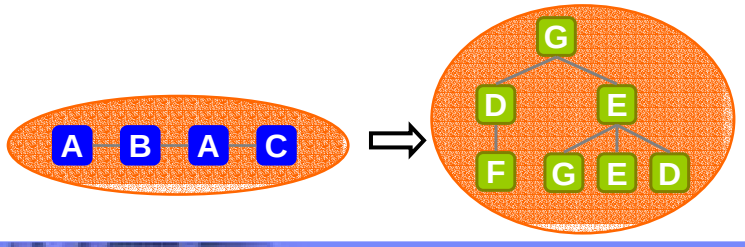
IBM Research

構造化データのラベル付け学習とカーネル法

鹿島 久嗣　　坪井祐太
IBM 東京基礎研究所

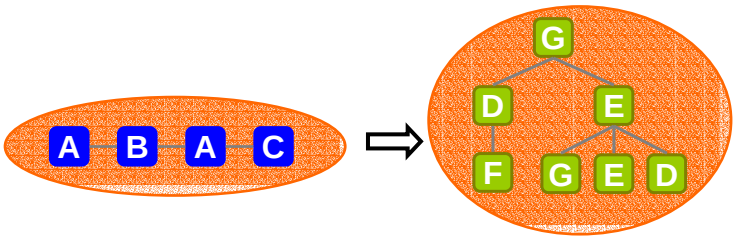
カーネル法による構造化データのマッピング学習

1）構造化データのマッピング問題とは
2）構造マッピング問題へのアプローチ
3）構造マッピング問題へのカーネル法の適用



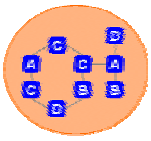
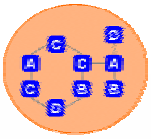
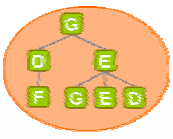
1) 構造化データのマッピング問題とは

- **構造マッピング問題とは**
- **構造マッピング問題の例**
- **構造ラベル付け問題**



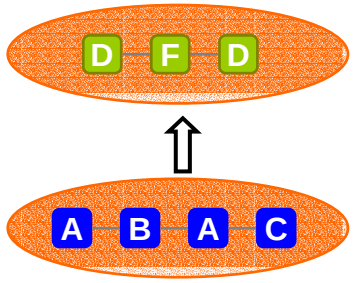
構造マッピング問題とは：構造分類問題の一般的なケース

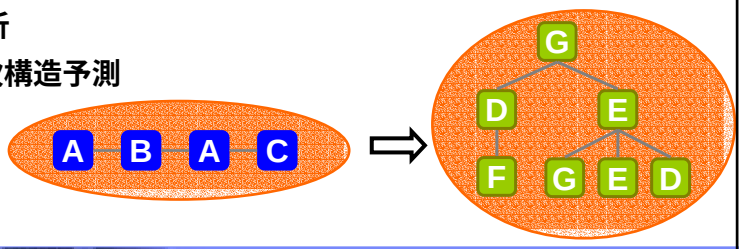
- **目的: 写像 *h*: *X* → *Y* を求める**
  - *X* も *Y* も 構造化データの集合
  - N 個の訓練データ：$(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{x}^{(2)}, \mathbf{y}^{(2)}), \cdots, (\mathbf{x}^{(N)}, \mathbf{y}^{(N)})$
    - ただし、$(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \in X \times Y$

	X	*Y*
2クラス分類	 構造化データ	{+1, -1}
構造マッピング	 構造化データ	 構造化データ

構造マッピング問題の例

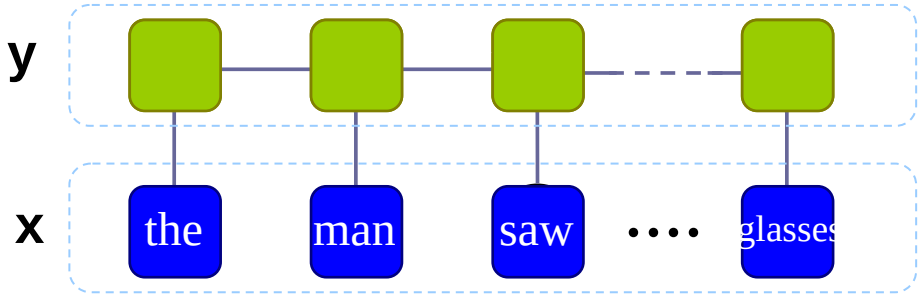
- **配列 ⇒ 配列**
  - 形態素解析
  - 固有表現抽出
  - 蛋白質2次構造予測
  - 遺伝子発見
- **配列 ⇒ 木構造**
  - 構文解析
  - RNA2次構造予測





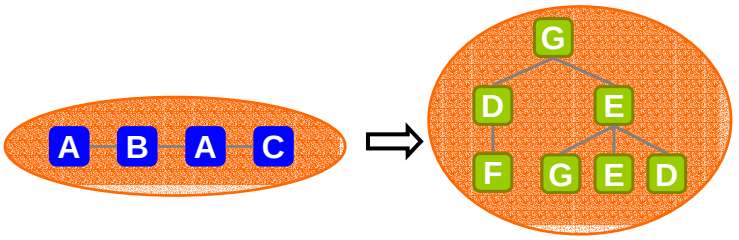
構造ラベル付け問題

- **構造マッピングの特殊なケース**
  - 構造は固定、y 変数のラベルを当てる
  - 例：品詞付け（単語列 ⇒ 品詞列）



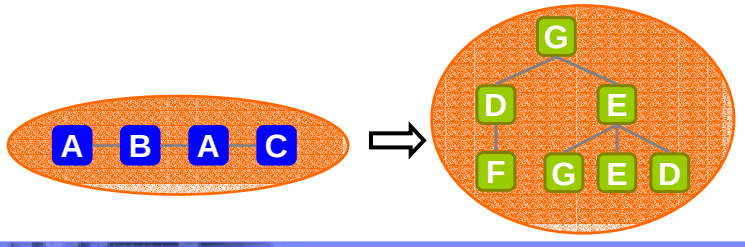
まとめ 1) 構造化データのマッピング問題

- **マッピング問題とは**
  - 構造化データから構造化データへの写像を求める問題
  - 分類問題の拡張
- **構造ラベリング問題とは、**
  - 構造は固定、y 変数のラベルを当てる問題
  - 構造マッピング問題の特殊な場合



2) 構造マッピング問題へのアプローチ

- **2つのアプローチ**
  - 直接マッピング方式と、複合特徴空間方式
- **複合特徴空間方式にもとづく学習アルゴリズムの設計指針**
  - データ表現、予測器、目的関数、最適化法

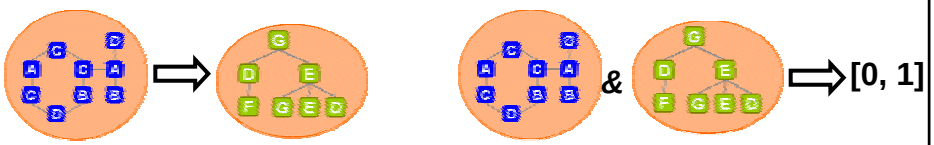


構造マッピング問題への2つのアプローチ

- **直接マッピング方式**: x の特徴空間から y の特徴空間へのマッピングを直接求める
  - $\Phi(y) = h(\Phi(x))$ を求め、$y = \Phi^{-1}(\Phi(y))$

今回のお話

- **複合特徴空間方式**: x と y を合わせた構造に対する良し悪しを評価する
  - x と y を合わせた構造の特徴ベクトル $\Phi(x, y)$ の条件付確率分布
  - 分類問題の自然な拡張



直接マッピング方式　　複合特徴空間方式

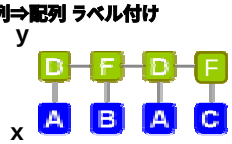
学習アルゴリズム設計において考えなければならないこと

- **学習**
  i. 対象の表現
  ii. 予測器のクラス
  iii. 目的関数
  iv. 最適化の戦略
- **予測**
  i. 最適化アルゴリズム

考えること (i) :対象の表現

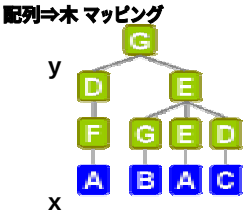
- **対象の特徴空間表現**
- **x と y を合わせた構造 (グラフ) に対する特徴ベクトル** $\Phi(x, y)$
  - 通常の分類問題は、y 変数が1つの場合に相当

配列⇒配列 ラベル付け



$\Phi(x, y)$

配列⇒木 マッピング

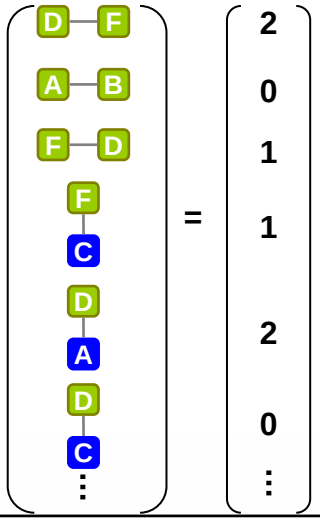


$\Phi(x, y)$

IBM Research

考えること (i) :対象の表現

- **特徴ベクトルの要素は、部分構造の出現数** $\Phi(x, y)$
  - たとえば、サイズ2のすべての部分構造を特徴に用いる
  - 一般的には、部分グラフ



=

2
0
1
1
2
0
⋮

考えること (ii) : 予測器のクラス = 条件付確率場 (CRF)

- **x が与えられたときの y の確率分布があればよい**
- **x と y を合わせた構造に対する特徴ベクトル $\Phi(\mathrm{x},\mathrm{y})$ の条件付確率分布 *P* (y |x) を考える**
  - Logistic regression **の拡張**
  - **重みベクトル** w **を使って、**

$$P(\mathrm{y}|\mathrm{x}) = \frac{\exp(\langle \mathrm{w}, \Phi(\mathrm{x},\mathrm{y})\rangle)}{Z(\mathrm{x})}$$

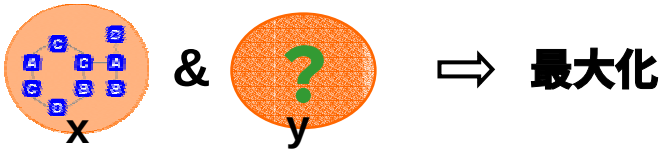
  - $Z(\mathrm{x})$ **は確率分布にするための正規化項**
    - **たとえば、条件付確率場** (Conditional Random Field; CRF) **では、すべての** Y**についての和をとる**

$$Z(\mathrm{x}) = \sum_{\mathrm{y}\in Y} \exp(\langle \mathrm{w}, \Phi(\mathrm{x},\mathrm{y})\rangle)$$

考えること (ii) : 予測器のクラス

- **x が与えられたときの y の予測**
- **重み w と特徴ベクトル $\Phi(\mathrm{x},\mathrm{y})$ の内積が最大になる y を出力する**

$$\begin{aligned}\hat{\mathrm{y}} &= \operatorname*{argmax}_{\mathrm{y}\in Y} \log P(\mathrm{y}|\mathrm{x}) \\ &= \operatorname*{argmax}_{\mathrm{y}\in Y} \log \frac{\exp(\langle \mathrm{w}, \Phi(\mathrm{x},\mathrm{y})\rangle)}{Z(\mathrm{x})} \\ &= \operatorname*{argmax}_{\mathrm{y}\in Y} \langle \mathrm{w}, \Phi(\mathrm{x},\mathrm{y})\rangle\end{aligned}$$



考えること (iii) : 学習の目的関数

- **何をもって、良い予測器とするか**
- **尤度**
  - 訓練データ $(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{x}^{(2)}, \mathbf{y}^{(2)}), \cdots, (\mathbf{x}^{(N)}, \mathbf{y}^{(N)})$ に対する尤度

$$L = \sum_{i=1}^{N} \log P(\mathbf{y}^{(i)} | \mathbf{x}^{(i)})$$

    - x が与えられたとき、y を当てることを目的にする
    - ⇔ HMM, CFGなどは $\Phi(\mathbf{x}, \mathbf{y})$ の条件付きでない確率分布の尤度

考えること (iii) : 学習の目的関数

- **あるいは、周辺化された尤度**
  - 部分的でもなるべく当てることを目指す
    - ⇔ 尤度: y を完全に当てることを目指す
  - 訓練データ $(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{x}^{(2)}, \mathbf{y}^{(2)}), \cdots, (\mathbf{x}^{(N)}, \mathbf{y}^{(N)})$ の各変数について周辺化された尤度

$$L' = \sum_{i=1}^{N} \sum_{t=1}^{|\mathbf{y}^{(i)}|} \log P(y_t | \mathbf{x}^{(i)}) = \sum_{i=1}^{N} \sum_{t=1}^{|\mathbf{y}^{(i)}|} \log \sum_{\mathbf{y}: y_t = y_t^{(i)}} P(\mathbf{y} | \mathbf{x}^{(i)})$$

- **この場合、予測器は、周辺化された条件付確率**
  - y 変数をひとつづつ予測する

$$P(y_t | \mathbf{x}) = \sum_{\hat{\mathbf{y}}: \hat{y}_t = y_t} \frac{\exp(\langle \mathbf{w}, \Phi(\mathbf{x}, \hat{\mathbf{y}}) \rangle)}{Z(\mathbf{x})}$$

考えること (iv) : 最適化の戦略

- **目的関数 (=尤度) をどういう基準で最適化するかによって、各種学習アルゴリズムが導かれる**
- **目的関数を尤度としたときの各種戦略**
  - **最尤推定: Lを直接最大化**

$$\mathbf{w} = \arg\max_{\mathbf{w}} L$$

  - **パーセプトロン: 正解の尤度を不正解の尤度より大きく**

$$\log P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) - \log P(\mathbf{y}^{\text{wrong}}|\mathbf{x}^{(i)}) > 0, \quad \mathbf{y}^{\text{wrong}} \neq \mathbf{y}^{(i)}$$

  - **サポートベクトルマシン: 正解の尤度と不正解の尤度のマージンを最大化** $\mathbf{w} = \arg\max_{\mathbf{w}} \sigma$

$$\min_i \log P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) - \log P(\mathbf{y}^{\text{wrong}}|\mathbf{x}^{(i)}) > \sigma, \quad \mathbf{y}^{\text{wrong}} \neq \mathbf{y}^{(i)}$$

最尤推定: Lを直接最大化

- **重みベクトルw についての微分を用いた反復解法が一般的**

$$\frac{\partial L}{\partial \mathbf{w}} = \sum_{i=1}^{N} \left(\Phi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) - \sum_{\mathbf{y} \in Y} \Phi(\mathbf{x}^{(i)}, \mathbf{y}) P(\mathbf{y}|\mathbf{x}^{(i)}) \right)$$

  - **ただし、2項目の計算がちょっと厄介**
    - **動的計画法で計算**
      - **配列ラベリングだと、O(特徴サイズ x 全長) の計算量が各反復で必要**

パーセプトロン：正解と不正解の尤度差に注目

- **尤度の差が正になればよしとする**

$$\log P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) - \log P(\mathbf{y}^{\text{wrong}}|\mathbf{x}^{(i)}) > 0, \quad \mathbf{y}^{\text{wrong}} \neq \mathbf{y}^{(i)}$$

$$\Updownarrow$$

$$\langle \mathbf{w}, \Phi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \rangle - \langle \mathbf{w}, \Phi(\mathbf{x}^{(i)}, \mathbf{y}^{\text{wrong}}) \rangle > 0, \quad \mathbf{y}^{\text{wrong}} \neq \mathbf{y}^{(i)}$$

- **逐次学習アルゴリズム**
  - 訓練データをひとつづつ処理しながら学習
  - 1) 訓練データ $\mathbf{x}^{(i)}$ に対する予測を行う（動的計画法）

$$\hat{\mathbf{y}}^{(i)} = \underset{\mathbf{y} \in Y}{\text{argmax}} \langle \mathbf{w}, \Phi(\mathbf{x}^{(i)}, \mathbf{y}) \rangle$$

  - 2) 予測が外れたとき（$\hat{\mathbf{y}}^{(i)} \neq \mathbf{y}^{(i)}$）のみ、重みベクトルを更新
    - $\mathbf{w} \leftarrow \mathbf{w} + \Phi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) - \Phi(\mathbf{x}^{(i)}, \hat{\mathbf{y}}^{(i)})$

重みベクトルに
特徴ベクトルを足す＆引く

SVM：正解と不正解の尤度差に注目

- **正解と不正解の尤度差の<u>マージン</u>**

$$\min_i \log P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) - \log P(\mathbf{y}^{\text{wrong}}|\mathbf{x}^{(i)}) > \sigma, \quad \mathbf{y}^{\text{wrong}} \neq \mathbf{y}^{(i)}$$

$$\Updownarrow$$

$$\min_i \langle \mathbf{w}, \Phi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \rangle - \langle \mathbf{w}, \Phi(\mathbf{x}^{(i)}, \mathbf{y}^{\text{wrong}}) \rangle > \sigma, \quad \mathbf{y}^{\text{wrong}} \neq \mathbf{y}^{(i)}$$

- **マージン最大化にもとづく尤度最適化**
  - $\mathbf{w} = \underset{\mathbf{w}}{\text{argmax}}\, \sigma$

- **性能はだいたい HMM << パーセプトロン < CRF < SVM の順 ？**

まとめ 2）構造マッピング学習問題へのアプローチ

- **2つのアプローチ**
  - 直接マッピング方式と、複合特徴空間方式
- **複合特徴空間方式にもとづく構造マッピング学習アルゴリズムの設計指針**
  - x と y を合わせた構造（グラフ）に対する特徴ベクトルを考える
  - 条件付確率モデル：条件付確率場（CRF）

$$P(\mathrm{y}|\mathrm{x}) = \frac{\exp(\langle \mathrm{w}, \Phi(\mathrm{x}, \mathrm{y}) \rangle)}{Z(\mathrm{x})}$$

  - 目的関数は尤度だが、最適化の基準によって、最尤推定やパーセプトロン、SVMなどが導かれる
- **位置ごとの予測モデルも考えられる**
  - 周辺化された条件付確率

$$P(y_t|\mathrm{x}) = \sum_{\hat{\mathrm{y}}:\hat{y}_t = y_t} \frac{\exp(\langle \mathrm{w}, \Phi(\mathrm{x}, \hat{\mathrm{y}}) \rangle)}{Z(\mathrm{x})}$$

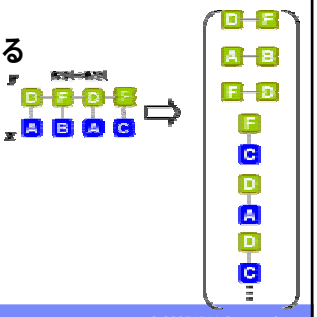
3）構造マッピング学習問題へのカーネル法の適用

- **カーネル化のモチベーション**
- **構造マッピング学習アルゴリズムのカーネル化**
- **カーネル化の問題点と解決策**

カーネル化のモチベーション

- 特徴として、大きなコンテキストを考慮することができるように、もっと大きなサイズの（あわよくば任意サイズの）特徴を使いたい
  - こんな特徴 

も使いたい
    - 非常に多くの特徴（指数個～無限個）を考える必要
    - 計算量的問題
- （畳み込み）カーネルを使おう
  - 指数個～無限個の特徴を、多項式時間で扱える



パーセプトロンからカーネル法へ

- 学習 = 重みベクトル w に特徴ベクトルを足す／引く

$$\mathbf{w} \leftarrow \mathbf{w} + \Phi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) - \Phi(\mathbf{x}^{(i)}, \hat{\mathbf{y}}^{(i)})$$

- w は訓練データの特徴ベクトルの線形和で表せる

$$\mathbf{w} = \sum_{j=1}^{N} \sum_{\mathbf{y} \in Y} \alpha^{(j)}(\mathbf{y})\ \Phi(\mathbf{x}^{(j)}, \mathbf{y})$$

  - $\alpha^{(j)}(\mathbf{y})$ は新たな重みパラメータ
    - ただし、パーセプトロンもSVMも、すべての $\mathbf{y}^{\text{wrong}} \neq \mathbf{y}^{(j)}$ を考える必要はないようにできている
  - CRFやSVMなどでも成立
    - Representer theorem

カーネル条件付確率場 (カーネルCRF)

- **w を特徴ベクトルの線形和で置きかえる**

$$P(\mathrm{y}|\mathrm{x}) = \frac{\exp(\langle \mathrm{w}, \Phi(\mathrm{x},\mathrm{y})\rangle)}{Z(\mathrm{x})}$$

$$\Updownarrow \quad \mathrm{w} = \sum_{j=1}^{N} \sum_{\mathrm{y}\in Y} \alpha^{(j)}(\mathrm{y})\ \Phi(\mathrm{x}^{(j)},\mathrm{y})$$

$$P(\mathrm{y}|\mathrm{x}) = \frac{\exp\left(\sum_{j=1}^{N}\sum_{\mathrm{y}\in Y}\alpha^{(j)}(\mathrm{y})\ \langle\Phi(\mathrm{x}^{(j)},\mathrm{y}),\Phi(\mathrm{x},\mathrm{y})\rangle\right)}{Z(\mathrm{x})}$$

$$Z(\mathrm{x}) = \sum_{\mathrm{y}\in Y}\exp\left(\sum_{j=1}^{N}\sum_{\mathrm{y}\in Y}\alpha^{(j)}(\mathrm{y})\ \langle\Phi(\mathrm{x}^{(j)},\mathrm{y}),\Phi(\mathrm{x},\mathrm{y})\rangle\right)$$

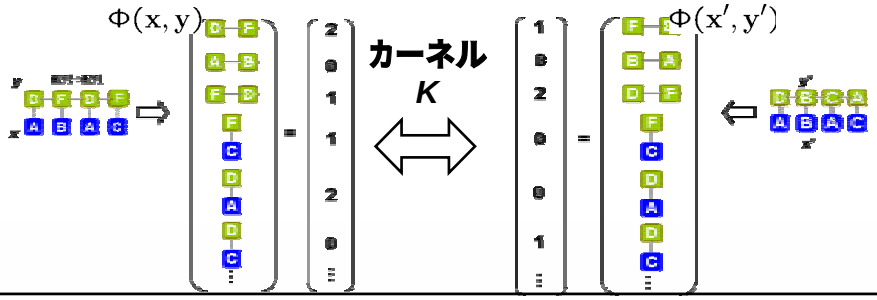
組み合わせた構造の特徴ベクトル同士の内積

(たとえば、Φ(A B C D / A B A C) と Φ(D B C A / A B A C))

IBM Research

構造マッピングにおけるカーネル関数

- **組み合わせた構造の2つの特徴ベクトルの内積を畳み込みカーネル $K((\mathrm{x},\mathrm{y}),(\mathrm{x}',\mathrm{y}')) = \langle\Phi(\mathrm{x},\mathrm{y}),\Phi(\mathrm{x}',\mathrm{y}')\rangle$ で置き換える**
  - 特徴として用いる部分構造のクラスを適切に設定すれば高速計算可
    - 動的計画法、接尾辞木、…
  - 分類問題においては、任意サイズの部分構造を考慮できる
    - 配列カーネル、木カーネル、グラフカーネル
  - 疑問：構造マッピングでもこれは可能か？ ⇒ NO



構造マッピングにおける畳み込みカーネル適用の問題点

- **構造マッピングでは任意サイズの特徴は使えない！**
- **予測や*L*の微分計算で、全てのYを考慮する部分に、部分構造の含む y 要素の数に対して指数的な計算量を要する**
  - Viterbiアルゴリズム（動的計画法）と同じ理由

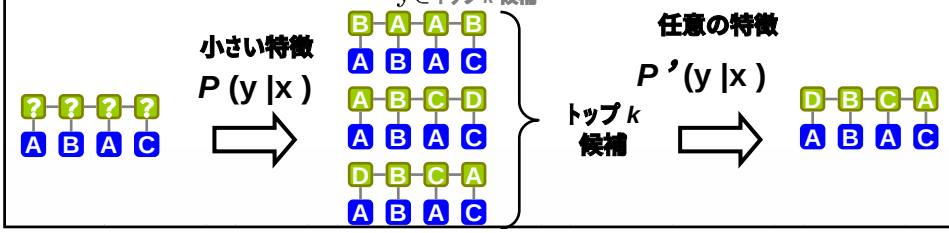
$$\frac{\partial L}{\partial \mathbf{w}} = \sum_{i=1}^{N} \left(\Phi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) - \sum_{\mathbf{y} \in Y} \Phi(\mathbf{x}^{(i)}, \mathbf{y}) P(\mathbf{y}|\mathbf{x}^{(i)}) \right)$$

$$\hat{\mathbf{y}}^{(i)} = \operatorname*{argmax}_{\mathbf{y} \in Y} \langle \mathbf{w}, \Phi(\mathbf{x}^{(i)}, \mathbf{y}) \rangle$$

解決策 1）2段階学習: 再ランキングによる方法

- **アイデア：すべての Y を考えることをあきらめる**
- **小さい特徴にもとづく *P* (y |x) によって、トップ*k*候補を出力**
- **トップ*k*候補と正解を用いて、大きい特徴を用いた*P* ’(y |x)で学習**
  - *Y* 全部ではなく、*P* (y |x) のトップ*k*候補と正解を使って学習
  - あらかじめ候補が*k*個に絞られているので、動的計画法不要
- **予測時、トップ *k* 中に正解が含まれるかは保証されない…**

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y} \in \text{トップ } k \text{ 候補}} \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$$



解決策 2）提案手法：*微妙に* 周辺化された条件付確率場

- **変数毎の予測モデル** $P'(y_t|\mathbf{x})$
  - すべての候補 Y を考慮
    - 「周辺化された条件付確率」のアイデアに基づく
    - 各 y 変数ごとに予測するのでViterbiの必要なし
    - ただし、周辺化の仕方が少し違う
      - 条件付確率の周辺化ではなく、特徴ベクトルの周辺化
  - 任意サイズの特徴を扱える
    - カーネル関数

変数ごとの予測モデル

参考：周辺化された条件付確率

$$P(y_t|\mathbf{x}) = \sum_{\hat{\mathbf{y}}:\hat{y}_t=y_t} \frac{\exp(\langle \mathbf{w}, \Phi(\mathbf{x},\hat{\mathbf{y}})\rangle)}{Z(\mathbf{x})}$$

- **周辺化した確率ではなく、周辺化した特徴ベクトルをつかう**
- ***P*(*y*|*x*)は小さい特徴を使った「一段めの」予測器**
  - すべてのYを考慮できる

任意サイズの特徴を用いた特徴ベクトル

$$P'(y_t|\mathbf{x}) = \frac{\exp\left(\langle \mathbf{w}, \sum_{\mathbf{y}:y_t=\tilde{y}_t} P(\mathbf{y}|\mathbf{x})\Phi(\mathbf{x},\mathbf{y})\rangle\right)}{Z(\mathbf{x})}$$

変数毎のラベル予測 (Viterbi不要)

ytを固定して周辺化した特徴ベクトル (すべてのラベル付けを考慮)

$$Z(\mathbf{x}) = \sum_{\tilde{y}_t} \exp\left(\langle \mathbf{w}, \sum_{\mathbf{y}:y_t=\tilde{y}_t} P(\mathbf{y}|\mathbf{x})\Phi(\mathbf{x},\mathbf{y})\rangle\right)$$

カーネル化

- **カーネルを使って、任意サイズの特徴を用いる**
- **周辺化する「一段めの」分布 $P(y|x)$は小さい特徴を使っているため、カーネル計算の動的計画可能**

$$P'(y_t|\mathbf{x}) = \frac{\exp\left(\sum_{j=1}^{N}\sum_{\tau=1}^{T^{(j)}}\sum_{\tilde{y}_\tau\in\Sigma_y}\alpha^{(j,\tau)}(\tilde{y}_\tau)K(\mathbf{x}^{(j)},\mathbf{x},\tau,t,\tilde{y}_\tau,\tilde{y}_t)\right)}{Z(\mathbf{x})}$$

周辺化カーネル

$$K(\mathbf{x}^{(j)},\mathbf{x},\tau,t,\tilde{y}_\tau,\tilde{y}_t) = \sum_{\mathbf{y}:y_\tau=\tilde{y}_\tau}\sum_{\mathbf{y}':y'_t=\tilde{y}_t} P(\mathbf{y}|\mathbf{x}^{(j)})P(\mathbf{y}'|\mathbf{x})\langle\Phi(\mathbf{x}^{(j)},\mathbf{y}),\Phi(\mathbf{x},\mathbf{y}')\rangle$$

位置

固定したラベル

ラベルをytとyτに固定し、周辺化されたカーネル

パーセプトロン学習も可能

- **予測**

$$\begin{aligned}\hat{y}_t(\mathbf{x}) &= \operatorname*{argmax}_{\tilde{y}_t\in\Sigma_y}\left\langle\mathbf{w},\sum_{\mathbf{y}:y_t=\tilde{y}_t}P(\mathbf{y}|\mathbf{x})\Phi(\mathbf{x},\mathbf{y})\right\rangle\\ &= \operatorname*{argmax}_{\tilde{y}_t\in\Sigma_y}\sum_{j=1}^{N}\sum_{\tau=1}^{T^{(j)}}\sum_{\tilde{y}_\tau\in\Sigma_y}\alpha_{j\tau}(\tilde{y}_\tau)K(\mathbf{x}^{(j)},\mathbf{x},\tau,t,\tilde{y}_\tau,\tilde{y}_t)\end{aligned}$$

- **重みの更新**
  - **予測を間違えたときのみ**

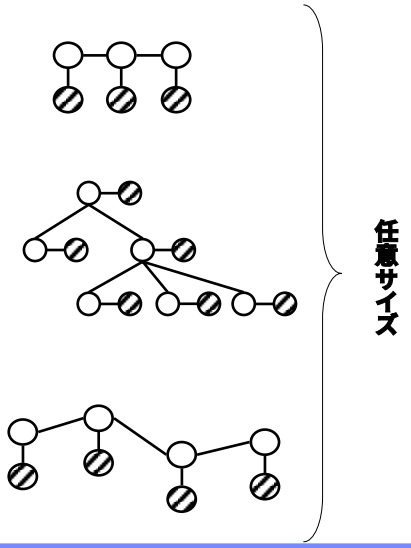
$$\mathbf{w} \Leftarrow \mathbf{w} + \sum_{\mathbf{y}:y_t=y_t^{(i)}}P(\mathbf{y}|\mathbf{x})\Phi(\mathbf{x},\mathbf{y}) - \sum_{\mathbf{y}:y_t=\hat{y}_t^{(i)}}P(\mathbf{y}|\mathbf{x})\Phi(\mathbf{x},\mathbf{y})$$

  - **カーネルの場合**

$$\begin{aligned}\alpha^{(i,t)}(y_t^{(i)}) &\Leftarrow \alpha^{(i,t)}(y_t^{(i)}) + 1\\ \alpha^{(i,t)}(\hat{y}_t^{(i)}) &\Leftarrow \alpha^{(i,t)}(\hat{y}_t^{(i)}) - 1\end{aligned}$$

周辺化カーネルは動的計画法で再帰計算

- **配列ラベル付け**
  - 配列型の特徴
- **順序木ラベル付け**
  - 木型の特徴
- **非循環グラフのラベル付け**
  - パス型の特徴



まとめ 3）構造マッピング学習問題へのカーネル法の適用

- **構造マッピング問題で、任意サイズの特徴を用いるためのひとつのアプローチはカーネル法**
- **しかし、そのままでは適用不可能なので、2段階学習などの方法が必要**
- **提案法はすべての候補を考えつつ、多項式時間で実現**
  - 位置毎のラベル予測と、周辺化された特徴ベクトル、周辺化カーネルによる

おわりに：評価実験

- **広範囲のコンテキストが見られることの有効性を検証**
- **2つの実験**
  - 配列のラベル付け：Named Entity Recognition (NER)
    人名、組織名、場所名などを示すフレーズの抽出
  - 木のラベル付け：Product Usage Information Extraction
    ある製品を使っているという事実を抽出。
    ラベル：製品名・企業名・数量・理由×
    導入・導入検討中・導入しない

いいわけ

- **本当は**
  - 小さい特徴にもとづく予測器
  - 2段階学習：小さい特徴での予測器→任意の特徴での予測器
  - 提案手法：小さい特徴での予測器→任意の特徴での予測器

で、比較したいが、

  - 小さい特徴にもとづく予測器
  - 提案手法：ランダム予測器→任意の特徴での予測器

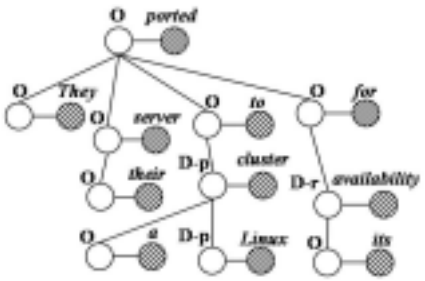
で、比較しました

Named Entity Recognition

- 300 文 (8,541 語) [Special Session of CoNLL2002 on NER]
- 人名、組織名、場所名などを示すフレーズの抽出
  - 9種類のラベルから予測 $|\Sigma_y| = 9$
- 素性 ([Altun et al. 2003] S2 feature)
  - 単語
  - スペリング
    - prefix and suffix of one, two, three letters
    - upper/lower case
    - contains dot
    - ・・・
- window size = 3 for HM-Perceptron
- 2 degree polynomial kernel (Wordの組み合わせfeature)

Product Usage Information Extraction task

- 184 文 (3,570 語)：営業のセールスログ (日本語)
- ある製品を使っているという事実を抽出。
  - 12種類のラベル (製品名・企業名・数量・理由×導入・導入検討中・導入しない) から予測 $|\Sigma_y| = 12$
- 特徴
  - 単語
  - part-of-speech
  - 原型
  - 文字の素性
    - アルファベット
    - 数値
    - ひらがな、カタカナ
- 2 degree polynomial kernel (Wordの組み合わせfeature)
- Japanese Statistical Parser [Kanayama et al. 2000]



結果 (5-fold cross validation)

- **Named entity recognition**

表 1　固有表現抽出結果 (括弧内は標準偏差)

	精度	適合率	再現率	F1
配列カーネル	**88.7**% (3.4)	**49.0**% (6.0)	**23.1**% (8.1)	**30.5** (6.7)
HM パーセプトロン	80.2% (11.5)	23.8% (14.6)	17.9% (3.0)	18.6 (5.2)

- **product usage information extraction**

表 2　製品使用情報抽出結果 (括弧内は標準偏差)

	精度	適合率	再現率	F1
SEQUENCE KERNEL	89.7% (2.0)	**52.2**% (9.5)	29.6% (4.0)	37.5(4.1)
TREE KERNEL	**89.9**% (2.7)	51.4% (10.9)	**32.5**% (14.4)	**38.9** (12.1)
HM-PERCEPTRON	89.7%(1.8)	51.5%(8.5)	24.0%(21.4)	28.9(20.3)

おわり

カーネル法による構造化データのマッピング学習

1) **構造化データのマッピング学習問題**
2) **構造マッピング学習問題へのアプローチ**
3) **構造マッピング学習問題へのカーネル法の適用**