つ前の状態にのみ依存する(マルコフ性)
    - 二つ前の状態と今の状態は何らかの関係があるかもしれないけど、独立した事象として扱われている。
    - 例: 左辺(“I have a”)の方が右辺(“I have a pen”)より不自然なのに必ず確率が高くなってしまう。

$$p(\text{"I"})p(\text{"have"})p(\text{"a"}) > p(\text{"I"})p(\text{"have"})\,p(\text{"a"})p(\text{"pen"})$$

生成モデルの問題点

- **生成モデルでの予測**

$$\hat{y} = \underset{y}{\operatorname{argmax}}\, p(y|x;\theta) = \underset{y}{\operatorname{argmax}}\, p(x,y;\theta)$$

  - しかし、

$$p(x,y;\theta) = p(y|x;\theta)p(x;\theta)$$

    であるから$p(x;\theta)$ が計算できてしまう分、冗長なモデルになっている。間接的に分類問題を解いている。
  - 独立性を仮定した事象に分解しないと同時確率を計算することができない

識別モデル

- **識別モデル**

  直接

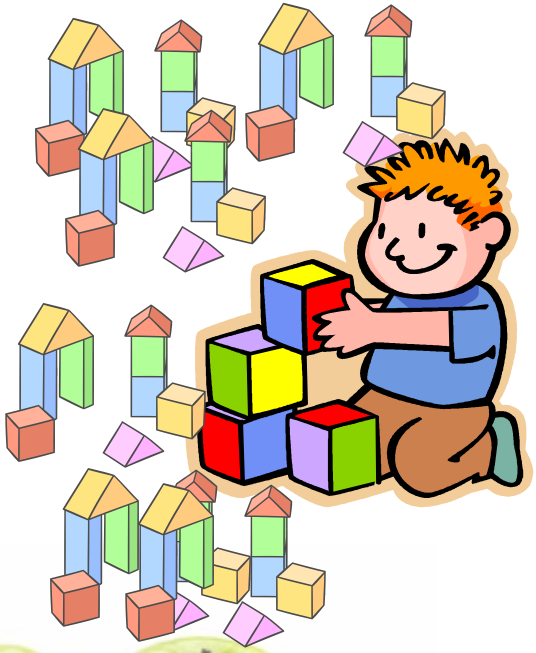
$$\hat{y} = \underset{y}{\operatorname{argmax}}\, p(y|x;\theta)$$

  を解く

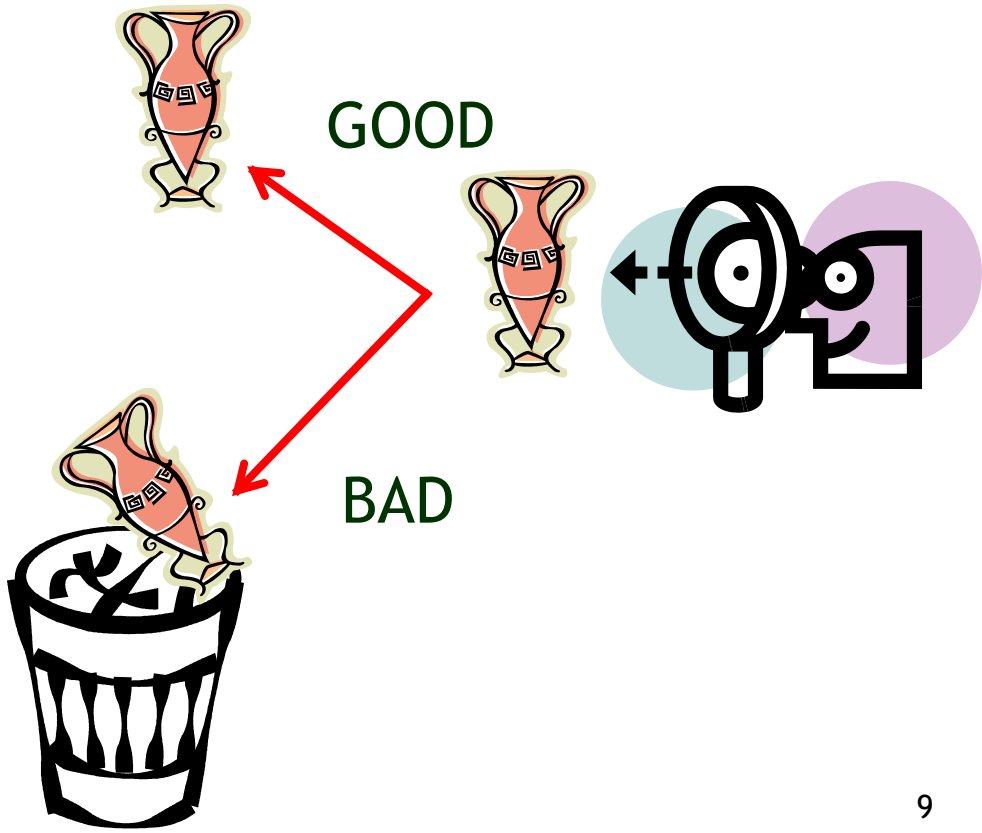
  - 独立な事象を仮定しない
  - 「条件部の確率」をモデルにいれない

生成モデルと識別モデル (イメージ)

生成モデル

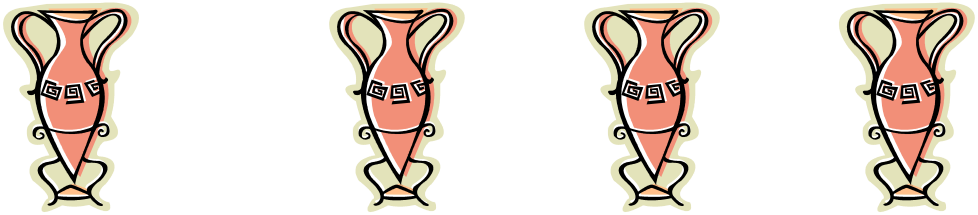


識別モデル

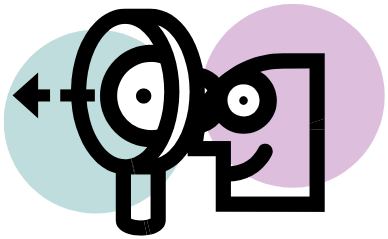


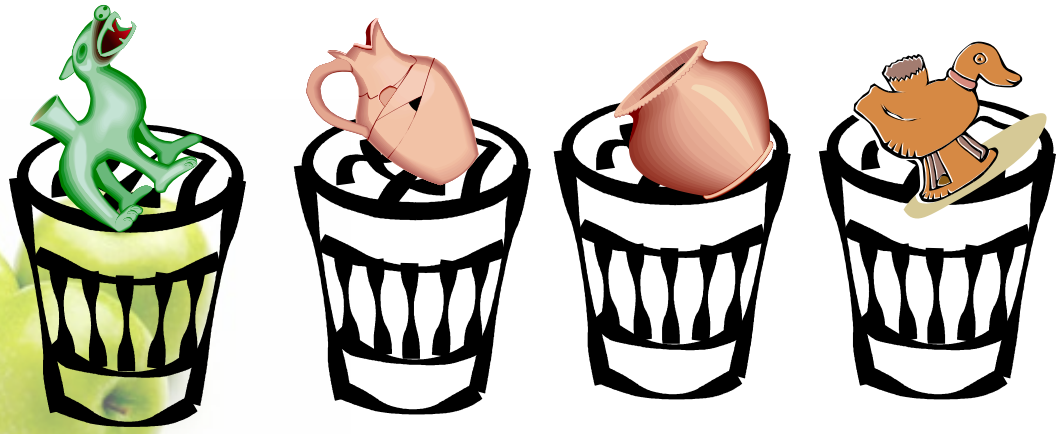
識別するための訓練

- **教師付学習**
  - 良い例と悪い例を与えて、どこに注目すればより良く識別出来るのか学習



good examples

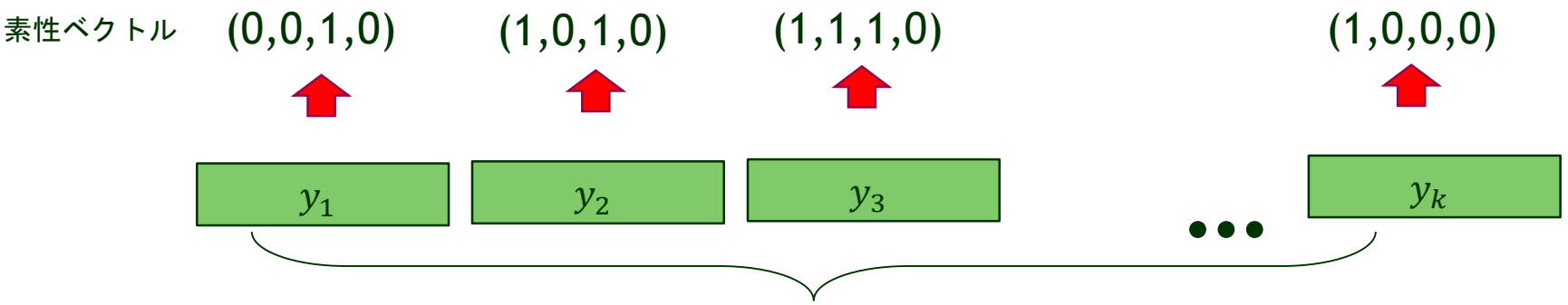




bad examples

識別モデル

- **品詞解析の例**
  - 入力 x =“A blue eye girl with white hair and skin walked”

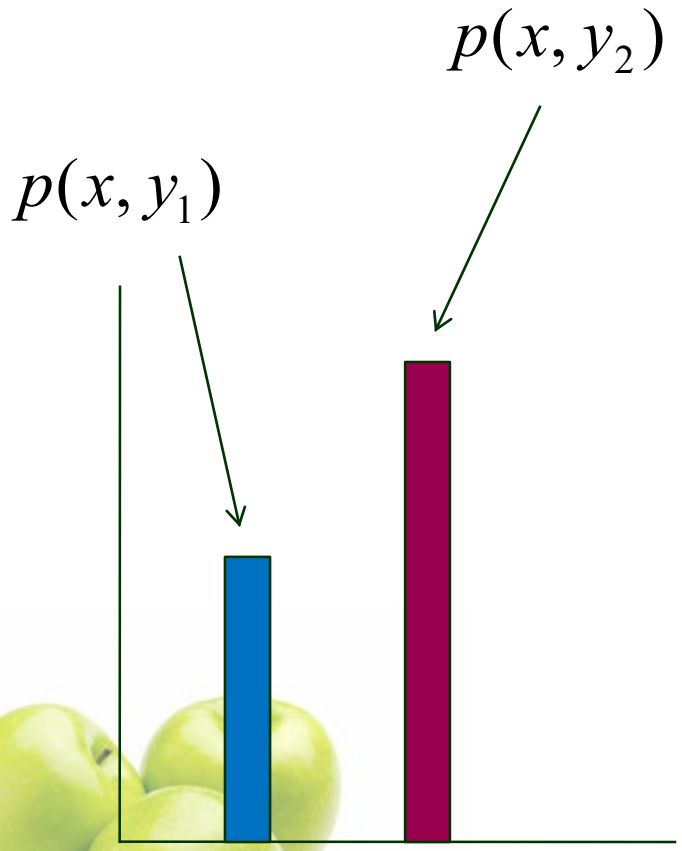


xに対する全ての可能な品詞列集合

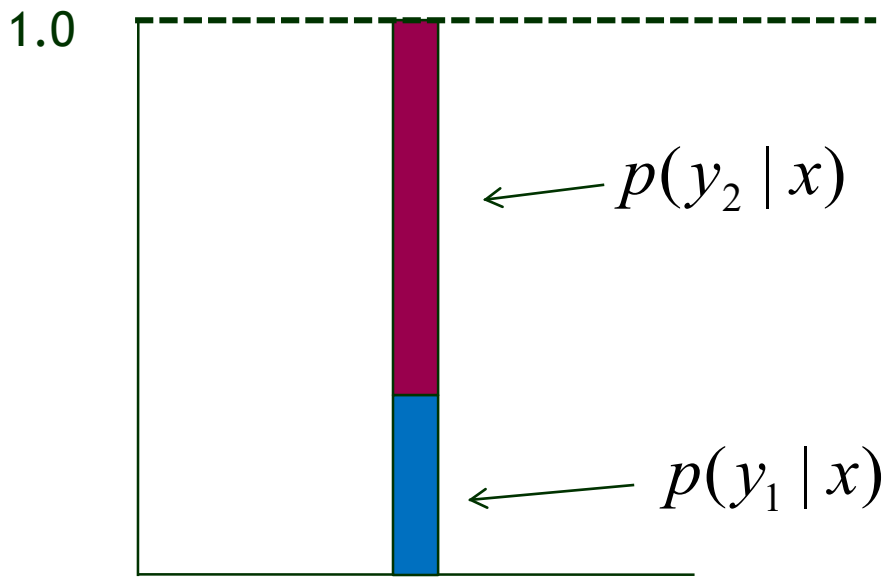
$p(y_3|x)$は$\{y_1, y_2, y_3, \ldots, y_k\}$から$y_3$を選択する確率

生成モデルと識別モデル

生成モデル



識別モデル



確率的識別モデル

問題設定

- **入力**: x (**任意の入力**)
- **出力**: $y \in Y(x)$
  - $Y(x)$: 入力xに依存して列挙できる有限の無向グラフ集合 ($Y(x) \subset G$)
- **訓練データ**
  - $(x_i, y_i)_{i=1,\ldots,n}$
  - 例
    - xは競馬情報、yは1位の馬 (yはラベル)
    - xは文、yはxに対する正解の品詞列 (yはラベル列)
    - xは文、yはxに対する正解の構文木 (yは木グラフ)
- **推論**
  - ある未知の入力xに対する出力yの予測

素性関数

- **入力と出力から特徴を抽出する素性関数(feature function)を複数定義**

$$\phi_j(x, y) \quad j = 1, \ldots, D$$

  - 注意
    - 人手で設計するか、素性関数を学習する仕組み(深層学習など)が必要
    - xだけでなくyからも特徴を抽出できる (yから特徴を抽出しなければならない)

- **素性ベクトル(feature vector)**

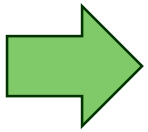
$$\boldsymbol{\phi}(x, y) = \left(\phi_1(x, y), \phi_2(x, y), \ldots, \phi_D(x, y)\right)$$

全体の流れ(1/2)

- **パラメータ推定**
  - 各素性ϕ_jに対する重みw_jを学習

訓練データ

入力	出力
x_1	y_1
x_2	y_2
…	…
x_n	y_n



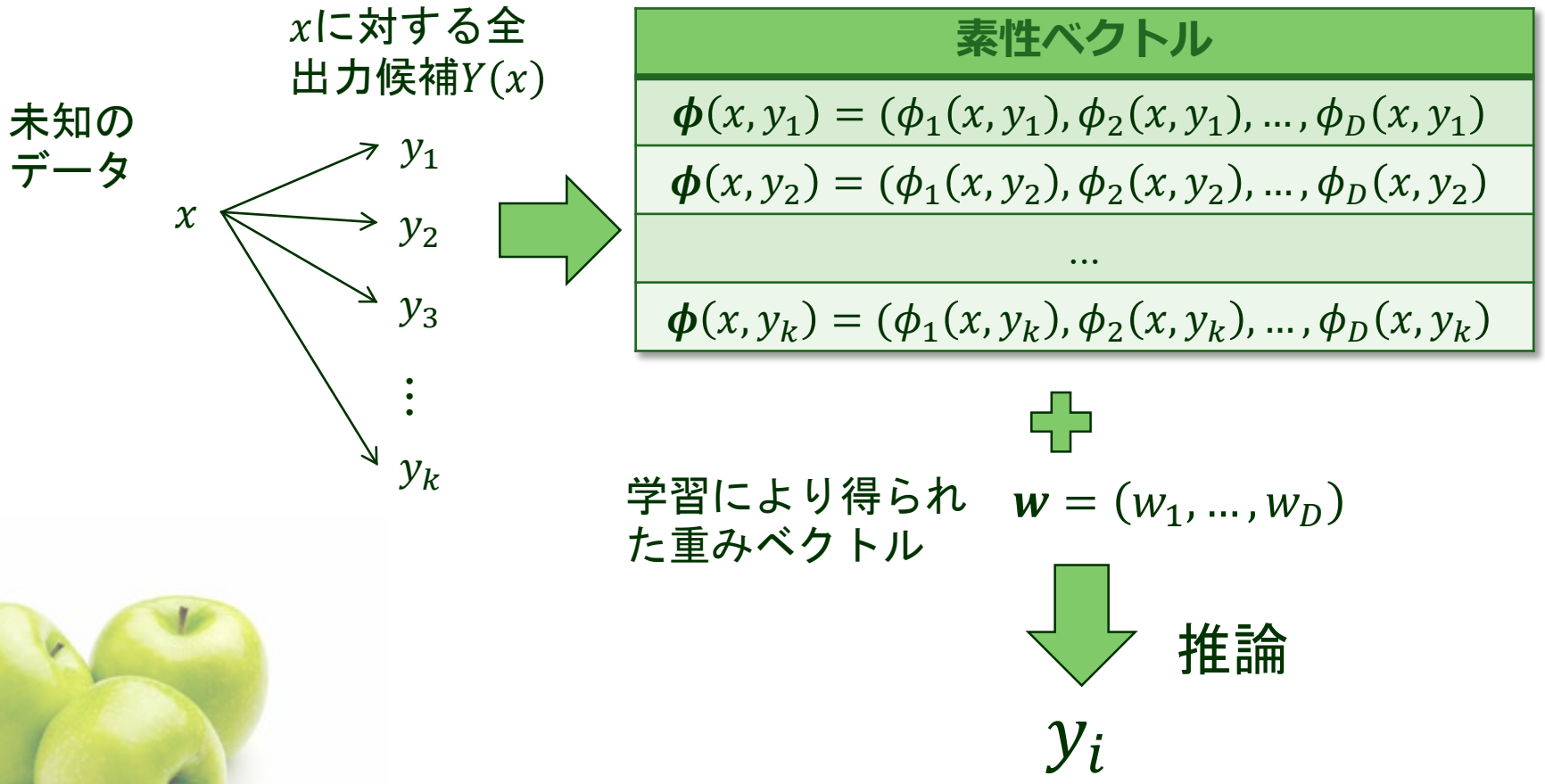
素性ベクトル
$\boldsymbol{\phi}(x_1, y_1) = (\phi_1(x_1, y_1), \phi_2(x_1, y_1), \ldots, \phi_D(x_1, y_1)$
$\boldsymbol{\phi}(x_2, y_2) = (\phi_1(x_2, y_2), \phi_2(x_2, y_2), \ldots, \phi_D(x_2, y_2)$
…
$\boldsymbol{\phi}(x_n, y_n) = (\phi_1(x_n, y_n), \phi_2(x_n, y_n), \ldots, \phi_D(x_n, y_n)$

学習

$$\boldsymbol{w} = (w_1, w_2, \ldots, w_D)$$

全体の流れ(2/2)

- **推論**
  - 未知のデータxに対する出力yの推定



確率的識別モデル

- **構造予測のための多クラスロジスティック回帰**

重みベクトル

$$p(y|x) = \frac{1}{Z(x)} \exp\left(\boldsymbol{w}^T \boldsymbol{\phi}(x, y)\right)$$

素性ベクトル

分配関数
(Partition function)

ただし、

$$Z(x) = \sum_{y' \in Y(x)} \exp\left(\boldsymbol{w}^T \boldsymbol{\phi}(x, y')\right)$$

パラメータ推定

- **勾配法を用いてパラメータを求める**
  - 訓練データに対する損失関数(負対数尤度)の勾配を求めれば良い

- **訓練データに対する損失関数(負対数尤度)**

$$-\log \prod_{i=1}^{n} p(y_i|x_i)$$

$$= -\sum_{i=1}^{n} \log p(y_i|x_i)$$

$$= -\sum_{i=1}^{n} \log \left\{ \frac{1}{Z(x_i)} \exp\left(\boldsymbol{w}^T \boldsymbol{\phi}(x_i, y_i)\right) \right\}$$

$$= \sum_{i=1}^{n} \log Z(x_i) - \sum_{i=1}^{n} \boldsymbol{w}^T \boldsymbol{\phi}(x_i, y_i)$$

$$\log ab = \log a + \log b$$
$$\log_e \exp(x) = \log_e e^x = x$$

パラメータ推定

- **損失関数**

$$L(\boldsymbol{w}) = \sum_{i=1}^{n} \log Z(x_i) - \sum_{i=1}^{n} \boldsymbol{w}^T \boldsymbol{\phi}(x_i, y_i)$$

- **勾配**

$$\frac{\partial L(\boldsymbol{w})}{\partial w_j} = -\sum_{i=1}^{n} \phi_j(x_i, y_i) + \sum_{i=1}^{n} \frac{1}{Z(x_i)} \frac{\partial Z(x_i)}{\partial w_j}$$

$$= -\sum_{i=1}^{n} \phi_j(x_i, y_i) + \sum_{i=1}^{n} \frac{1}{Z(x_i)} \sum_{y \in Y(x_i)} \phi_j(x_i, y) \exp\left(\boldsymbol{w}^T \boldsymbol{\phi}(x_i, y)\right)$$

$$= -\sum_{i=1}^{n} \phi_j(x_i, y_i) + \sum_{i=1}^{n} \sum_{y \in Y(x_i)} p(y|x_i) \phi_j(x_i, y)$$

$$\nabla L(\boldsymbol{w}) = \left(\frac{\partial L(\boldsymbol{w})}{\partial w_1}, \ldots, \frac{\partial L(\boldsymbol{w})}{\partial w_D} \right)$$

識別モデルのいいところ

- **独立性を仮定していない**
  - （戦略として）思いつく限りいろんな素性をいれる
  - 訓練データに対してより良い予測ができる
  - 逆にoverfittingする可能性がある
    - c.f. ガウス事前分布によるMAP推定でoverfittingを緩和

素性に関する注意

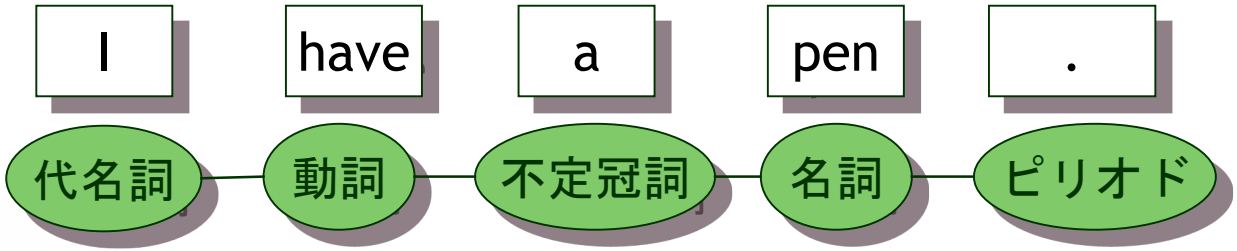
- **素性の組み合わせ**
  - 素性同士の共起情報が別素性として自動的に組み込まれるわけではない
    - 一つ前の単語が”the”で、今見ている単語が”flies”。一つ前の品詞が動詞で、今見ている品詞が冠詞など。
    - SVMでは多項式カーネルを用いるとこれらの組み合わせを自動的に計算したことになる
  - 素性の組み合わせを手で指示する。自動的に組み合わせる研究もある。c.f. 素性選択

系列解析のための確率的識別モデル

LINEAR-CHAIN CRF

系列解析のための確率的識別モデル
Linear-chain CRF

- **CRF (Conditional Random Fields, 条件付確率場)**
  - 構造予測のための確率的識別モデル$p(y|x)$
  - 出力yが無向グラフ
  - 素性を各ノードの近傍で定義
- **Linear-chain CRF**
  - グラフのノードが線形につながったCRF（系列解析のためのCRF）
  - 例えば、品詞解析の場合、入力xが文で、出力yが単語−品詞列となる。単語と品詞のペアがノードになる。素性を各ノードの近傍で定義する。



x ="I have a pen."
y ="I-代名詞 have-動詞 a-不定冠詞 pen-名詞 .-ピリオド"

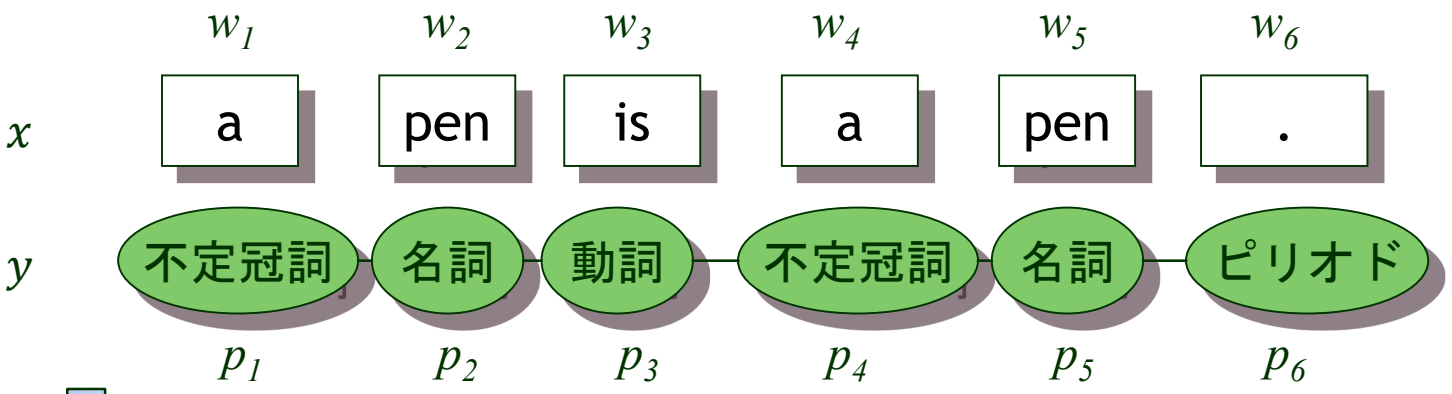
に対し$p(y|x)$を求める

Linear-chain CRF: 素性テンプレート

- **素性テンプレート**
  - どのような素性を採用するか表記したもの
  - 訓練データに対し素性テンプレートを適用し、実際に出現した値だけを素性として採用
- **素性テンプレートの例**
  - i番目のノードをc_iとする
  - i番目の品詞をp_iとする
  - i番目の単語をw_iとする
  - HMMと同様の素性の場合
    - 各ノードc_iに対し、$p_{i-1}p_i$ [最大で品詞数×品詞数の次元ができうる]
    - 各ノードc_iに対し、w_ip_i [最大で単語数×品詞数の次元ができうる]
  - 他にもいろんな組合せの素性を追加できる
    - 各ノードc_iに対し、$w_{i-1}w_ip_i$
    - 各ノードc_iに対し、$p_{i-2}p_{i-1}p_i$

Linear-chain CRF: 素性テンプレートと素性

- **素性テンプレートの例**
  - HMMと同様の素性の場合
    - 各ノードc_iに対し、$p_{i-1}p_i$
    - 各ノードc_iに対し、w_ip_i

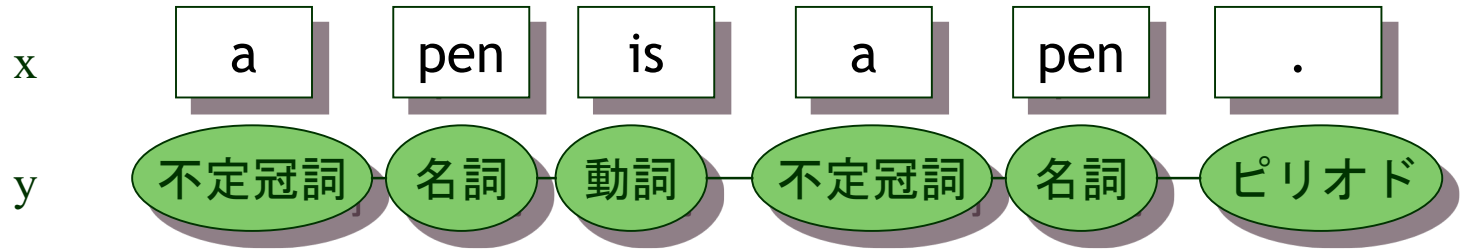


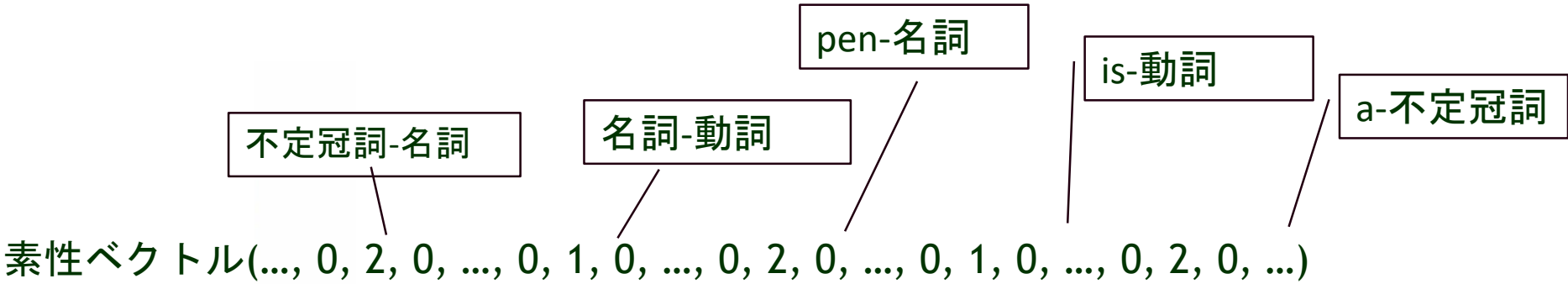
$p_{i-1}p_i$から生成される素性	w_ip_iから生成される素性
不定冠詞-名詞	a-不定冠詞
名詞-動詞	pen-名詞
動詞-不定冠詞	is-動詞
名詞-ピリオド	.-ピリオド

Linear-chain CRF: 全体の素性ベクトルと局所的素性ベクトルの関係

- 各素性関数の値は1文中に各素性が発火(出現)した回数

$$\phi_j(x,y) = \sum_c \phi_j(c) \quad (\forall j) \quad \Leftrightarrow \quad \boldsymbol{\phi}(x,y) = \sum_c \boldsymbol{\phi}(c)$$





Linear-chain CRF: 全体のスコアと局所的なスコアの関係

- **全体のスコア=各ノードcに対するスコアの積**

$$p(y|x) = \frac{1}{Z}\exp\left(\boldsymbol{w}^T\boldsymbol{\phi}(x,y)\right) = \frac{1}{Z}\exp\left(\boldsymbol{w}^T\sum_c \boldsymbol{\phi}(c)\right) = \frac{1}{Z}\exp\left(\sum_c \boldsymbol{w}^T\boldsymbol{\phi}(c)\right)$$

$$= \frac{1}{Z}\prod_c \exp\left(\boldsymbol{w}^T\boldsymbol{\phi}(c)\right)$$

文全体の素性ベクトル: (2,0,3,1,1) ↔ $e^{w_1\cdot 2+w_2\cdot 0+w_3\cdot 3+w_4\cdot 1+w_5\cdot 1}$

掛算

$e^{w_1\cdot 1+w_2\cdot 0+w_3\cdot 1+w_4\cdot 1+w_5\cdot 0}$　$e^{w_1\cdot 0+w_2\cdot 0+w_3\cdot 1+w_4\cdot 0+w_5\cdot 0}$　$e^{w_1\cdot 1+w_2\cdot 0+w_3\cdot 1+w_4\cdot 0+w_5\cdot 1}$

(1,0,1,1,0)　(0,0,1,0,0)　(1,0,1,0,1)

w_1　w_2　w_3

p_1　p_2　p_3

$$e^{a+b} = e^a e^b$$

Linear-chain CRF: スコアの分解とHMMとの対応

- **単語-品詞列の確率**

$$p(y|x) = \frac{1}{Z(x)} \exp\left(\boldsymbol{w}^T \boldsymbol{\phi}(x, y)\right) = \frac{1}{Z(x)} \prod_c \exp\left(\boldsymbol{w}^T \boldsymbol{\phi}(c)\right)$$

品詞列集合全体のスコアの和 (前向きアルゴリズムで計算)

HMMの遷移確率と出力確率に対応

- **解析: ビタビアルゴリズムの適用**
  - 遷移確率と出力確率 ⇔ ノードのスコア $\exp\left(\boldsymbol{w}^T \boldsymbol{\phi}(c)\right)$
- **学習: 前向き後向きアルゴリズムの適用**
  - 遷移回数と出力回数 ⇔ 素性値(素性の発火回数)

1次のLinear-chain CRF

- ラベル集合をQ、入力をx、出力を$q_1 q_2 \dots q_T$とする。
- Linear-chain CRF

$$p(q_1 \cdots q_T | x) = \frac{1}{Z(x)} \prod_{t=1}^{T} \exp\{\boldsymbol{w}^T \boldsymbol{\phi}(q_{t-1}, q_t, x, t)\}$$

$$Z(x) = \sum_{q'_1 \in Q, \cdots, q'_T \in Q} \prod_{t=1}^{T} \exp\{\boldsymbol{w}^T \boldsymbol{\phi}(q'_{t-1}, q'_t, x, t)\}$$

ただし、$Z(x)$は分配関数、$\boldsymbol{w}$は$\boldsymbol{\phi}$に対する重み、$q_0 = DUMMY$(ダミーのラベル)とし、素性関数ϕの引数は次のように定義される。

$$\phi(\text{時刻}t-1\text{のラベル}, \text{時刻}t\text{のラベル}, \text{入力}, \text{時刻})$$

- 入力(第三引数)と時刻(第四引数)を与えることによって、入力の情報を素性として用いることができる
  - 例えば、品詞解析の場合、ラベルが品詞であり、入力が単語列となり、時刻tやその前後における単語を素性として用いることができる。

まとめ

- **生成モデルと識別モデル**
- **確率的識別モデル**
  - 構造予測のための多クラスロジスティック回帰
  - 条件付き確率場（CRF）
  - Linear-chain CRF

- **今までの講義資料**

http://aiweb.cs.ehime-u.ac.jp/~ninomiya/iips/