言語処理における識別モデルの発展
- HMMからCRFまで -

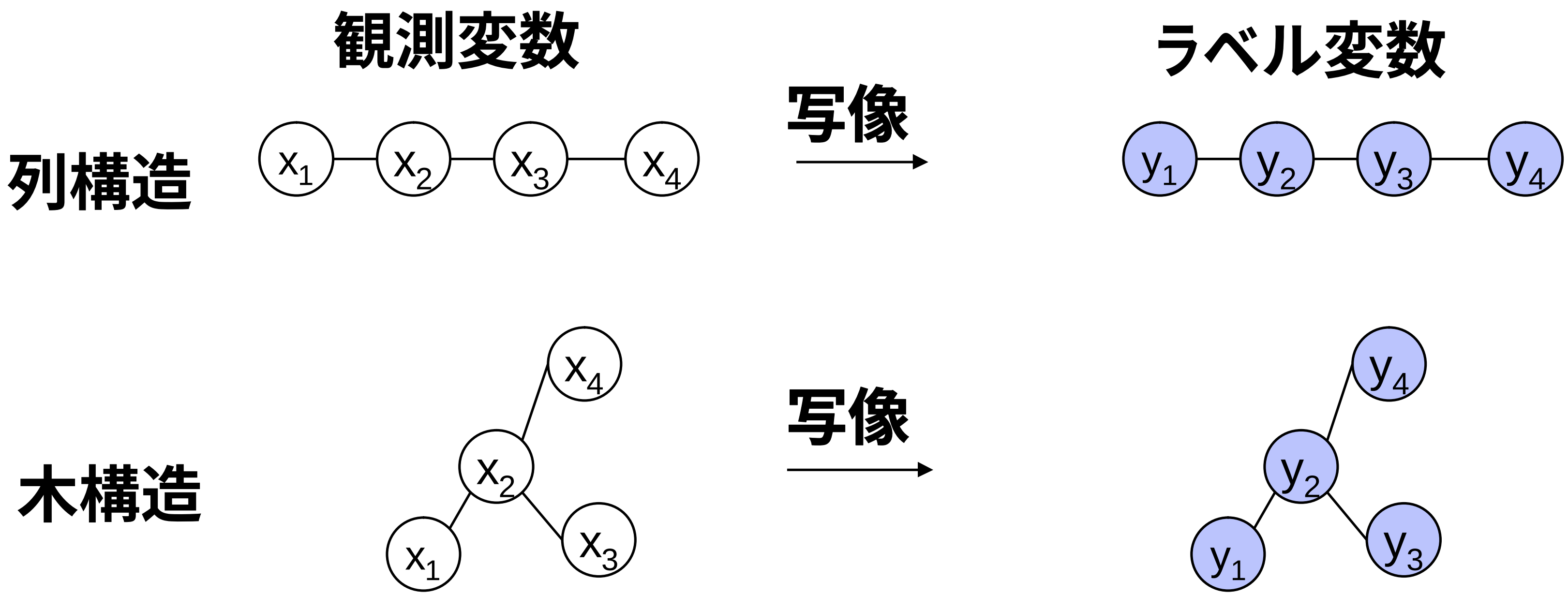
坪井祐太, 鹿島久嗣 (IBM 東京基礎研究所)
工藤 拓 (Google)

チュートリアルの流れ

- 構造のラベル付け問題とは（鹿島）
- 2つのアプローチとその比較
  - 生成モデル：隠れマルコフモデル（鹿島）
  - 識別モデル：条件付確率場（坪井）
- そのほかの識別モデル（坪井）
- 計算機実験による性能比較（工藤）
- 利用可能なツール（工藤）

構造ラベル付与問題とは？

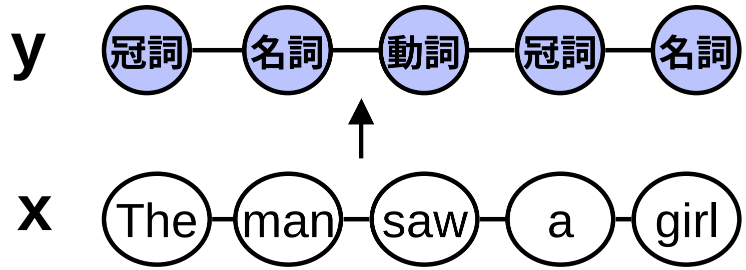
- 観測された構造データ x に対応するラベル y への写像をおこなう問題



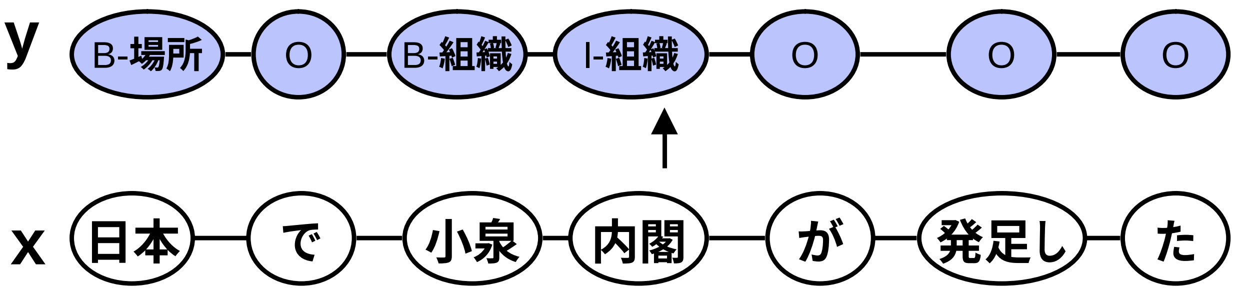
  - 自然言語処理、バイオインフォマティックスにおいて、数多くみられる

構造ラベル付与問題の例 (列構造)

- 品詞タグ付与タスク
  - 単語列に対して品詞ラベル (動詞、名詞…) を付与するタスク

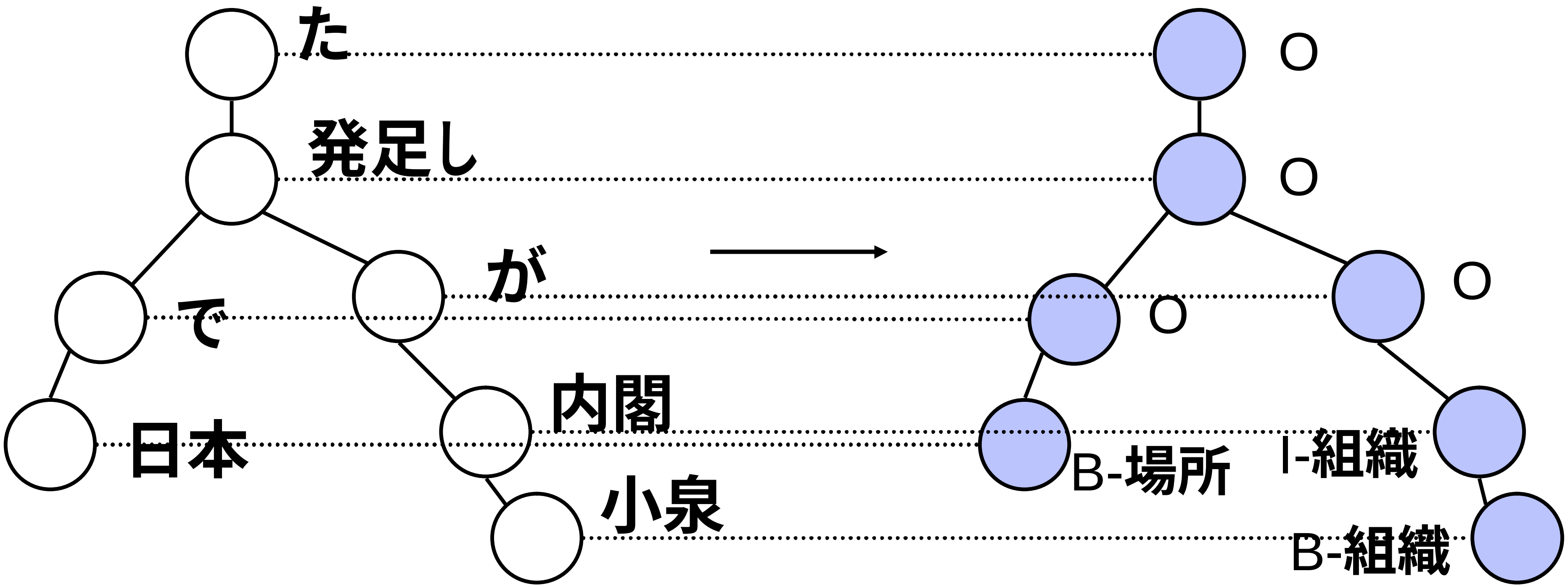


- 固有表現抽出タスク
  - 人名・組織名等の固有表現をテキスト中から抽出するタスク
  - 単語列に対して固有表現の「始まり (B-XXX)」と「続く (I-XXX)」、「それ以外 (O)」を示すラベルを付与



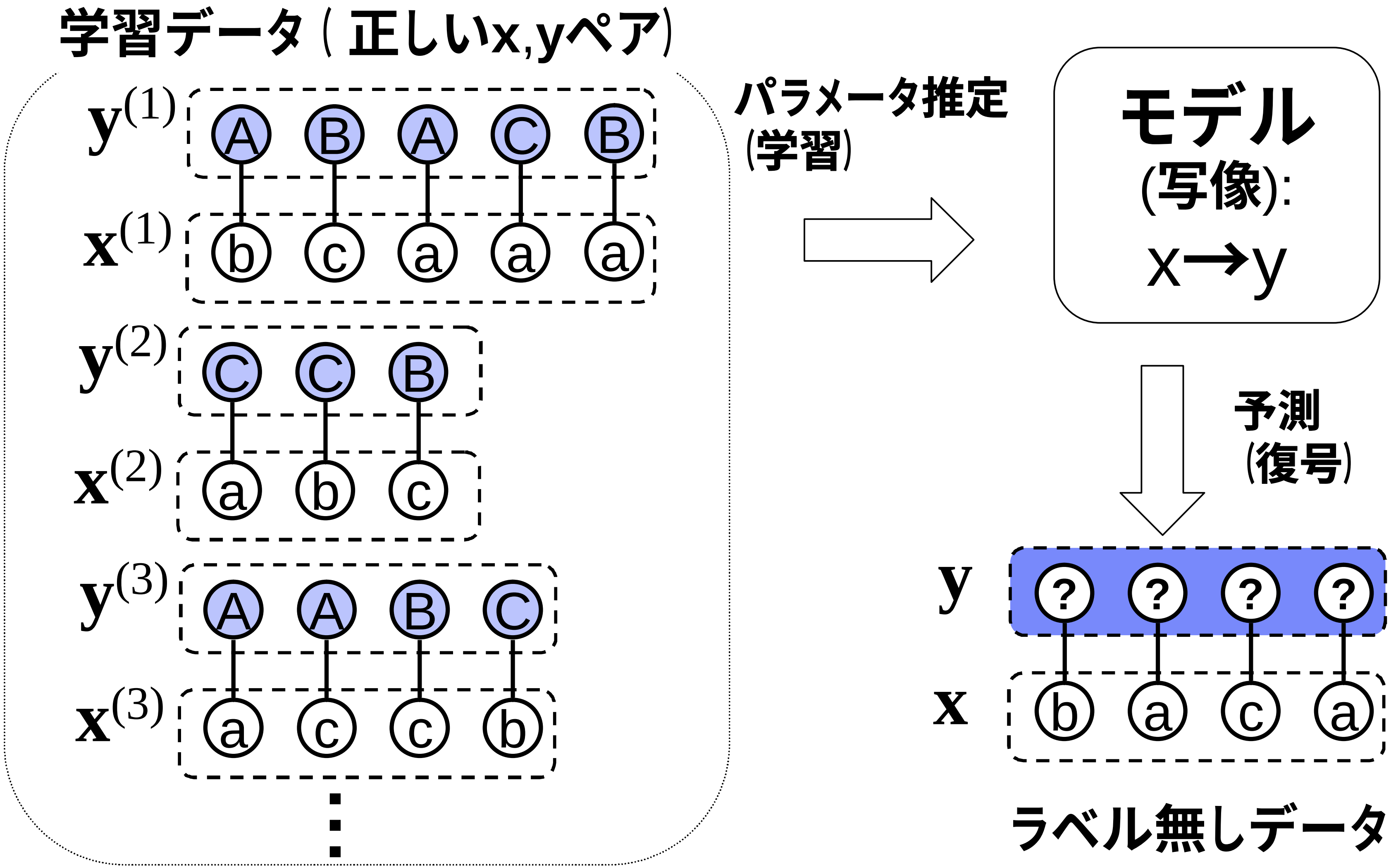
構造ラベル付与問題の例 (木構造)

- 係り受け木に対する固有表現抽出タスク
  - 係り受け解析によって生成された、単語間の関係を表す係り受け木に対して、ラベルを付与
  - 主語と述語の関係など言語構造を考慮



教師付き学習によるアプローチ

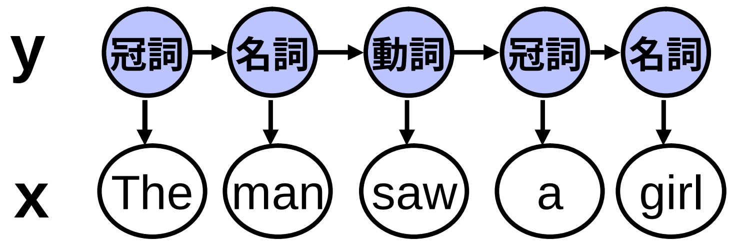
- 過去に正しいラベル付けを行ったデータをもとにモデルを学習



教師つき学習による構造ラベル付与学習モデルへのアプローチ

- 生成モデルに基づく手法
  – 隠れマルコフモデル
- 識別モデルに基づく手法
  – 条件付確率場

生成モデルによる構造ラベル付与学習
隠れマルコフモデル(HMM)

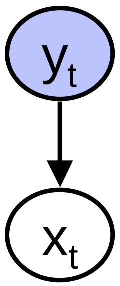


- **xとyの同時分布に基づくモデル**
- **同時分布を出力確率と遷移確率に分解してモデル化**

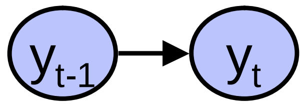
$$\mathrm{P}(\mathbf{x},\mathbf{y}) = \mathrm{P}(\mathbf{x}\,|\,\mathbf{y})\,\mathrm{P}(\mathbf{y})$$

$$= \prod_{t=1}^{T} \mathrm{P}(x_t|y_t)\mathrm{P}(y_t\,|\,y_{t-1})$$

(Tは構造x, yのサイズ)



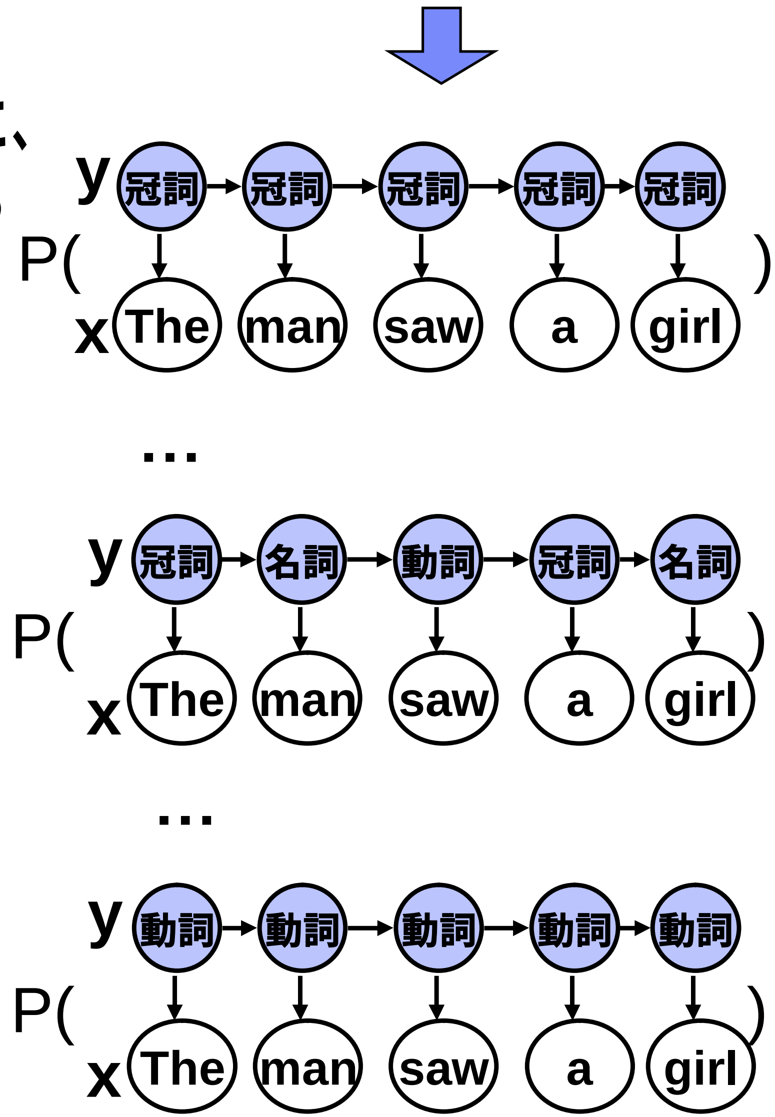
出力確率
(ある位置の観測変数は、同じ位置のラベル変数によって決定)



遷移確率
(ある位置のラベル変数は、ひとつ前のラベル変数によって決定)

HMMでのラベル列の予測（復号問題）

全部で$|\Sigma_y|^T$個の候補



- 予測（復号）：観測xが与えられたときに、確率が最大になるラベル列yを見つける

$$\mathbf{y}^{predict} = \operatorname*{argmax}_{\mathbf{y}\in\Sigma_y^{T}} \mathrm{P}(\mathbf{y}\,|\,\mathbf{x}) = \operatorname*{argmax}_{\mathbf{y}\in\Sigma_y^{T}} \mathrm{P}(\mathbf{x},\mathbf{y})$$

$$= \operatorname*{argmax}_{\mathbf{y}\in\Sigma_y^{T}} \prod_{t=1}^{T} \mathrm{P}(x_t|y_t)\mathrm{P}(y_t\,|\,y_{t-1})$$

- しかし、ありうるラベル列全て($|\Sigma_y|^T$個)の列挙は不可能⇒ 計算の工夫が必要

（Σ_yは、ラベル変数の取りうる値の集合）

Viterbi 復号法による最適ラベル列の求め方（1／3）

- テーブル $\delta_t(y_t)$を使い、最適なラベル列を再帰的に計算
  - $\delta_t(y_t)$ ：位置 t でラベルy_t をとる、t までのラベル列の確率の最大値

$$\delta_t(y_t) = \max_{y_1,\ldots,y_{t-1}\in\Sigma^{t-1}} \prod_{\tau=1}^{t-1} P(y_\tau \mid y_{\tau-1}) P(x_\tau \mid y_\tau) \qquad \delta_1(y_1) = \mathrm{P}(x_1 \mid y_1)$$

$$= \max_{y_{t-1}\in\Sigma_y} \delta_{t-1}(y_{t-1}) P(y_t \mid y_{t-1}) P(x_t \mid y_t)$$

x： 観測変数

英語品詞タグ付けタスクでのδ

y：ラベル変数

y \ x	the	man	saw	…
冠詞	$\mathrm{P}(\text{the} \mid \text{冠詞})$	$\max_{y_1\in\Sigma_y} \delta_1(y_1) \times \mathrm{P}(\text{冠詞} \mid y_1)\,\mathrm{P}(\text{man} \mid \text{冠詞})$	$\max_{y_2\in\Sigma_y} \delta_2(y_2) \times \mathrm{P}(\text{冠詞} \mid y_2)\,\mathrm{P}(\text{man} \mid \text{冠詞})$	…
名詞	$\mathrm{P}(\text{the} \mid \text{名詞})$	$\max_{y_1\in\Sigma_y} \delta_1(y_1) \times \mathrm{P}(\text{名詞} \mid y_1)\,\mathrm{P}(\text{man} \mid \text{名詞})$	$\max_{y_2\in\Sigma_y} \delta_2(y_2) \times \mathrm{P}(\text{名詞} \mid y_2)\,\mathrm{P}(\text{man} \mid \text{名詞})$	…
動詞	$\mathrm{P}(\text{the} \mid \text{動詞})$	$\max_{y_1\in\Sigma_y} \delta_1(y_1) \times \mathrm{P}(\text{動詞} \mid y_1)\,\mathrm{P}(\text{man} \mid \text{動詞})$	$\max_{y_2\in\Sigma_y} \delta_2(y_2) \times \mathrm{P}(\text{動詞} \mid y_2)\,\mathrm{P}(\text{man} \mid \text{動詞})$	…

Viterbi 復号法による最適ラベル列の求め方（2／3）

- 具体例： δ_tを最大にするy_{t-1}からy_tへの遷移（矢印）が決まった時のδテーブル

y \ x	the	man	saw	…
冠詞	P(the | 冠詞)	P(the | 動詞) ×P(冠詞 | 動詞) P(man | 冠詞)	P(the | 名詞) ×P(動詞 | 名詞) P(man | 動詞) ×P(冠詞 | 動詞) P(saw | 冠詞)	…
名詞	P(the | 名詞)	P(the | 冠詞) ×P(名詞 | 冠詞) P(man | 名詞)	P(the | 名詞) ×P(動詞 | 名詞) P(man | 動詞) ×P(名詞 | 動詞) P(saw | 名詞)	…
動詞	P(the | 動詞)	P(the | 名詞) ×P(動詞 | 名詞) P(man | 動詞)	P(the | 冠詞) ×P(名詞 | 冠詞) P(man | 名詞) ×P(動詞 | 名詞) P(saw | 動詞)	…

Viterbi 復号法による最適ラベル列の求め方（3／3）

- **δテーブルを用いて最大確率がもとまったとする ⇒ 本当に欲しいのは最大確率を実現するラベル列**
- $\delta_t(y) = \max_{y_{t-1}\in\Sigma_y} \delta_{t-1}(y_{t-1})P(y_t \mid y_{t-1})P(x_t \mid y_t)$ **を求めたときに、max を実現するラベル y_{t-1} を記憶するテーブル π も同時に計算しておく**

$$\pi_t(y_t) = \operatorname*{argmax}_{y_{t-1}\in\Sigma_y} \delta_{t-1}(y_{t-1})P(y_t \mid y_{t-1})P(x_t \mid y_t)$$

- **$\delta_T(y_T)$が最大になる$\pi_T(y_T)$ からバックトラックすることで、確率が最大になるラベル列を得ることができる。**

y \ x	the	man	saw	…
冠詞		**π_2(冠詞)=動詞**	**π_3(冠詞)=動詞**	…
名詞		**π_2(名詞)=冠詞**	**π_3(名詞)=動詞**	…
動詞		**π_2(動詞)=名詞**	**π_3(動詞)=名詞**	…

隠れマルコフモデルのパラメータ推定 (学習)

- 推定するべきパラメータθ
  - 出力確率　$P(x_t|y_t)$
  - 遷移確率　$P(y_t|y_{t-1})$
- 最尤推定: 学習データをもっとも再現するパラメータを求める

$$\hat{\boldsymbol{\theta}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_i \mathrm{P}_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$$

(iは学習データの索引)

$$= \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_i \prod_t^{T^{(i)}} \mathrm{P}_\theta(x_t^{(i)} \mid y_t^{(i)}) \mathrm{P}_\theta(y_t^{(i)} \mid y_{t-1}^{(i)})$$

- 最尤パラメータは学習データ内での出力および遷移の頻度のカウントで計算可能
  - AのあとにAが2回、Bが3回出現していたら、P(A|A)=2/5、P(B|A)=3/5

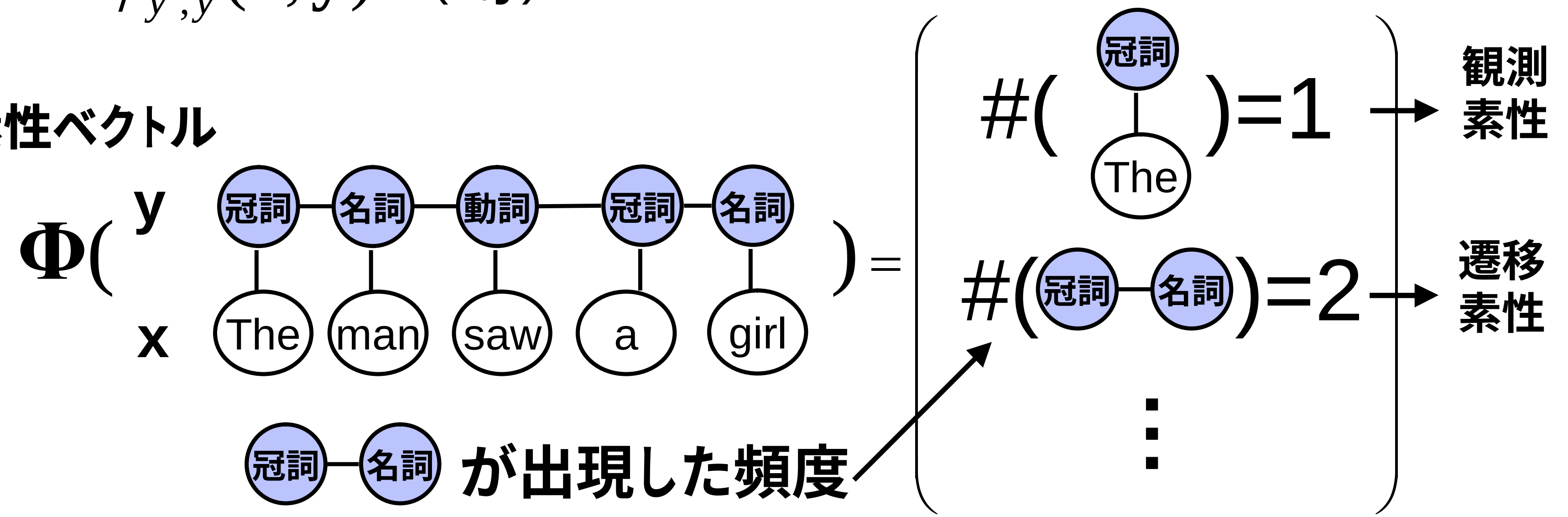
隠れマルコフモデル(HMM)の素性ベクトル表現（1／2）

- HMMは、パラメータベクトルと素性ベクトルの内積の形でかける

$$\log \mathrm{P}(\mathbf{x},\mathbf{y}) = \langle \Theta, \Phi(\mathbf{x},\mathbf{y}) \rangle$$
$$= \sum_{x\in\Sigma_x}\sum_{y\in\Sigma_y} \phi_{x,y}(\mathbf{x},\mathbf{y}) \log \mathrm{P}(x|y) + \sum_{y'\in\Sigma_y}\sum_{y\in\Sigma_y} \phi_{y',y}(\mathbf{x},\mathbf{y}) \log \mathrm{P}(y'|y)$$

  - $\phi_{x,y}(\mathbf{x},\mathbf{y})$：(x,y)におけるある「ラベル-観測」出力の出現回数
  - $\phi_{y',y}(\mathbf{x},\mathbf{y})$：(x,y)におけるある「ラベル-ラベル」遷移の出現回数

素性ベクトル



隠れマルコフモデル(HMM)の素性ベクトル表現

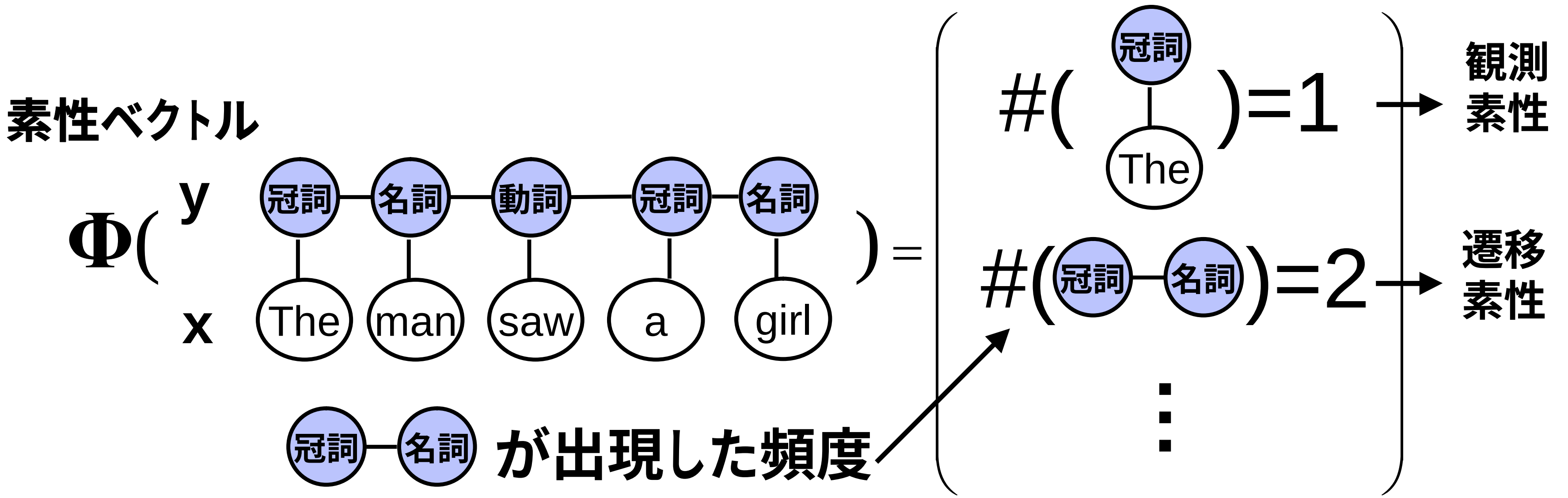
$$\log \mathrm{P}(\mathbf{x},\mathbf{y}) = \langle \Theta, \Phi(\mathbf{x},\mathbf{y}) \rangle$$

- 予測：パラメータベクトルとの内積が最も大きい素性ベクトルを求める

$$\mathbf{y}^{predict} = \underset{\mathbf{y}\in\Sigma_y{}^T}{\operatorname{argmax}} \langle \Theta, \Phi(\mathbf{x},\mathbf{y}) \rangle$$

- 学習：素性ベクトルとの内積が最も大きいパラメータベクトルを求める

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \prod_i \langle \Theta, \Phi(\mathbf{x}^{(i)},\mathbf{y}^{(i)}) \rangle$$



隠れマルコフモデルの問題点

- 同時分布を推定しようとしている
- 素性の独立性を仮定している

隠れマルコフモデルの2つの問題点：(1) 同時分布の推定

- HMMは、同時分布 $\mathrm{P}(\mathbf{x},\mathbf{y})$ の最尤推定をすることで、間接的に予測問題を解いている
  - 最尤推定 $\hat{\boldsymbol{\theta}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_i \mathrm{P}_{\boldsymbol{\theta}}(\mathbf{x}^{(i)},\mathbf{y}^{(i)})$
  - 予測 $\overset{predict}{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}\in\Sigma_y} \mathrm{P}(\mathbf{x},\mathbf{y})$
- ホントは、条件付分布 $\mathrm{P}(\mathbf{y}\mid\mathbf{x})$ がわかれば予測はできるはず
  - あるべき最尤推定？

$$\hat{\boldsymbol{\theta}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_i \mathrm{P}_{\boldsymbol{\theta}}(\mathbf{y}^{(i)}\mid\mathbf{x}^{(i)})$$

隠れマルコフモデルの2つの問題点：(2) 素性の独立性

- 隠れマルコフモデルが仮定している確率的な制約

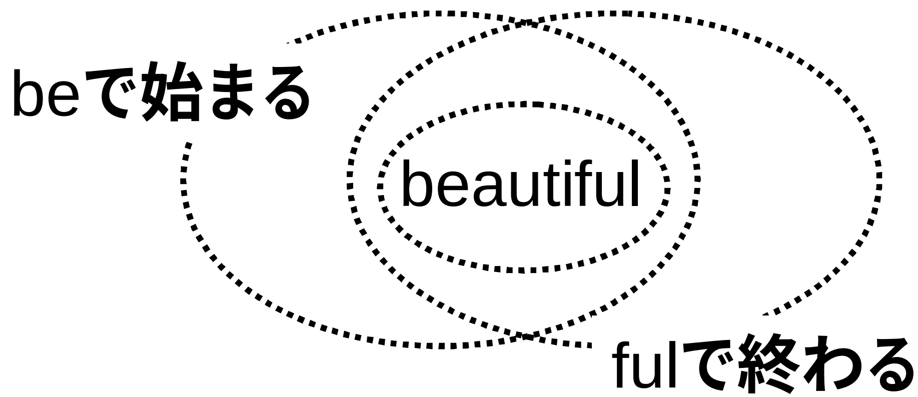
$$\sum_{y_t \in \Sigma_y} \mathrm{P}(y_t \mid y_{t-1}) = 1 \qquad \sum_{x_t \in \Sigma_x} \mathrm{P}(x_t \mid y_t) = 1$$

の意味するところ ＝ 素性の独立性

- – ある $y_t \in \Sigma_y$ と $y'_t \in \Sigma_y$ が同時に起こることはない

⇒ 独立でない素性をうまく扱えない

- 一方、自然言語処理では単語それ自身以外に、単語の部分文字列等が素性に使われることが多い
  - – 品詞タグ付け
    - P(beautiful|形容詞)
    - P(fulで終わる単語|形容詞)
    - P(beで始まる単語|形容詞)



隠れマルコフモデル から 条件付確率場へ

- 同時分布を推定しようとしている
- 素性の独立性を仮定している

を解決するのが、条件付確率場（CRF）

識別モデル

- xからyを直接推定するモデル
- 識別モデルの利点
  - 直接分類問題を解くことが出来る
  - 素性間の重なりを考慮して重みを学習
- 多クラスのロジスティック回帰モデル (最大エントロピーモデル)
  - 確率分布 (条件付分布) の形をした識別モデル

0以上になるように

(x, y)ペアのスコア

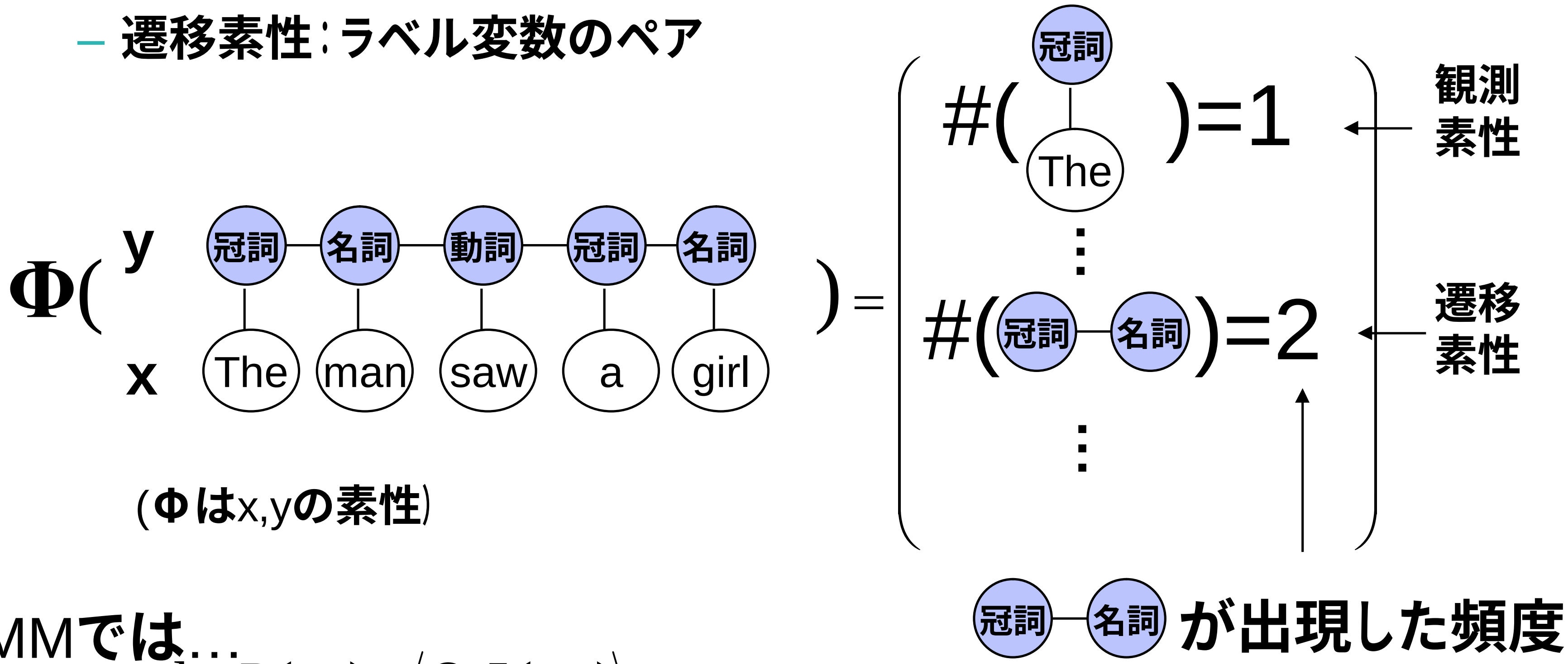
足して1になるように全てのyの和で割る

$$P(y \mid \mathbf{x}) = \frac{\exp(\langle \mathbf{\Theta}, \mathbf{\Phi}(\mathbf{x}, y) \rangle)}{\sum_{\tilde{y} \in \mathbf{Y}} \exp(\langle \mathbf{\Theta}, \mathbf{\Phi}(\mathbf{x}, \tilde{y}) \rangle)}$$

(Φはx,yの素性、Θは素性に対する重み, <a,b>は内積)

識別モデルによる構造ラベル付与学習：
条件付確率場 (Conditional Random Fields: CRF)

- ロジスティック回帰モデルを基に、ローカルな変数間の関係を素性(遷移素性)で表現したモデル
- 素性ベクトルの各要素は、素性が構造中に出現した頻度
  - 観測素性：観測変数とラベル変数のペア
  - 遷移素性：ラベル変数のペア



HMMでは…

$$\log P(\mathbf{x},\mathbf{y}) = \langle \Theta, \Phi(\mathbf{x},\mathbf{y}) \rangle$$

識別モデルによる構造ラベル付与学習：
条件付確率場 (Conditional Random Fields: CRF)

- ロジスティック回帰モデルを基に、ローカルな変数間の関係を素性(遷移素性)で表現したモデル

$$P(\mathbf{y} \mid \mathbf{x}) = \frac{\exp\left(\left\langle \boldsymbol{\Theta}, \boldsymbol{\Phi}(\mathbf{x}, \mathbf{y}) \right\rangle\right)}{\sum_{\tilde{\mathbf{y}}} \exp\left(\left\langle \boldsymbol{\Theta}, \boldsymbol{\Phi}(\mathbf{x}, \tilde{\mathbf{y}}) \right\rangle\right)}$$

列全体のスコア

$$= \frac{\exp\left(\sum_{t=1}^{T} \left\langle \boldsymbol{\Theta}, \boldsymbol{\Phi}(\mathbf{x}, \mathbf{y}_t^{t+1}) \right\rangle\right)}{\sum_{\tilde{\mathbf{y}}} \exp\left(\sum_{\tau=1}^{T} \left\langle \boldsymbol{\Theta}, \boldsymbol{\Phi}(\mathbf{x}, \tilde{\mathbf{y}}_\tau^{\tau+1}) \right\rangle\right)}, \quad \mathbf{y}_t^{t+1} = (y_t, y_{t+1})$$

ある点tのスコア

(Φはx,yの素性、θは素性に対する重み)

条件付確率場の予測 (復号問題)

- 隠れマルコフモデルと同じくViterbi法で行う

$$(\mathbf{y}_t^{t+1} = (y_t, y_{t+1}))$$

$$\arg\max_{\mathbf{y}} \log \mathrm{P}(\mathbf{y} \mid \mathbf{x}) = \arg\max_{\mathbf{y}} \log \frac{\exp\left(\sum_{t=1}^{T} \left\langle \mathbf{\Theta}, \mathbf{\Phi}(\mathbf{x}, \mathbf{y}_t^{t+1}) \right\rangle\right)}{\sum_{\tilde{\mathbf{y}}} \exp\left(\sum_{\tau=1}^{T} \left\langle \mathbf{\Theta}, \mathbf{\Phi}(\mathbf{x}, \tilde{\mathbf{y}}_\tau^{\tau+1}) \right\rangle\right)}$$

yに依らないので
省略

$$= \arg\max_{\mathbf{y}} \sum_{t=1}^{T} \left\langle \mathbf{\Theta}, \mathbf{\Phi}(\mathbf{x}, \mathbf{y}_t^{t+1}) \right\rangle$$

logでexpが消える

⇒ $\sum_{\tau=1}^{T} \left\langle \theta, \phi(\mathbf{x}, \mathbf{y}_\tau^{\tau+1}) \right\rangle$ が最大になるyを求めればよい

⇒ スコア最大 (コスト最小化)法

条件付確率場のパラメータ推定（学習）の概要

- 山登り法による最尤推定（関数の最大値探索問題）

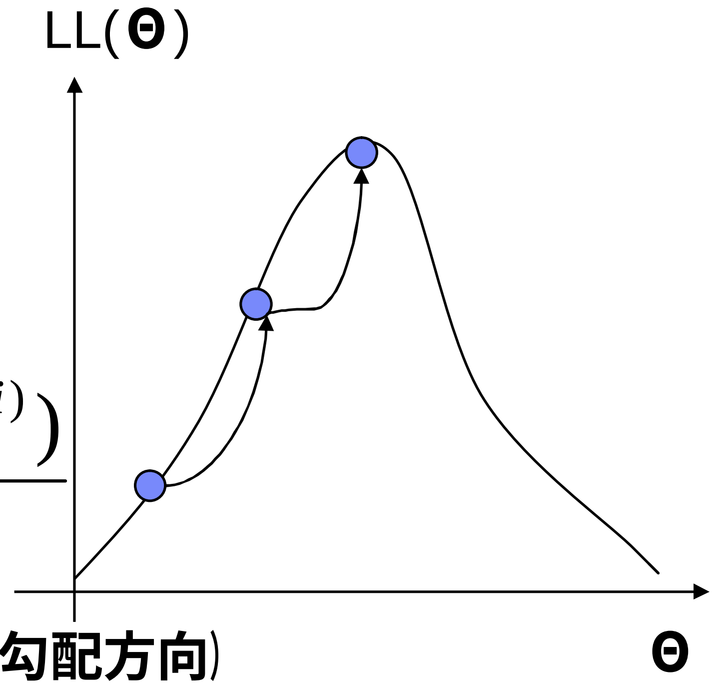
1. 現在のパラメータの対数尤度(LL)の計算

$$LL(\boldsymbol{\Theta}) = \sum_{i \in \text{training data}} \log \mathrm{P}_{\boldsymbol{\Theta}}(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)})$$

2. 偏微分(勾配)の計算　（0なら終了）

$$\frac{LL(\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} = \sum_{i \in \text{training data}} \frac{\log \mathrm{P}_{\boldsymbol{\Theta}}(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)})}{\partial \boldsymbol{\Theta}}$$

3. 勾配方向へパラメータの更新（たとえば、最大勾配方向）して、1に戻る



CRFでの計算の特徴：
ステップ1,2で動的計画法で効率的に計算する必要がある

条件付確率場の最適化法の速度比較 (Sha et al, 2003)

- タスク：NP Chunking (82万素性を使用)
- 各手法によるCRFの収束時間の比較
  - Preconditioned conjugate-gradient (Precond. CG)
  - Mixed conjugate-gradient (Mixed CG)
  - Conjugate-gradient (Plain CG)
  - Limited-memory quasi-Newton (L-BFGS)
  - Generalized iterative scaling (GIS)
- 一般的な最適化手法が速い

対数尤度

training method	time	F score	$\mathcal{L}'_\lambda$
Precond. CG	130	94.19%	-2968
Mixed CG	540	94.20%	-2990
Plain CG	648	94.04%	-2967
L-BFGS	84	94.19%	-2948
GIS	3700	93.55%	-5668

収束が速いため、良く使用される

条件付確率場のパラメータ推定 (対数尤度の計算1)

- **対数尤度の計算**

$$\sum_{i \in \text{training data}} \log \mathrm{P}(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)}) = \sum_{i} \log \frac{\exp(\langle \boldsymbol{\Theta}, \boldsymbol{\Phi}(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \rangle)}{\sum_{\mathbf{y} \in \mathbf{Y}} \exp(\langle \boldsymbol{\Theta}, \boldsymbol{\Phi}(\mathbf{x}^{(i)}, \tilde{\mathbf{y}}) \rangle)}$$

$$= \sum_{i} \left(\boxed{\sum_{t}^{T} \langle \boldsymbol{\Theta}, \boldsymbol{\Phi}(\mathbf{x}^{(i)}, \mathbf{y}_t^{t+1(i)})} \rangle - \log \boxed{\sum_{\tilde{\mathbf{y}} \in \mathbf{Y}} \exp(\sum_{\tau}^{T} \langle \boldsymbol{\Theta}, \boldsymbol{\Phi}(\mathbf{x}^{(i)}, \tilde{\mathbf{y}}_\tau^{\tau+1}) \rangle)} \right)$$

Θでの$(x^{(i)},y^{(i)})$のスコア

Z: Θでのあらゆるyでのスコア合計

→動的計画法で効率的に計算

条件付確率場のパラメータ推定 (対数尤度の計算2)

- 再帰式 (Forward アルゴリズム)

$$\alpha_t(y) = \sum_{y' \in Y} \left(\alpha_{t-1}(y') \cdot \exp\left(\left\langle \boldsymbol{\Theta}, \boldsymbol{\Phi}(\mathbf{x}^{(i)}, y', y) \right\rangle \right) \right)$$

⇦ |Y|×|T|の2次元配列に記憶

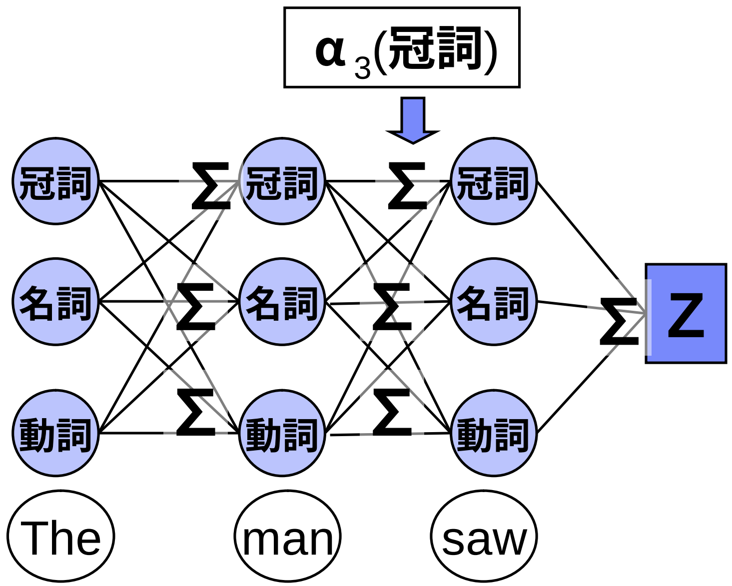
= 位置tがラベルyになるy_1^tまでの全てのパスのスコアの合計

$$Z = \sum_{\tilde{\mathbf{y}} \in \mathbf{Y}} \exp\left(\sum_{\tau}^{T} \left\langle \boldsymbol{\Theta}, \boldsymbol{\Phi}(\mathbf{x}^{(i)}, \tilde{\mathbf{y}}_{\tau}^{\tau+1}) \right\rangle \right)$$

$$= \sum_{y \in Y} \alpha_T(y)$$

Viterbi復号法 →max

Forward アルゴリズム →sum



条件付確率場のパラメータ推定 (偏微分の計算1)

- 尤度最大化の方向の計算

$$\sum_{i \in \text{training data}} \frac{\log P_{\boldsymbol{\Theta}}(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)})}{\partial \boldsymbol{\Theta}} = \sum_i \left(\boldsymbol{\Phi}(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) - \sum_{\tilde{\mathbf{y}} \in \mathbf{Y}} P(\tilde{\mathbf{y}} \mid \mathbf{x}^{(i)}) \boldsymbol{\Phi}(\mathbf{x}^{(i)}, \tilde{\mathbf{y}}) \right)$$

素性の出現頻度

モデルでの素性期待出現頻度

$$= \sum_i \sum_t^T \left(\boldsymbol{\Phi}(\mathbf{x}^{(i)}, \mathbf{y}_t^{t+1(i)}) - \boxed{\sum_{y', y'' \in Y} \sum_{\tilde{\mathbf{y}} : \tilde{y}_t = y', \tilde{y}_{t+1} = y''} P(\tilde{\mathbf{y}} \mid \mathbf{x}^{(i)}) \boldsymbol{\Phi}(\mathbf{x}^{(i)}, \tilde{\mathbf{y}}_t^{t+1})} \right)$$

⇦ 位置t毎の計算

位置tのラベルがy’、t+1のラベルがy”の周辺期待値

→動的計画法で効率的に計算

条件付確率場のパラメータ推定 (偏微分の計算2)

- **再帰式(Forward-Backwardアルゴリズム)**

$$\alpha_t(y) = \sum_{y'\in Y}\left(\alpha_{t-1}(y')\cdot\exp(\langle\boldsymbol{\Theta},\boldsymbol{\Phi}(\mathbf{x}^{(i)},y',y)\rangle)\right)$$

$$\beta_t(y) = \sum_{y'\in Y}\left(\exp(\langle\boldsymbol{\Theta},\boldsymbol{\Phi}(\mathbf{x}^{(i)},y,y')\rangle)\cdot\beta_{t+1}(y')\right)$$

「対数尤度の計算2」で出て来ました。

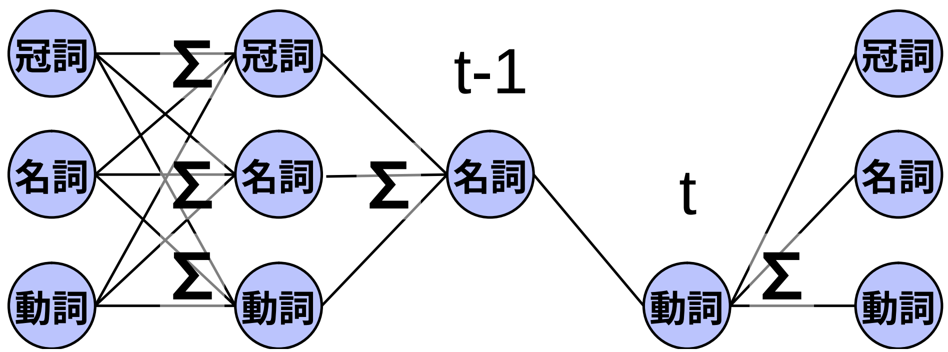
= **位置tがラベルyになるy_t^Tまでの全てのパスのスコアの合計**

周辺期待値= $$\sum_{y',y''\in Y}\ \sum_{\tilde{\mathbf{y}}:\tilde{y}_t=y',\tilde{y}_{t+1}=y''} \mathrm{P}(\tilde{\mathbf{y}}\mid\mathbf{x}^{(i)})\boldsymbol{\Phi}(\mathbf{x}^{(i)},\tilde{\mathbf{y}}_t^{t+1})$$

$$=\alpha_{t-1}(y')\cdot\exp(\langle\boldsymbol{\Theta},\boldsymbol{\Phi}(\mathbf{x}^{(i)},y',y'')\rangle)\cdot\beta_t(y'')$$

y’=**名詞**

y’’=**動詞**



条件付確率場のパラメータ推定のまとめ

- 山登り法による最尤推定
  - 一般的な数値計算手法が利用可能
- 動的計画法で効率的に尤度と偏微分の計算
  - 尤度計算
    - Forwardアルゴリズム
  - 尤度の偏微分計算
    - Forward- Backwardアルゴリズム

まとめ：隠れマルコフモデル(HMM)と条件付確率場(CRF)

	隠れマルコフモデル	条件付確率場
確率モデル	P(Y, X)	P(Y|X)
定式化	生成モデル	識別モデル
学習	P(Y,X)の最尤推定 計算は容易で高速	P(Y|X)の最尤推定 計算に工夫が必要
構造	列構造、木構造、DAG	列構造、木構造、DAG
柔軟な素性設計	困難 (独立性を仮定)	可能
予測モデル	Viterbi復号法	Viterbi復号法
言語モデルとしての使用	可能	不可能

その他の識別モデル: 分類手法に基づくアプローチ

- **CRFのパラメタ推定の別アプローチ**
  - 分類手法 (パーセプトロン, SVM, …) にもとづいたアプローチ
- **構造学習として何がうれしいか？→ 通常のアプローチでは解けない問題に適用できる**

そのほかの識別モデル：隠れマルコフパーセプトロン

- CRFの推定は、訓練データの $y^{(i)}$ が出力される確率を「最大にする」ように学習される

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \prod_{i \in \text{trainingdata}} \mathrm{P}_{\boldsymbol{\theta}}(y^{(i)} \mid \mathbf{x}^{(i)})$$

- 別の考え方：訓練データの $y^{(i)}$ が出力される確率が、「ほかの $\tilde{y}^{(i)} \neq y^{(i)}$ の出力確率よりも大きければよい」

$$\log \mathrm{P}_{\boldsymbol{\theta}}(y^{(i)} \mid \mathbf{x}^{(i)}) > \log \mathrm{P}_{\boldsymbol{\theta}}(\tilde{y}^{(i)} \mid \mathbf{x}^{(i)})$$

$$\Leftrightarrow \left\langle \boldsymbol{\theta}, \boldsymbol{\Phi}(\mathbf{x}^{(i)}, y^{(i)}) \right\rangle - \left\langle \boldsymbol{\theta}, \boldsymbol{\Phi}(\mathbf{x}^{(i)}, \tilde{y}^{(i)}) \right\rangle > 0$$

隠れマルコフパーセプトロンのアルゴリズム

1. i番目の訓練データに対して予測してみる

$$\mathbf{y}^{(i)}_{predict} = \underset{\mathbf{y}}{\operatorname{argmax}} \mathrm{P}(\mathbf{y} \mid \mathbf{x}^{(i)})$$

2. 当たっていれば($\mathbf{y}^{(i)}_{predict} \neq \mathbf{y}^{(i)}$)何もしない
3. 外れていたら、パラメータを修正

$$\boldsymbol{\theta}^{new} \leftarrow \boldsymbol{\theta}^{old} + \eta\left(\boldsymbol{\Phi}(\mathbf{x}^{(i)}, y^{(i)}) - \boldsymbol{\Phi}(\mathbf{x}^{(i)}, \tilde{y}^{(i)})\right)$$

4. 1.に戻って繰り返す

最尤推定と隠れマルコフパーセプトロンの違いは、argmax操作のみで実現できるところ

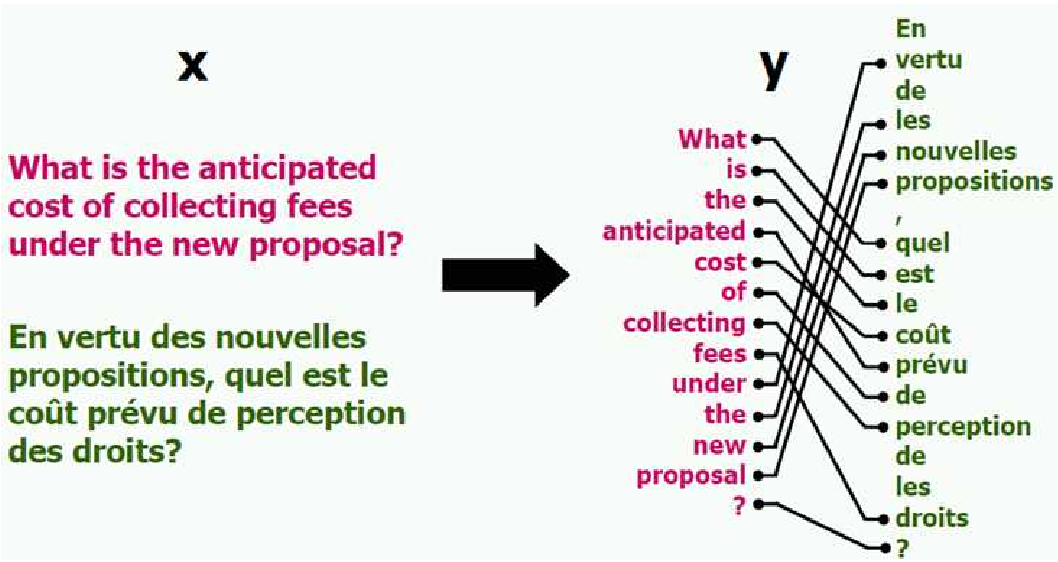
- **隠れマルコフパーセプトロンは訓練と予測が両方argmax操作**

	CRF	隠れマルコフパーセプトロン
訓練	sum	argmax
予測	argmax	argmax

- **通常、argmaxもΣΠも動的計画法で実現できるのであまり変わりはない…しかし…**

隠れマルコフパーセプトロンは、最尤推定では解けない問題が解ける？

- 問題によっては、argmax操作は、動的計画法以外の多項式時間アルゴリズムで実現できる
- 動的計画法で多項式時間で解けない問題は、最尤推定では解けない
  - たとえば、異なる言語の文章間での単語のマッチング
    - argmax操作が線形計画法 (多項式時間) で解ける

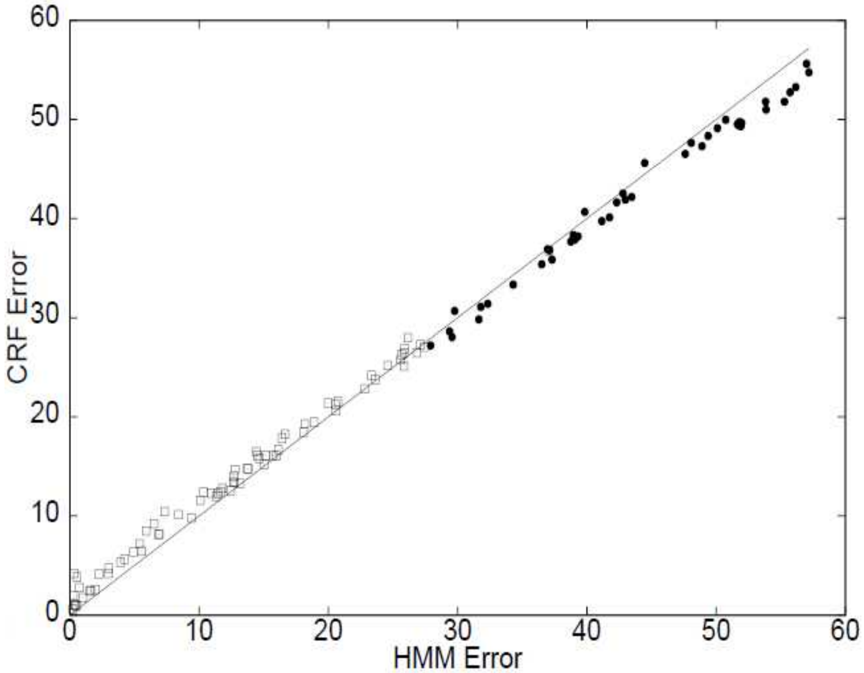


(Taskar, 2005)

実験

- 人工データ
- 英語の品詞タグ付け
- 日本語形態素解析

人工データによる HMM と CRF の比較 (Lafferty 01)



α 小　　α 大

□: α < ½　● α > 1/2

- Second/first mixture HMMからランダムサンプリング
  - P(y|y',y'') = α P(y|y') + (1 – α) P(y|y',y'')
  - P(x|y,x') = α P(x|y) + (1 – α) P(y|y',x')
  - |Y| = 5, |X| = 26
- HMM, CRF 両方とも first order
- 結果
  - α 小: HMM wins
  - α 大: CRF wins
- 真のモデルから外れてくると CRF の識別性能が生きてくる

英語の品詞タグ付け (Lafferty 01)

model	*error*	*oov error*
HMM	5.69%	45.99%
MEMM	6.37%	54.61%
CRF	5.55%	48.05%
MEMM^{+}	4.81%	26.99%
CRF^{+}	4.27%	23.76%

$^{+}$Using spelling features

- **Penn treebank の 50%-50% split, first-order**
- **MEMM (最大エントロピー法を逐次適用するモデル)**
- **spelling feature**
  - -ing, -ogy, -ed, -s, -ly, -ion, -tion, -ity, -ies

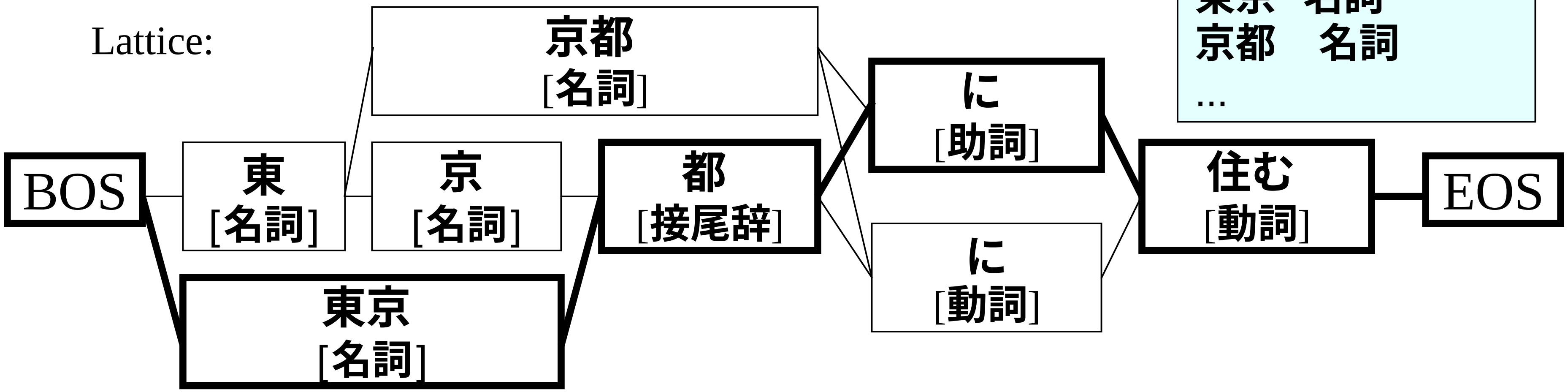
日本語形態素解析 (Kudo 04)

辞書

に	助詞, 動詞
東	名詞
京	名詞
東京	名詞
京都	名詞
…	

Input: “東京都に住む”

Lattice:



- 出力 y の長さが出力経路によって異なる問題
- 古くから HMM が使われてきた
- CRF も HMM と同じアナロジーで応用可能

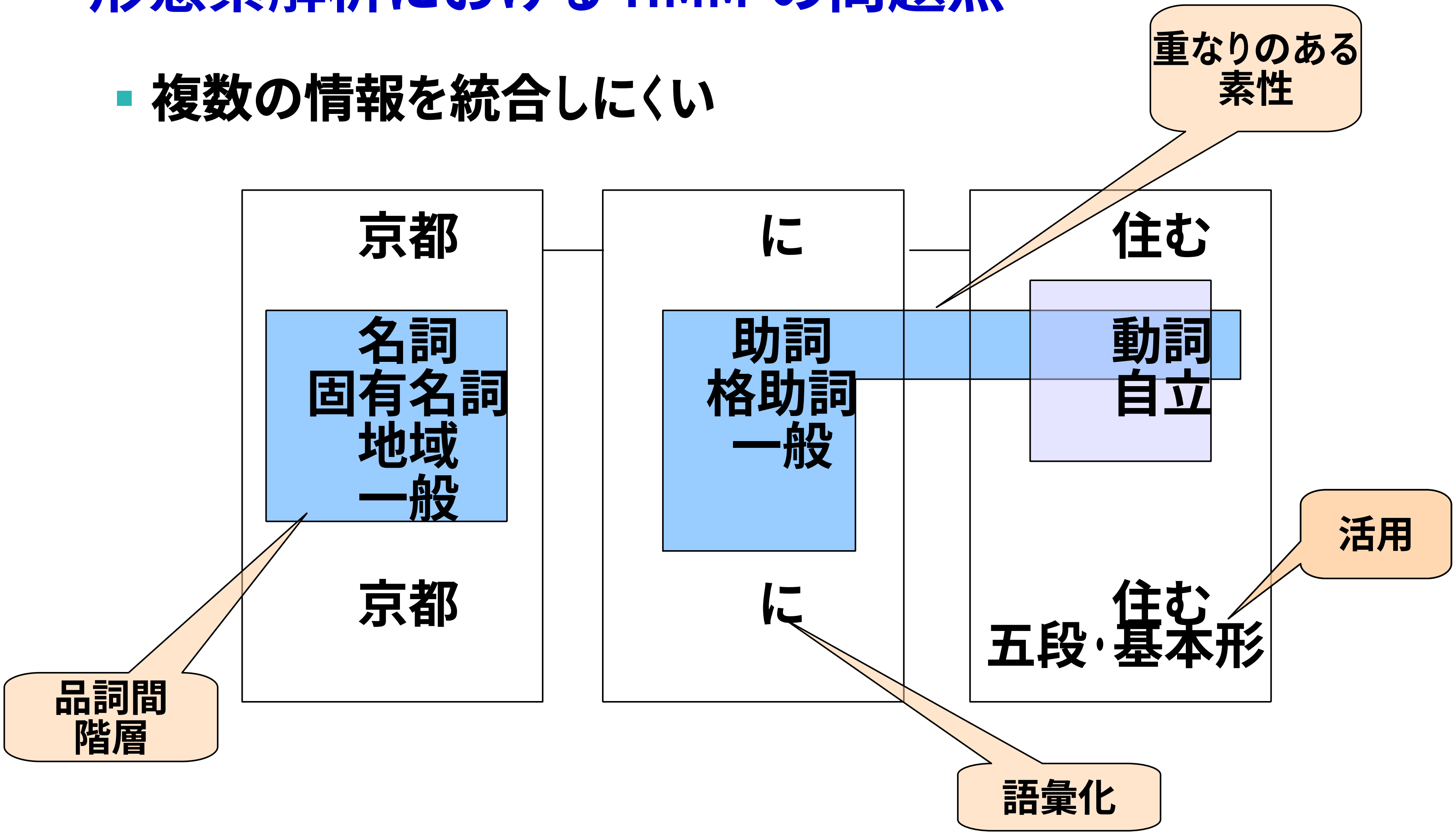
形態素解析における HMM の問題点

- 品詞 = 隠れクラスでいいの?
  - 品詞は階層構造を持つ
  - 何を隠れクラスにすればいいのか?
    - 名詞,動詞ぐらいの浅い階層→細かい違いを区別できない
    - 深い階層 (名詞-固有名詞-人名-姓) →データスパースネス
    - 助詞(は,が,を) は語彙そのものを品詞としたい (語彙化)
  - HMMの粒度の粗さと辞書定義の粒度の細かさのギャップ

京都 名詞 固有名詞 地域 一般 京都

形態素解析における HMM の問題点

- 複数の情報を統合しにくい



CRF は辞書定義が複雑な日本語形態素解析に適している

日本語形態素解析

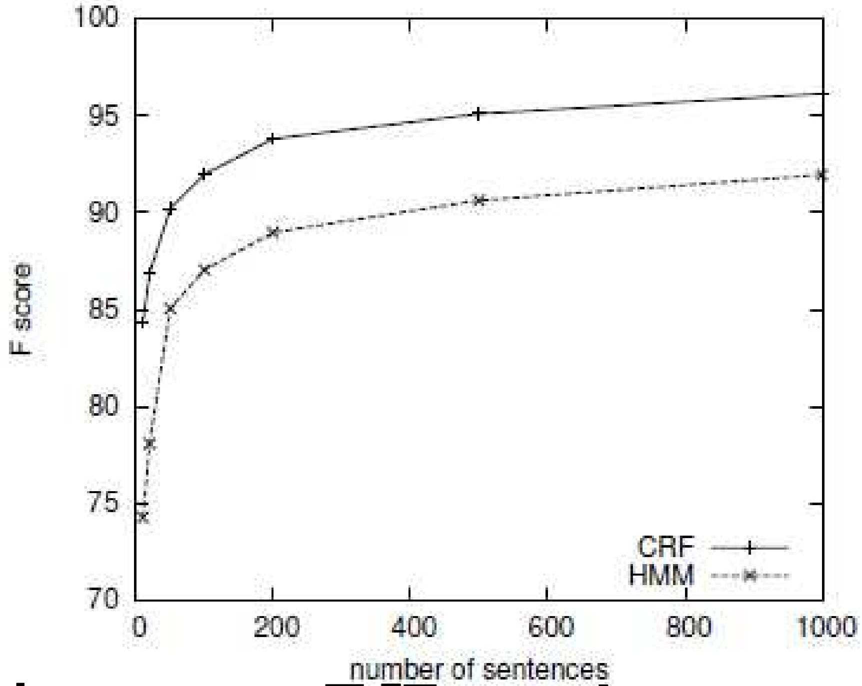
図 3: 実験結果: KC

system	$F_{\beta=1}$ (*seg* / *top* / *all*)
CRF	98.96 / 98.31 / 96.75
HMM	96.22 / 94.99 / 91.85
JUMAN	98.70 / 98.09 / 94.35

図 4: 実験結果: RWCP

system	$F_{\beta=1}$ (*seg* / *top* / *all*)
CRF	99.11 / 98.72 / 97.65
HMM	96.42 / 95.81 / 94.16
ChaSen	98.86 / 98.38 / 97.00

図 5: 学習データ量と精度の関係



- 二つのコーパス, seg:分割のみ, top:品詞, all:すべて
- CRF: 階層化品詞, 字種, 重なりのある複数の素性
- 複雑な辞書定義を CRF がうまくとらえている

CRF のツールキット

- **MALLET: A Machine Learning for Language Toolkit**
  - http://mallet.cs.umass.edu/index.php/Main Page
  - **CRF の他に, 文書分類, クラスタリング, 情報抽出が可能**
- **CRF Project Page**
  - http://crf.sourceforge.net/
  - **API の提供がメインであり, 単体の実装よりは組み込み向け**
- **FlexCRF**
  - http://www.jaist.ac.jp/hieuxuan/flexcrfs/flexcrfs.html
  - 1st-order **の他に**2nd-order (trigram) **のCRFをサポート**
  - **並列計算機を使って大量データの学習が可能**

CRF のツールキット

- **CRF++**
  - http://chasen.org/~taku/software/CRF++/
  - コンパクトな設計, 導入が簡単
  - 素性の定義をテンプレートファイルとして定義
  - n-best 解, 周辺確率の計算
  - 高速
- **MeCab**
  - http://mecab.sourceforge.jp
  - CRF を採用した日本発の形態素解析エンジン
  - 任意のコーパスから学習可能
  - 素性の自由な設計
  - IPA 辞書以外に Juman, CSJ コーパスをサポート

CRF++ (日本語固有表現の例)

```
ノーベル カタカナ 名詞-固有名詞-人名-一般 B-ARTIFACT
文学 漢字 名詞-一般 I-ARTIFACT
賞 漢字名詞-一般 I-ARTIFACT
授賞 漢字 名詞-サ変接続 O
式 漢字 名詞-一般 O
の ひらがな 助詞-連体化 O
大江 漢字 名詞-固有名詞-人名-姓 B-PERSON
健三郎 漢字 名詞-固有名詞-人名-名 I-PERSON
さん ひらがな 名詞-接尾-人名 O
、記号 記号-読点 O
晩さん 漢字-ひらがな 名詞-一般 O
```

学習データ

```
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]]
```

素性テンプレート

- % crf_learn template train_corpus model
- % crf_test -m model < test_corpus