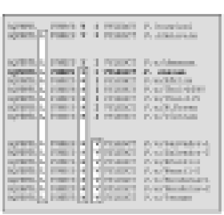
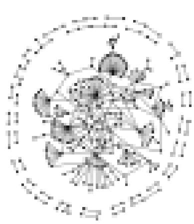
カーネル法による 構造データの解析

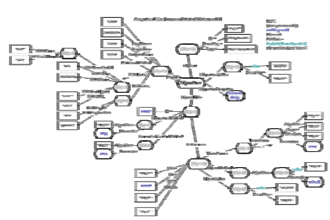
鹿島 久嗣
IBM 東京基礎研究所

はじめに : 構造データ解析のモチベーション

- **従来の機械学習の対象はベクトル形式で表現されていることが前提**
  - **その上での、学習アルゴリズムを議論**
- **これらの枠では捉えきれない「(グラフ) 構造をもつデータ」が増加している**
  - **テキスト**、HTML、XML、**購買履歴**、DNA**配列、蛋白質立体構造、**Web、**遺伝子ネットワーク、社会ネットワーク、**…





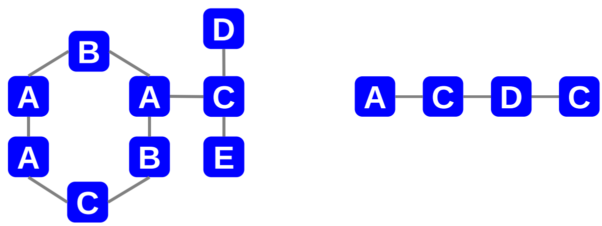


本講演の内容：カーネル法による構造データの解析手法

1）カーネル法とは

2）構造カーネル法 – 畳み込みカーネルを中心に

3）分類問題から、構造マッピング問題へ

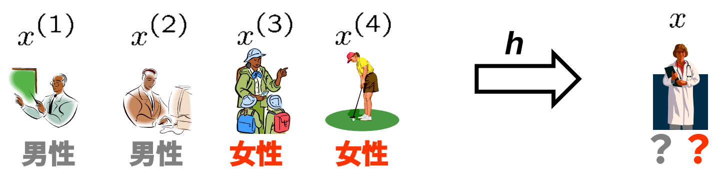


1）カーネル法とは？

- カーネル法の例
  - 2クラスの分類学習問題を題材に紹介
  - パーセプトロンを使った説明
- カーネル法の一般的な概念

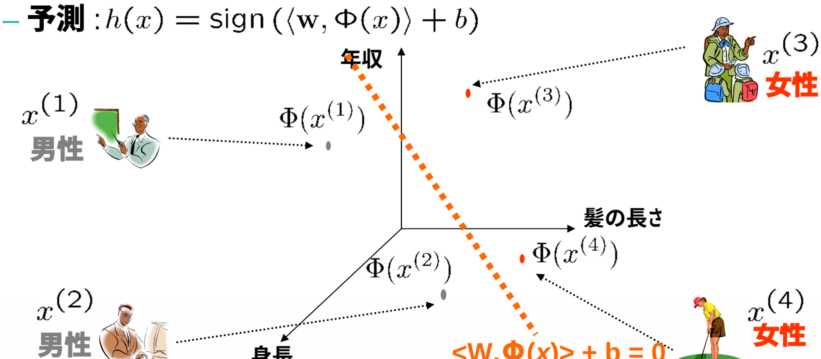
題材: 2クラスの分類学習問題

- **目的: 写像 *h*: *X* → *Y* を求める**
  - *X* :**全ての解析対象の集合**
    - **例: 人間全体の集合**
  - *Y* = {+1, -1}: **解析対象が属するクラスの集合**
    - **例: 男／女**
- **手がかり**
  - *N* **個の訓練データ**: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(N)}, y^{(N)})$
    - **ただし、** $(x^{(i)}, y^{(i)}) \in X \times Y$



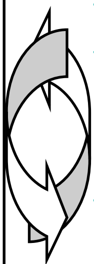
2クラスの分類学習問題への一般的アプローチ

- **解析対象** $x \in X$ **を、特徴空間中の点 (=特徴ベクトル)** $\Phi(x)$ **として表現**
- **それぞれのクラスに属する点集合を分類する面を求める**
  - **超平面** : $\langle \mathrm{w}, \Phi(x) \rangle + b = 0$
    - (w, b) : **重みベクトル**
  - **予測** : $h(x) = \mathrm{sign}\,(\langle \mathrm{w}, \Phi(x) \rangle + b)$



パーセプトロン

- **2クラス分類の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 $x^{(i)}$ に対する予測を行う



$$h(x^{(i)}) = \text{sign}\left(\langle \mathbf{w}, \Phi(x^{(i)})\rangle + b\right)$$

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

      $b \;\leftarrow\; b + y^{(i)}R^2$

重みベクトルに
特徴ベクトルを足す／引く
($y^{(i)} \in \{+1, -1\}$)

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**

$$\mathbf{w} \;\leftarrow\; \mathbf{w} + y^{(i)}\Phi(x^{(i)})$$

- **つまり、w は訓練データの特徴ベクトルの線形和で表せる**

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i \; \Phi(x^{(i)})$$

  - α_i は新たな重みパラメータ
- **カーネル法としての、パーセプトロン**
  - 代入による分類器の書き換え

$$h(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i \langle \Phi(x^{(i)}), \Phi(x)\rangle + b\right)$$

内積による
  - カーネル関数: 内積の置き換えデータアクセス

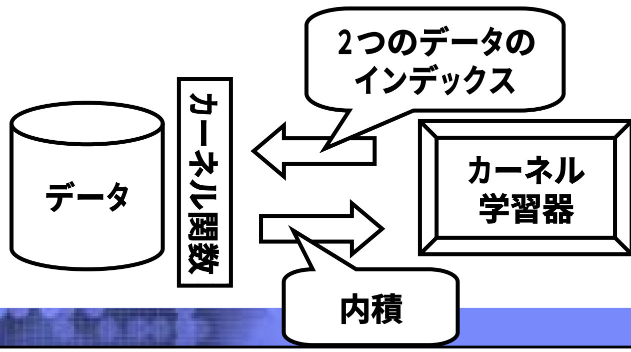
$K(x, x') = \langle \Phi(x), \Phi(x')\rangle$

カーネル法とは

- カーネル関数 (=内積) によってのみデータアクセスを行う学習器

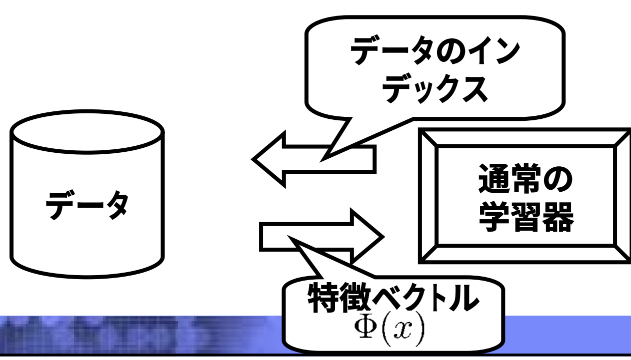
$$K(x,x') = \langle \Phi(x), \Phi(x') \rangle \qquad \text{(カーネル関数)}$$

$$h(x) = \mathrm{sign}\left(\sum_{i=1}^{N} \alpha_i K(x^{(i)}, x) \right) \qquad \text{(カーネル分類器)}$$



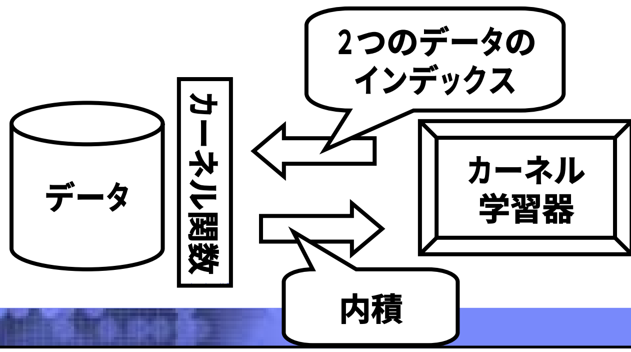
カーネル法のポイント

- 特徴ベクトル $\Phi(x)$が陽に現れない
  - (カーネル関数はさておき) データアクセス部分が次元に依存しない
- x や x' が何であれ、適当に類似度 $K(x,x')$ を定義すれば それがカーネル関数として使える
  - 特徴ベクトルを陽に設計しなくてもよい



カーネル法のポイント

- 特徴ベクトル $\Phi(x)$ が陽に現れない
  – (カーネル関数はさておき) データアクセス部分が次元に依存しない
- x や x' が何であれ、適当に類似度 $K(x,x')$ を定義すればそれがカーネル関数として使える
  – 特徴ベクトルを陽に設計しなくてもよい

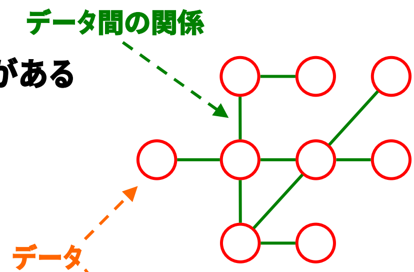


2) 構造カーネル法

- 構造をもったデータ
  – 内部構造と外部構造
- 構造データに対するカーネル
  – 畳み込みカーネルの概念
  – グラフカーネルの設計
  – 様々な構造に対する畳み込みカーネル

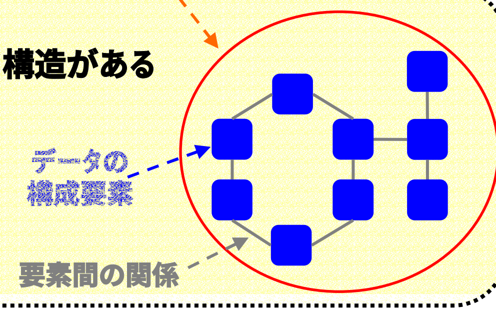
構造をもったデータ

- **外部構造: データ間にグラフ構造がある**
  - Web
  - 社会ネットワーク
  - 遺伝子／蛋白質ネットワーク



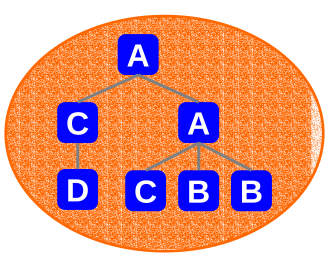
本日のお話

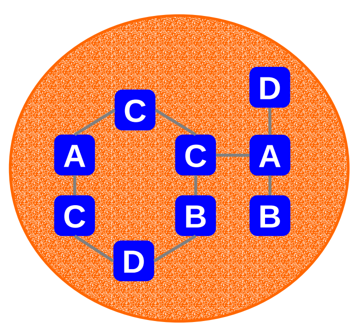
- **内部構造: データ内にグラフ構造がある**
  - HTML、XML
  - DNA
  - 化合物



当面の目標: グラフとグラフの間のカーネル関数をつくる

- **仕様**
  - グラフとグラフの類似度をうまく表している
  - それなりの時間で計算できる





簡単なケース1：単一ノード同士のカーネル関数

- ラベルを比較して、同じなら1、異なるなら 0

$$K_{\Sigma}(\sigma,\sigma') = \begin{cases} 1 & (\sigma = \sigma') \\ 0 & (\sigma \neq \sigma') \end{cases}$$

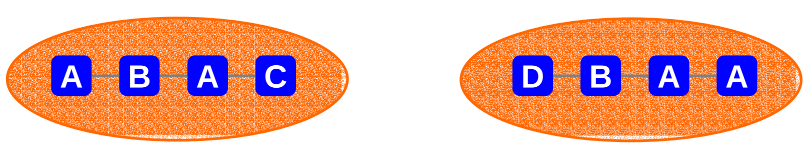
- あるいは、何らかの類似度を定義
  - 原子の性質、大文字／小文字、…

簡単なケース2：長さ *n* の配列同士のカーネル関数

- ノード同士のカーネル関数の積

$$K_S(s,s') = \prod_{i=1}^{n} K_{\Sigma}(\sigma(s_i),\sigma(s'_i))$$

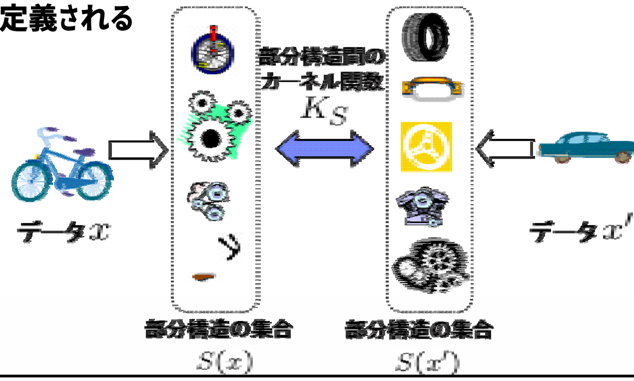
$s = (s_1, s_2, \ldots, s_n)$　　$s' = (s'_1, s'_2, \ldots, s'_n)$



⇒ この調子で拡張していっても
少しでも形が異なった時点でアウト

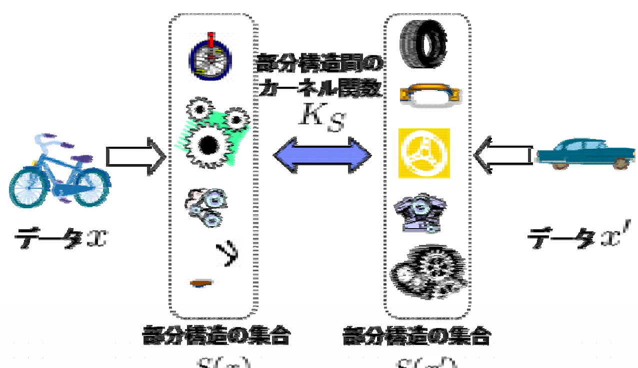
畳み込みカーネル:構造カーネル設計の一般的な枠組み

- 構造データの特徴は、その部分構造が担っている
  - 例: 自動車の特徴は、部品の特徴が担っている
- 2つの構造の間のカーネル関数は、部分構造間のカーネル関数によって再帰的に定義される
  - 例: 自転車と自動車のカーネル関数は、それぞれの部品同士のカーネル関数によって定義される



畳み込みカーネルの定義

- **定義：** $K(x,x') = \sum_{s \in S(x)} \sum_{s' \in S(x')} K_S(s,s')$
  - $S(x)$ ： x の部分構造の集合
  - K_S ：部分構造間のカーネル関数



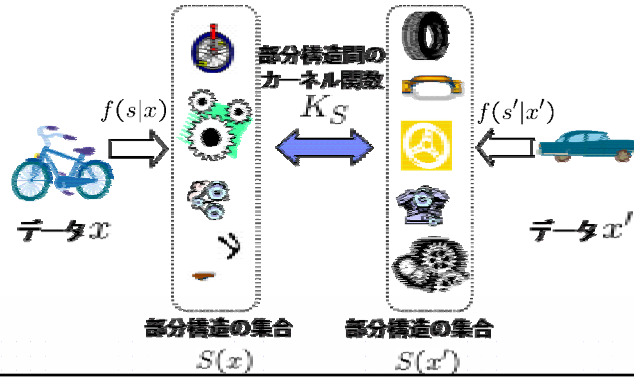
(重み付き) 畳み込みカーネルの定義

- **定義：** $$K(x,x') = \sum_{s \in S(x)} \sum_{s' \in S(x')} f(s|x) f(s'|x') K_S(s,s')$$
  - $S(x)$: x **の部分構造の集合**
  - K_S : **部分構造間のカーネル関数**
  - $f(s|x)$: x **の部分構造** $s \in S(x)$ **の重み**

これらを定義

⇩

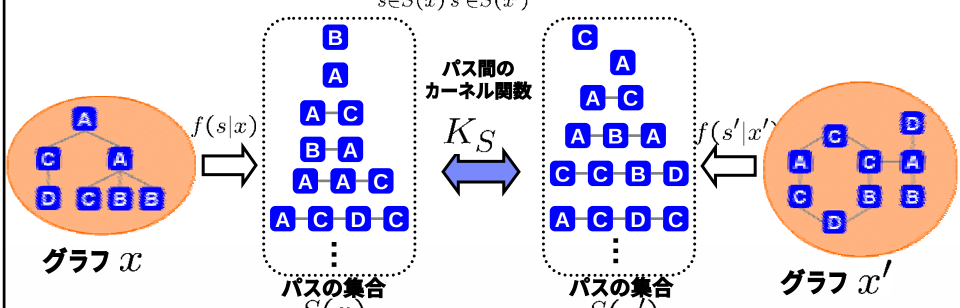
(再帰) 計算



グラフ・カーネルの定義

- **部分構造の集合** $S(x)$ **として何を使うか？**
  - **部分グラフ？ ⇒ ダメ (計算が**NP-hard)
  - **グラフ上のランダム・ウォークによって生成されるパスを使う！**
    - K_S **は同じ長さの配列同士のカーネル**
    - $f(s|x) = \lambda^{|s|}$ **はパス** s **の長さによって減衰** (0< **λ** <1)

$$K(v,v') = \sum_{s \in S(x)} \sum_{s' \in S(x')} \lambda^{|s|} \lambda^{|s'|} K_S(s,s')$$



$$K(v,v') = \sum_{s\in S(x)} \sum_{s'\in S(x')} \lambda^{|s|}\lambda^{|s'|} K_S(s,s')$$

グラフ･カーネルの計算

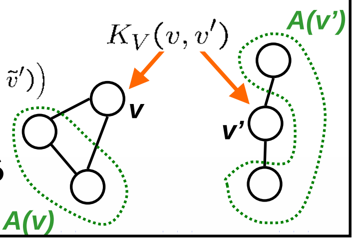
- グラフ上のランダム･ウォークによって生成される<u>パスは無限個ありうる</u>ため、ナイーブな計算はできない
- 再帰計算
  - $S_v(x)$: ノード v で終わるパスの集合　$\left(S(x) = \cup_{v\in V} S_v(x) \right)$
  - ノード対 (v, v') で終わるパスのみに注目したときのカーネルK_Vの和に分解

$$K_V(v,v') = \sum_{s\in S_v(x)} \sum_{s'\in S_{v'}(x')} \lambda^{|s|}\lambda^{|s'|} K_S(s,s')$$

$$K(x,x') = \sum_{v\in V} \sum_{v'\in V'} K_V(v,v')$$

  - K_Vが<u>再帰的に</u>書ける！

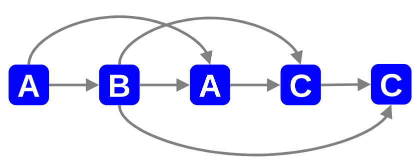
$$K_V(v,v') = \lambda^2 K_\Sigma(v,v')\left(1 + \sum_{\tilde{v}\in A(v)} \sum_{\tilde{v}'\in A(v')} \lambda^2 K_V(\tilde{v},\tilde{v}')\right)$$



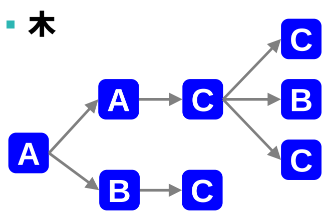
    - 連立方程式を解けばカーネルが計算できる（多項式時間）

特殊な場合

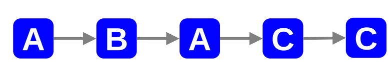
- 非循環有向グラフ（DAG）
  - 再帰式がループしないので、<u>動的計画法</u>で、より速く解ける



- 木



- 配列



様々な畳み込みカーネル

部分構造 \ 対象	配列	順序木	木	DAG	グラフ
パス	○ (※2)	○ (※2)	○ (※2)	○	△ (※1)
部分グラフ		○	×	×	×

※ 1 : 部分構造が同じノードを何度も使える

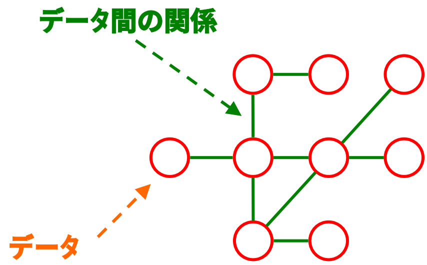
※ 2 : $K_\Sigma(\sigma,\sigma') = \begin{cases} 1 & (\sigma = \sigma') \\ 0 & (\sigma \neq \sigma') \end{cases}$ のときには接尾辞木による線形時間解法

構造カーネルのまとめ

- データのもつ構造は外部構造と内部構造にわけられる
- 内部構造をもつデータのカーネル関数は、部分構造間のカーネル関数によって再帰的に定義される（畳み込みカーネル）
- 部分構造の数は通常、非常に多くなるので畳み込みカーネルのナイーブな計算は困難
- 畳み込みカーネルは、通常、再帰計算によって効率よく計算することができる

補足: 外部構造に対するカーネル関数

- 隣り合うデータ間のカーネル関数が与えられたとき、離れたデータ間のカーネル関数を定義する
  - 例: 拡散カーネル
    - 隣の隣は隣

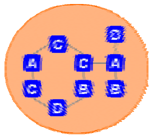
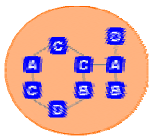
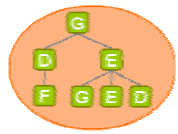


3) 構造マッピング問題

- 構造マッピング問題の定義
- 2つのアプローチ
- 隠れマルコフ･パーセプトロン
- 構造カーネルの適用と問題点
- 2段階学習による解決

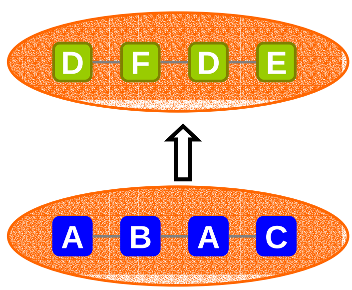
構造マッピング問題

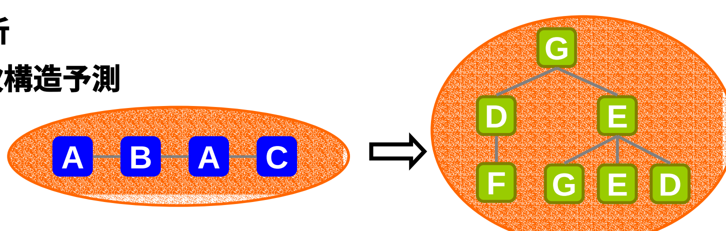
- 構造分類問題の、より一般的なケース

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

構造マッピング問題の例

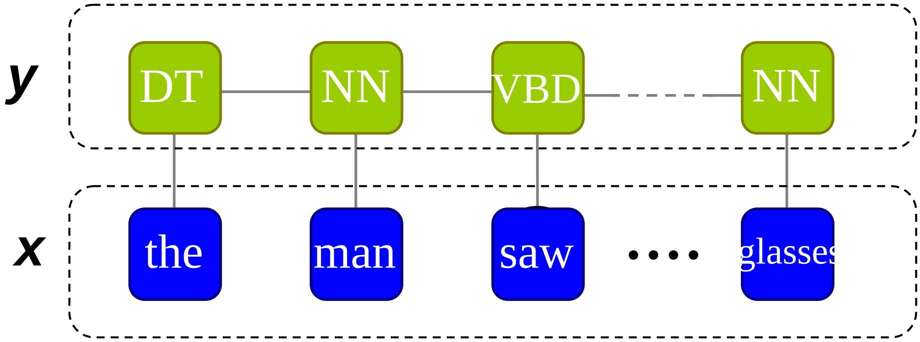
- 長さ*T*の配列⇒長さ*T*の配列
  - 品詞付け
  - 固有表現抽出
  - 蛋白質2次構造予測
  - 遺伝子発見
- 配列⇒木構造
  - 構文解析
  - RNA2次構造予測





品詞付け

- **長さ*T*の配列⇒長さ*T*の配列のマッピング問題**
  - **構造は固定、ラベルだけ当てる**

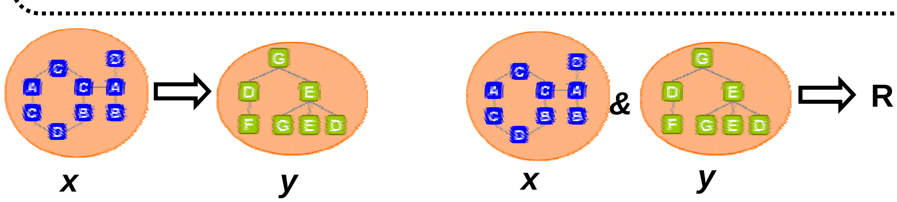


構造マッピング問題への2つのアプローチ

- **<u>直接マッピング方式</u>: *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、** $y = \Phi^{-1}(\Phi(y))$

本日のお話

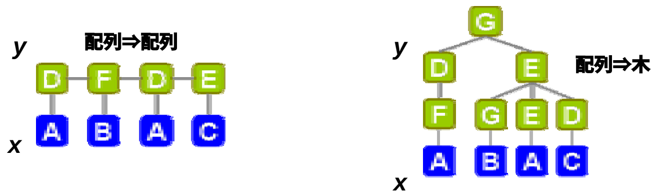
- **<u>ランキング方式</u>: *x* と *y* を合わせた構造に対する良し悪しを評価する**
  - ***x* と *y* を合わせた構造の特徴ベクトル** $\Phi(x, y)$ **のランキング問題**



<u>直接マッピング方式</u>　　**<u>ランキング方式</u>**

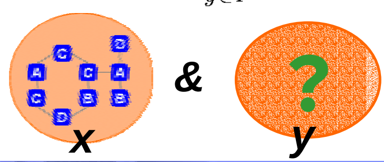
ランキング方式

- x と y を合わせた構造に対する特徴ベクトル $\Phi(x,y)$ を評価



- 重み w による評価値が最大になる y を出力する

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



⇨ 最大化

IBM Research

隠れマルコフ・パーセプトロン

- ランキング方式の構造マッピング学習アルゴリズム
  - 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 $x^{(i)}$ に対する予測を行う

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

  - 2) 予測が外れたとき ($\hat{y}^{(i)} \neq y^{(i)}$) のみ、重みベクトルを更新
    - $\mathbf{w} \leftarrow \mathbf{w} + \Phi(x^{(i)}, y^{(i)}) - \Phi(x^{(i)}, \hat{y}^{(i)})$
  - 全ての誤ったマッピング $y \neq y^{(i)}$ に対し、

$$\langle \mathbf{w}^*, \Phi(x^{(i)}, y^{(i)}) \rangle > \langle \mathbf{w}^*, \Phi(x^{(i)}, y) \rangle$$

やはり、
重みベクトルに
特徴ベクトルを足す＆引く

隠れマルコフ·カーネル·パーセプトロン

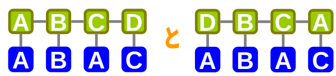
- パーセプトロンと同様、カーネル化できる

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

$$\hat{y}^{(i)} = \operatorname*{argmax}_{y \in Y} \sum_{j=1}^{N} \sum_{\tilde{y} \in Y} \alpha_j(\tilde{y}) \langle \Phi(x^{(j)}, \tilde{y}), \Phi(x^{(i)}, y) \rangle$$

$$= \operatorname*{argmax}_{y \in Y} \sum_{j=1}^{N} \sum_{\tilde{y} \in Y} \alpha_j(\tilde{y}) K((x^{(j)}, \tilde{y}), (x^{(i)}, y))$$

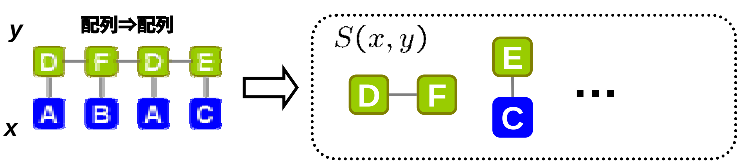
マッピング同士のカーネル関数
(組み合わせた構造)



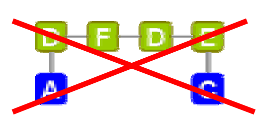
のカーネル関数

マッピング (組み合わせた構造) 同士のカーネル関数

- 畳み込みカーネル $K((x, y), (x', y'))$が使える
- 組み合わせた構造の部分構造 S(*x*, *y*)



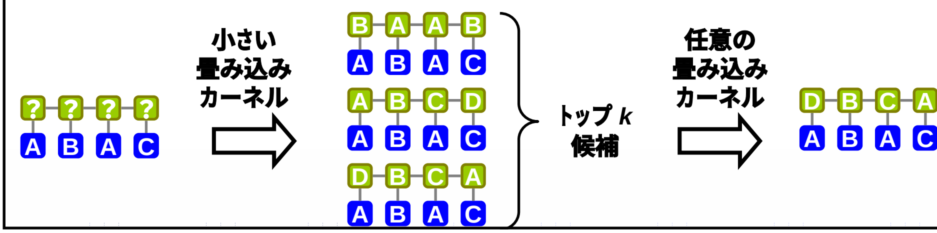
– 注意: *y* の要素は、任意個は使えない
  - argmax 操作が、部分構造の含む*y*要素の数に対して指数的な計算量
  - Viterbiアルゴリズム (動的計画法) と同じ理由



2段階学習: 再ランキングによる方法

- 小さい部分構造にもとづく隠れマルコフ・パーセプトロンによって、トップk候補を出力
  - 但し、この中に正解が含まれるかは保証されない
- そのなかで一番よいものを、大きい部分構造を用いた隠れマルコフ・パーセプトロンで予測
  – あらかじめ候補がk個に絞られているので、Viterbiの必要なし

$$\hat{y}^{(i)} = \underset{y \in \text{トップ}\, k\, \text{候補}}{\operatorname{argmax}} \langle \mathrm{w}, \Phi(x^{(i)}, y) \rangle$$



IBM Research

構造マッピングのまとめ

- 構造マッピング問題には、直接マッピング方式と、ランキング方式がある
- ランキング方式の一手法が隠れマルコフパーセプトロンであり、パーセプトロンと同様、カーネル化できる
- 構造マッピング問題においては、畳み込みカーネルで用いることのできる部分構造に制限がある
- そのため、あくまで任意サイズの部分構造を用いたい場合には、学習を2段階にするなどの工夫が必要

構造カーネルの課題

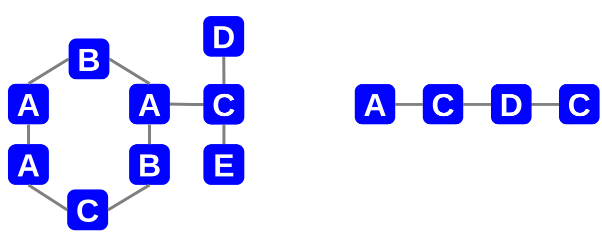
- **高速化**
  - 動的計画法などによって、効率的に計算できるといっても、遅い
    - より高速なアルゴリズム（線形時間、確率、近似）
  - カーネル法の計算量は、訓練例の数に依存
    - 例の効率的な管理（変換、データ構造）
- **特徴選択（属性選択）による精度向上**
  - カーネル関数によって特徴空間表現が隠蔽されているため、特徴選択が陽に行えない
    - 部分構造の重み $f(s|x)$ の学習が特徴選択に相当

おわり

1）カーネル法とは

2）構造カーネル法 – 畳み込みカーネルを中心に

3）分類問題から、構造マッピング問題へ



ありがとうございました