

THE UNIVERSITY OF TOKYO

復合情報学特別講義第二
拡がる機械学習の可能性
ー グラフとネットワークの機械学習を中心として

東京大学大学院
情報理工学系研究科
鹿島 久嗣
東京大学
DEPARTMENT OF MATHEMATICAL INFORMATICS

講師の紹介：企業の研究所で10年間働いたのち、一昨年前に大学へ異動。機械学習が多くの場面で活躍できるよう頑張っています

- 1999年に京都大学工学研究科システム科学専攻を修士課程修了、以降、10年間IBM東京基礎研究所にて研究員として勤務
 - 機械学習／データマイニング技術の研究開発とデータ解析コンサルティング
 - バイオインフォマティクス、コンピュータシステムの障害解析、ビジネス・データ解析（購買管理、人材マネジメント、マーケティング）、製造システム/自動車のセンサーデータ解析、特許データ分析、等々…
- 2007年に社会人博士課程にて博士（情報学）の学位を取得
- Googleで「機械学習」を検索すると、個人ページとしては最もランク上位に表示
- 2009年より東京大学 情報理工学系研究科 数理情報学専攻 准教授
 - データマイニングと機械学習の研究室（教授もNECからの移動）
- 研究のスタンス「データ解析がより多くの重要な場面で活躍できるように！」
- これまで扱うことができなかった形式のデータや新しい問題を見つける

2

THE UNIVERSITY OF TOKYO

本講義の目的：
機械学習の概論＋グラフ／ネットワークデータへの応用

- 統計的機械学習の概論
 - 教師付き学習／教師なし学習
 - 基本的なモデル
 - 学習アルゴリズム
- グラフデータ、ネットワークデータを対象とした解析を行うための方法
 - グラフデータの予測
 - ネットワーク上での予測
 - グラフ構造の予測
 - ネットワーク構造の予測

3

THE UNIVERSITY OF TOKYO

教師付き学習 と 教師なし学習

4

THE UNIVERSITY OF TOKYO

例1 あるなしクイズ：これは「あり」？「なし」？

- ヒント：「あり」なものと、「なし」なもの

あり	なし
うさぎ	ねずみ
はがねのつるぎ	てつのおの
きんとき	あんこ
たわし	わたし
- では…
 - 「ししゃも」は？
 - 「ほっけ」は？
 - 「しゃけ」は？

5

THE UNIVERSITY OF TOKYO

部分文字列に注目してみると… 判別するルールが みえてきます

- ヒント：「あり」なものと、「なし」なもの

あり	なし
うさぎ	ねずみ
はがねのつるぎ	てつのおの
きんとき	あんこ
たわし	わたし
- では…
 - 「ししゃも」は？ ⇒ あり
 - 「ほっけ」は？ ⇒ なし
 - 「しゃけ」は？ ⇒ なし

6

THE UNIVERSITY OF TOKYO

例2 なかまはずれさがし：仲間はずれはどれ？

- 以下のうち、仲間はずれはどれでしょうか？

くも
やどかり
たこ
いか
たらばかに
毛がに
えび

7

THE UNIVERSITY OF TOKYO

グループ分けしてみると…なかまはずれが 見えてきます

- 「足の数」と「かたさ」で分類してみると…

		足の数	
		8本	10本
かたさ	グループ1 やわらかい	くも たこ	いか
	グループ2 かたい	たらばかに やどかり	毛がに えび
		グループ3	

- あるいはもっと安直に、棲んでいる場所に注目すると「くも」であろう

棲んでいる場所	
陸上	水中
くも	その他

8

THE UNIVERSITY OF TOKYO

前述の例は、それぞれ機械学習の2大タスクである「教師つき学習（予測）」と「教師なし学習（発見）」に対応しています

- あるなしクイズの場合：
 - 「ある」「なし」を区別するルールを与えられた事例から見つける
 - 未知の対象に対してルールを適用し分類する
- なかまはずれ探しの場合：
 - ある視点から対象をグループ分けする
 - それぞれのメンバーを評価
- これらはそれぞれ機械学習の2大タスク
 - 「教師つき学習」＝予測
 - 「教師なし学習」＝発見
 に対応している

9

THE UNIVERSITY OF TOKYO

教師つき学習は、入出力関係の推定問題であるといえます

- 目的：入力 x が与えられたとき、対応する出力 y を予測したい
 - 入力 x ：「ししゃも」や「ねずみ」
 - 出力 y ：「あり」「なし」か
 ※ 厳密にはこれは教師つき学習の「分類」と呼ばれるタスク
- つまり、 $y = f(x)$ となる関数 f がほしい
- しかし、ヒントなしでこれではできない…
そこでヒント（過去の事例＝訓練データ）が必要
 - 「うさぎ」は「あり」、「ねずみ」は「なし」、など
- 訓練データをもとに入出力関係 f を推定するのが教師つき学習
 - 正しい出力を与えてくれる「教師」がいるというイメージ
 - 訓練データは f を「訓練する」ためのデータ

10

THE UNIVERSITY OF TOKYO

一方、教師なし学習は、入力データのグループわけ問題と考えることができます

- 教師なし学習では入出力関係についてのヒントがない（出力が与えられず、入力のみが与えられる）
- 入力だけから出力らしきものをつくる必要がある（＝自習）
- 「あり」「なし」などのラベルが明示的に与えられないので、グループ分けが難しい
- 目的：入力 x が与えられたとき、これらをグループ分けしたい
 - 入力 x ：「くも」や「やどかり」
 - 出力 y ：グループ1、グループ2、…など（明示的なラベルを付ける必要はない）
 - 通常グループの数は指定される
 ※ 厳密には教師なし学習の「クラスタリング」と呼ばれるタスク

11

THE UNIVERSITY OF TOKYO

歴史的経緯：結局のところ、機械学習とは、データ分析技術の一流派のようなものです

- 機械学習とは、本来「人間の持つ“学習能力”を機械（計算機）にも持たせる」ことを目指す研究分野
 - もともとは人工知能の一分野として始まる
 - 論理推論がベース
 - 現在では、「統計的」機械学習が主流（＝機械学習）
 - 遺伝子情報処理、自然言語処理、他、ビジネス分野での成功
- 現在では、データ解析技術一般を指すほかの言葉とあまり変わらない
 - 統計／データマイニング／パターン認識など（多少のニュアンスの違いはあるが、基本的に好みの問題）

12

THE UNIVERSITY OF TOKYO

教師付き学習と教師無し学習は機械学習の基本問題です

- 学習者を、入出力のあるシステムと捉え、学習者に対する入力と、それに対する出力の関係を数理的にモデル化する
 - 入力：視覚などからの信号（実数値ベクトルで表現）
 - 出力：入力を表す概念、入力に対してとる行動
- どうやら2つの重要な基本問題があるらしいということになった
 - 教師付き学習：入力に対する出力を試行錯誤するうちに、どういう入力のときにどういう出力をすればよいかがわかってくる
 - 教師無し学習：入力を見ているうちに、どんなものが現れやすいかなどのパターンが分かってくる

13

THE UNIVERSITY OF TOKYO

機械学習を実現するためには、入力の数理的表現が必要です

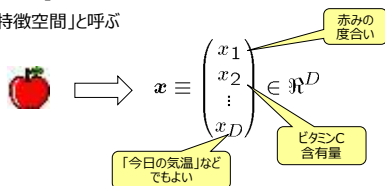
- 学習機能を計算機上に実現するために、まず、学習問題を数理的にとらえる必要がある
- まずは、入力をどう数理的（＝計算機可読な形式）に表現するか？
 - 「やどかり」「ねこ」「りんご」は計算機上でどのように扱うか？
- 出力については比較的自明
 - 「あり」を+1、「なし」を-1と割り当てる

14

THE UNIVERSITY OF TOKYO

入力の表現： 通常、実数値ベクトル（特徴ベクトル）として表現します

- 入力を、その特徴量を列挙した D 次元の実数値ベクトル \mathbf{x} として表現する
 - \mathbf{x} を「特徴ベクトル」と呼ぶ
 - その領域を「特徴空間」と呼ぶ



- 特徴ベクトル \mathbf{x} はどのようにデザインしたらよいか？
 - 完全にドメイン依存。
 - 一般的解はなく、目的に合わせユーザーがデザインする

15

THE UNIVERSITY OF TOKYO

教師付き学習は、条件付確率分布の推定問題です

- 条件付分布 $P(y|\mathbf{x})$ ：入力信号 \mathbf{x} を条件とした、出力 y の確率分布
 - 入力信号 \mathbf{x} は、 D 次元の実数値ベクトル
 - 一方、出力 y は 1次元
 - データの属するカテゴリ
 - > +1 もしくは -1 の 2つ ($y \in \{+1, -1\}$)
(例：🍎 か否か)
 - > 複数のカテゴリ {A, B, C, D, ...}
(例：🍎 か 🍌 か)
 - 実数値： $y \in \mathbb{R}$
- たとえば $\mathbf{x} := \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{pmatrix}$ が 🍎 である確率はいくつか？という質問に答える

16

THE UNIVERSITY OF TOKYO

教師付き学習の訓練データ： 教師付き学習では、入力ベクトルと出力の組が複数与えられます

- 訓練データは、 N 個の入力と出力のペア

$$\{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\}$$

1つ目の
入力ペア

2つ目の
入出力ペア

...

N個目の
入出力ペア

 - $\mathbf{x}^{(i)}$: i 番目の事例の入力ベクトル
 - $y^{(i)}$: i 番目の事例に対する正しい出力
(🍎 ならば +1, 違うなら -1)
- 教師付き学習：与えられた入力信号に対する、あるべき出力を教師信号として、入出力の関係 $P(y|\mathbf{x})$ を学習する

17

THE UNIVERSITY OF TOKYO

教師無し学習は、 特徴空間上での確率分布の推定問題として捉えられます

- 教師無し学習：たくさんの入力信号を見ているうちに、どんなものが現れやすいかなどのパターンが分かってくる
- 入力信号 \mathbf{x} ： D 次元の実数値ベクトルとして表現

$$\mathbf{x} \equiv \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{pmatrix} \in \mathbb{R}^D$$
- この入力信号 \mathbf{x} 上の確率分布 $P(\mathbf{x})$ を考える
 - $P(\mathbf{x})$ の形をみることで、どのあたりの入力信号が現れやすいか／どのようなグループがあるか、などがわかる



18

THE UNIVERSITY OF TOKYO

教師無し学習では、入力ベクトルのみが複数与えられます

- データはN個の入力信号

$$(x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(N)})$$

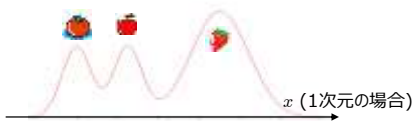
1つめのデータ

2つめのデータ

...

$$x \equiv \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{pmatrix} \in \mathbb{R}^D$$

- 教師無し学習は、(大げさにいえば) 明示的に指定されることなく、
「概念」を形成するプロセスを表している



19

THE UNIVERSITY OF TOKYO

学習のモデル

20

THE UNIVERSITY OF TOKYO

教師付き学習のモデル：

線形モデルは もっともシンプルな出力予測モデルです

- 入力 $x = (x_1, x_2, \dots, x_D)^T$ に対し、
出力 $\{+1, -1\}$ を予測する分類モデル f を考える

$$f(x) = \text{sign}(w^T x) = \text{sign}(w_1 x_1 + w_2 x_2 + \dots + w_D x_D)$$

— $\text{sign}(\cdot)$ は引数が0以上なら+1、0未満なら-1を返す関数

— $w = (w_1, w_2, \dots, w_D)^T$ はモデルパラメータ

- w_d は x_d の出力への貢献度を表す
 - $w_d > 0$ なら出力+1に貢献、 $w_d < 0$ なら出力-1に貢献

21

THE UNIVERSITY OF TOKYO

教師付き学習の確率モデルの典型：ロジスティック回帰モデル 2カテゴリ分類の標準的な線形モデルです

- 出力が2カテゴリの場合の代表的な条件付確率モデル

$$P(y = +1 | x; w) \equiv \sigma(w^T x) = \sigma(w_1 x_1 + w_2 x_2 + \dots + w_D x_D)$$

$$P(y = -1 | x; w) = 1 - \sigma(w^T x)$$

— なお、 w はモデルを定めるパラメータベクトル

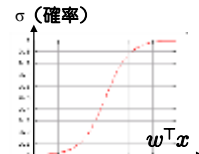
- w の各次元は x の各次元の $P(y = +1 | x; w)$ への寄与度

$$w \equiv \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_D \end{pmatrix} \quad x \equiv \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{pmatrix}$$

— σ はロジスティック関数

- 連続値を確率値 $[0, 1]$ にマップする

$$\sigma(a) := \frac{1}{1 + e^{-a}}$$



22

THE UNIVERSITY OF TOKYO

教師なし学習モデルの典型：混合正規分布モデル 単一の正規分布では表現力が十分ではありません

- D 次元のデータの確率分布として、 D 次元の多次元正規分布 $g(x)$ を考える

$$g(x; \mu, \Sigma) := \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} \exp\left(-\frac{(x - \mu)^T \Sigma^{-1} (x - \mu)}{2}\right)$$

— 1次元の正規分布 $g(x; \mu, \sigma) := \frac{1}{(2\pi)^{1/2} \sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$ の拡張

— パラメータ

- μ : 平均 (D 次元)
- Σ : 共分散行列 ($D \times D$)

- 単峰なので、表現力が不十分



23

THE UNIVERSITY OF TOKYO

教師なし学習モデルの典型：混合正規分布モデル 複数の正規分布によって複雑な分布を表現できます

- K 個の D 次元正規分布 $g^{(1)}(x), g^{(2)}(x), \dots, g^{(K)}(x)$ の混合分布

$$P(x) := \sum_{k=1}^K w^{(k)} g^{(k)}(x; \mu^{(k)}, \Sigma^{(k)})$$

k 番目の正規分布の重み

ただし $\sum_{k=1}^K w^{(k)} = 1, w^{(k)} \geq 0$

k 番目の正規分布

$$g^{(k)}(x; \mu^{(k)}, \Sigma^{(k)}) := \frac{1}{(2\pi)^{D/2} |\Sigma^{(k)}|^{1/2}} \exp\left(-\frac{(x - \mu^{(k)})^T \Sigma^{(k)-1} (x - \mu^{(k)})}{2}\right)$$

- 単一の正規分布より複雑な確率分布を表現できる

- モデルのパラメータは

— 混合比パラメータ $\{w^{(k)}\}_{k=1, \dots, K}$

— 各正規分布のパラメータ

$\{\mu^{(k)}, \Sigma^{(k)}\}_{k=1, \dots, K}$



24

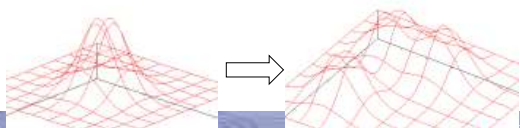
D=2(2次元)、K=3(3つの正規分布の混合)の場合の例

教師なし学習モデルの典型：混合正規分布モデル
2段階の生成過程として理解できます

$$P(\mathbf{x}) := \sum_{k=1}^K w^{(k)} g(\mathbf{x}; \mu^{(k)}, \Sigma^{(k)})$$

データ \mathbf{x} の生成過程を考えると

1. 確率 $\{w^{(1)}, w^{(2)}, \dots, w^{(K)}\}$ (ただし $\sum_{k=1}^K w^{(k)} = 1$) を使って、どの正規分布からデータを生成するか決める
2. k 番目の正規分布 $g(\mathbf{x}; \mu^{(k)}, \Sigma^{(k)})$ から \mathbf{x} を生成する



25

THE UNIVERSITY OF TOKYO

ここまでのまとめ：
機械学習の2つの問題設定と代表的モデル

機械学習の代表的なタスクは2つある

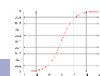
教師無し学習

- 入力を見ているうちに、どんなものが現れやすいかなどのパターンが分かってくる
- 実際には、入力の確率分布の推定問題として扱われる
- 代表的なモデル：混合正規分布



教師付き学習

- 入力に対する出力を試行錯誤するうちに、どういう入力のときにどういう出力をすればよいかがわかってくる
- 実際には、入力が与えられたときの出力の条件付確率分布の推定問題として扱われる
- 代表的なモデル：ロジスティック回帰



26

THE UNIVERSITY OF TOKYO

機械学習の応用

27

THE UNIVERSITY OF TOKYO

機械学習の応用：
実際、いろいろ役に立ちます

応用

- 信用リスク評価（教師付き学習）
- テキスト分類（教師付き学習）
- 画像認識（教師付き学習）
- 異常検知（教師無し学習）
- クラスタリング（教師無し学習）

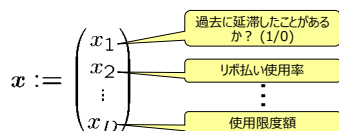
28

THE UNIVERSITY OF TOKYO

教師付き学習の応用例：信用リスク評価
「この人にお金貸して、返ってくるんだろうか？」

ある顧客に、融資を行ってよいか

- 顧客 \mathbf{x} を、さまざまな特徴を並べたベクトルで表現
- 融資を行ってよいか y
 - 融資を行ってよい（返済してくれる）：+1
 - 融資してはいけない（貸し倒れる）：-1
- マーケティングの文脈では、買ってくれる(+1),買ってくれない(-1)



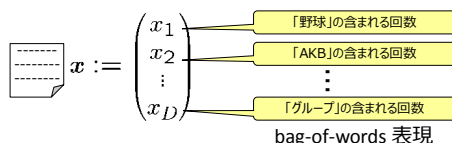
29

THE UNIVERSITY OF TOKYO

教師付き学習の応用例：テキスト分類
「あのタレントの不祥事、世間の評判はどうだろう？」

自然言語の文書が、あるカテゴリに入るかどうか



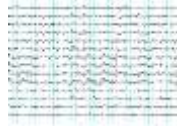
- 文書 \mathbf{x} を、含まれる単語ベクトルで表現
- (たとえば) ある事柄に好意的かどうか y
 - 好意的：+1
 - 否定的：-1
- トピック y ：「スポーツ」「政治」「経済」…（多クラス分類）



30

THE UNIVERSITY OF TOKYO


教師付き学習の応用例：画像認識、脳波解析、...
「これ、何て書いてあるの?」「いま何考えてる?」

- 手書き文字認識
 
 → {ある文字が(+1)否か(-1)
どの文字か? {"0","1","2", ...}}
- BCI (Brain Computer Interface)
 
 → 
 → どちらを思い浮かべている?
右(+1)? 左(-1)?
- ほか、顔画像認識や、動画認識

31 THE UNIVERSITY OF TOKYO

教師なし学習の応用例：異常検知
「ちょっと出かけてくるけど、ヤバそうだったら教えて」

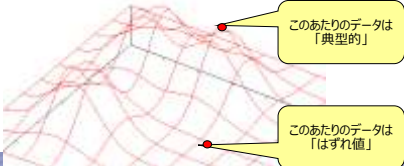
- 機械システム/コンピュータシステムの異常を、なるべく早く検知したい
 - 早い段階で検出できれば、それだけコスト減
- システムに仕込まれたセンサーからの取得データを分析する
 - システムの異常/変化、不正な操作により、システムが通常とは異なった振る舞いをするようになる
 - 計測機器の異常によって、通常とは異なった計測値が得られるようになる



32 THE UNIVERSITY OF TOKYO

教師なし学習の応用例：異常検知
確率の低いデータ = 異常 と考えます

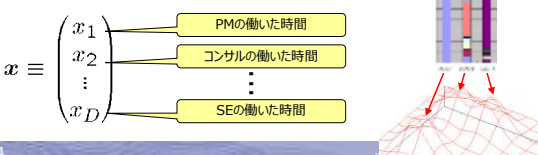
- システムの状態をベクトル \mathbf{x} で表現し、教師無し学習による確率分布 $P(\mathbf{x})$ の推定を行う
 - コンピュータ間の通信量、各コマンドやメッセージ頻度
 - 各センサーの計測値の平均、分散、センサー同士の相関
- $P(\mathbf{x})$ の小さいデータ \mathbf{x} は「めったに起こらない状態」
 - = システム異常、不正操作、計測機器故障などの可能性がある



33 THE UNIVERSITY OF TOKYO

教師なし学習の応用例：クラスタリング（人材管理での例）
「とりあえず、同じような塊に分けて整理しといて」

- プロジェクトには様々な職種の人々が様々な配分でかわかる
 - プロジェクト・マネージャ、コンサルタント、ソフトウェア・エンジニア、アーキテクト、...
- 実際のプロジェクトで使われた人的リソース配分を \mathbf{x} として、混合分布による教師無し学習を行う
- 混合分布の各分布の中心が典型的な人材配置のテンプレートを作成
- クラスタリング：グループを発見する



34 THE UNIVERSITY OF TOKYO

ここまでのまとめ：機械学習にはさまざまな応用がある

- 紹介した応用：信用リスク評価、テキスト分類、画像認識、異常検知、クラスタリング
- データあるところには、学習の問題がほぼ確実にある
 - 教師付き学習では1%の予測性能改善が、収益に直結する
 - 異常検出の需要は、コストのかかるシステムを抱える組織ならば常に存在する
- まだまだビジネスの現場において、機械学習（先進的なBI）が十分に入り込んでいない

35 THE UNIVERSITY OF TOKYO

機械学習問題の定式化

36 THE UNIVERSITY OF TOKYO

学習の定式化：
機械学習の問題をどのように数理的に定式化するか？

- 機械学習の問題を数理的に扱うために、まず、学習の対象と、学習するべきモデルを表現した
 - 対象：ものごとを実数値ベクトルで表現した
 - モデル：その上での確率モデルを考えた
- また、教師付き／教師無しという学習の2つの目的を定義した
- つぎに、その目的を、最適化問題として定式化する
 - 最尤推定による最適化問題としての定式化

37

THE UNIVERSITY OF TOKYO

最尤推定：モデル推定問題のもっとも標準的な定式化です
データを最もよく再現するパラメータを求める最適化問題として定式化します

- 最尤推定の基本的な考え方：訓練データを最もよく再現するパラメータが良いパラメータとする
 - 訓練データを最もよく再現する = 最も高い確率を与える
- 訓練データが互いに独立であるとする、その同時確率は
教師無し学習なら $\prod_{i=1}^N P(\mathbf{x}^{(i)})$ 、教師付き学習なら $\prod_{i=1}^N P(y^{(i)} | \mathbf{x}^{(i)})$
で与えられる 「尤度」とよぶ
- これ（の対数）を最大にするパラメータを求める
教師無し学習の場合 $L := \sum_{i=1}^N \log P(\mathbf{x}^{(i)})$
教師付き学習の場合 $L := \sum_{i=1}^N \log P(y^{(i)} | \mathbf{x}^{(i)})$
「対数尤度」とよぶ
- モデル推定の問題が、対数尤度を目的関数とした最適化問題として捉えられる

38

THE UNIVERSITY OF TOKYO

最尤推定：モデル推定問題のもっとも標準的な定式化です
教師付き/教師なしそれぞれで実際例を見てみます

- 目的：訓練データから、モデルのパラメータを推定する
- 混合正規分布（教師無し学習）の場合
 - 訓練データ $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}, \dots, \mathbf{x}^{(N)})$
 - パラメータ $P(\mathbf{x}) := \sum_{k=1}^K w^{(k)} g^{(k)}(\mathbf{x})$

$$g^{(k)}(\mathbf{x}) := \frac{1}{(2\pi)^{D/2} |\Sigma^{(k)}|^{1/2}} \exp\left(-(\mathbf{x} - \mu^{(k)})^T \Sigma^{(k)-1} (\mathbf{x} - \mu^{(k)})\right)$$
- ロジスティック回帰（教師付き学習）の場合
 - 訓練データ $((\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), (\mathbf{x}^{(3)}, y^{(3)}) \dots, (\mathbf{x}^{(N)}, y^{(N)}))$
 - パラメータ $P(y = +1 | \mathbf{x}) := \sigma(\mathbf{w}^T \mathbf{x})$

39

THE UNIVERSITY OF TOKYO

最尤推定の例：多次元正規分布（教師なし学習）
最適なパラメータは、閉じた形で求められます

- 多次元正規分布の最尤推定（平均のみ。共分散行列は定数とする）

$$P(\mathbf{x}; \mu) := \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} \exp\left(-(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)\right)$$
- 対数尤度は

$$L := \sum_{i=1}^N \log P(\mathbf{x}^{(i)}; \mu)$$

$$= \sum_{i=1}^N \left(-(\mathbf{x}^{(i)} - \mu)^T \Sigma^{-1} (\mathbf{x}^{(i)} - \mu)\right) + \text{const.}$$
- μ で微分 $\frac{\partial L}{\partial \mu} = \sum_{i=1}^N 2\Sigma^{-1}(\mathbf{x}^{(i)} - \mu) = 2\Sigma^{-1} \sum_{i=1}^N (\mathbf{x}^{(i)} - \mu)$
- $= 0$ と置いて解くと、

$$\mu = \frac{\sum_{i=1}^N \mathbf{x}^{(i)}}{N}$$
 データの平均になった

40

THE UNIVERSITY OF TOKYO

ここまでのまとめ：機械学習の問題は最尤推定によって
最適化問題として定式化されます

- 最尤推定：
「訓練データを、最もよく再現するパラメータが良いパラメータとする」に基づいて学習を行う
- 対数尤度を目的関数として、パラメータについての最大化を行う
- 多次元正規分布の平均パラメータの最尤推定による推定値は、データの平均によってもとまる
 - ちなみに、共分散行列の推定値は、データの共分散行列によって求まる
- もっと複雑なモデル（混合正規分布、ロジスティック回帰）では、最尤推定はどのように行えばいいだろうか？

41

THE UNIVERSITY OF TOKYO

機械学習のアルゴリズム

42

THE UNIVERSITY OF TOKYO

学習のアルゴリズム：
対数尤度を最大化するパラメータを数値的に求めます

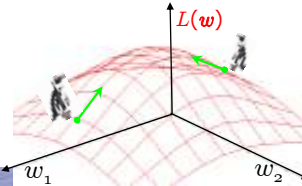
- 必ずしも正規分布のように閉じた形で解が求まるわけではない
- 最尤推定を数値的に行うためのアルゴリズム
 - 勾配法 → ロジスティック回帰
 - EMアルゴリズム → 混合正規分布
- さらに、大規模なデータを用いた学習を、効率的に行うための方法
 - オンライン学習アルゴリズム

43

THE UNIVERSITY OF TOKYO

勾配法：もっとも基本的な最適化法
目的関数が最も急な方向にパラメータ更新を繰り返します

- 山（対数尤度 L ）の頂点を（なるべく速く）目指したい
- もっとも坂が急な方向に向かって（パラメータ上で）1m進む
 - 頂上付近だと、頂上を越えて向こう側についてしまうことも
 - 実際には歩幅はだんだん小さくする



44

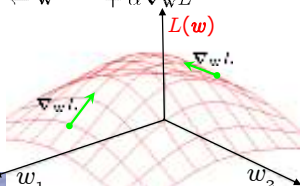
THE UNIVERSITY OF TOKYO

勾配法：もっとも基本的な最適化法
最急方向（勾配）は対数尤度の偏微分で求めます

- もっとも急な方向 = 勾配
- 勾配は、目的関数（対数尤度） $L(w)$ のパラメータ w での偏微分

$$\nabla_w L \equiv \left(\frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, \dots, \frac{\partial L}{\partial w_D} \right)$$
- 勾配の方向に、少し（正の定数 α ）更新する

$$w^{NEW} \leftarrow w^{OLD} + \alpha \nabla_w L$$



45

THE UNIVERSITY OF TOKYO

ロジスティック回帰に対する勾配法：
勾配を実際に計算してみます

- 条件付分布の対数尤度の勾配を求める
- カテゴリ+1 の訓練データのインデックス集合をPos, カテゴリ-1 のインデックス集合をNegとすると、対数尤度は、

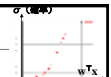
$$L(w) \equiv \sum_{i=1}^N \log P(y^{(i)} | x^{(i)}; w) = \sum_{i \in \text{Pos}} \log \sigma(w^T x^{(i)}) + \sum_{i \in \text{Neg}} \log(1 - \sigma(w^T x^{(i)}))$$

- 勾配は、

$$\frac{\partial L(w)}{\partial w} = \sum_{i \in \text{Pos}} \frac{\partial \log \sigma(w^T x^{(i)})}{\partial w} + \sum_{i \in \text{Neg}} \frac{\partial \log(1 - \sigma(w^T x^{(i)}))}{\partial w}$$

+1カテゴリのデータ

-1カテゴリのデータ



46

THE UNIVERSITY OF TOKYO

ロジスティック回帰に対する勾配法：
比較的シンプルに求められます

- 勾配は、

$$\frac{\partial L(w)}{\partial w} = \sum_{i \in \text{Pos}} \frac{\partial \log \sigma(w^T x^{(i)})}{\partial w} + \sum_{i \in \text{Neg}} \frac{\partial \log(1 - \sigma(w^T x^{(i)}))}{\partial w}$$
- ここで、 $\sigma(a) := \frac{1}{1 + e^{-a}}$ 、 $(1 - \sigma(a)) := \frac{e^{-a}}{1 + e^{-a}}$ より

$$\log \sigma(a) = -\log(1 + e^{-a}) \Rightarrow \frac{\partial \log \sigma(a)}{\partial a} = \frac{e^{-a}}{1 + e^{-a}} = 1 - \sigma(a)$$

$$\log(1 - \sigma(a)) = -a - \log(1 + e^{-a}) \Rightarrow \frac{\partial \log(1 - \sigma(a))}{\partial a} = -\sigma(a)$$
- 結局、

$$\frac{\partial L(w)}{\partial w} = \sum_{i \in \text{Pos}} (1 - \sigma(w^T x^{(i)})) x^{(i)} - \sum_{i \in \text{Neg}} \sigma(w^T x^{(i)}) x^{(i)}$$

47

THE UNIVERSITY OF TOKYO

EMアルゴリズム：「隠れ変数」が考えられるときの最適化法
混合分布を効率的に推定できます

- 混合正規分布の最尤推定も、勾配法で行ってよいが...
- EM (Expectation-Maximization) アルゴリズム
 - 本来なかった「隠れ変数」の存在が自然に導入できるようなモデルの最尤推定法
 - 混合正規分布は、正規分布をひとつ選んで、データを生成していると考えられる
 - 「どのデータがどの正規分布から発生したか」を「隠れ変数」として導入
 - 2つのステップの繰り返しアルゴリズム
 - 隠れ変数を固定したときのパラメータの最尤推定
 - 単一の正規分布の最尤推定は、閉じた形で求まる
 - パラメータを固定したときの隠れ変数の推定

48

THE UNIVERSITY OF TOKYO

混合正規分布のためのEMアルゴリズム：
メンドウなので、K-meansアルゴリズムを紹介しします

- 混合正規分布 ($\Sigma^{(k)}$ は固定)

$$P(\mathbf{x}; \{w^{(k)}\}, \{\mu^{(k)}\}) := \sum_{k=1}^K w^{(k)} g^{(k)}(\mathbf{x}; \mu^{(k)})$$

$$g^{(k)}(\mathbf{x}; \{\mu^{(k)}\}) := \frac{1}{(2\pi)^{D/2} |\Sigma^{(k)}|^{1/2}} \exp \left(-(\mathbf{x} - \mu^{(k)}) \Sigma^{(k)-1} (\mathbf{x} - \mu^{(k)}) \right)$$

において、対数尤度 \downarrow を最大化するパラメータを求めたい

$$L := \sum_{i=1}^N \log P(\mathbf{x}^{(i)}; \{w^{(k)}\}, \{\mu^{(k)}\})$$

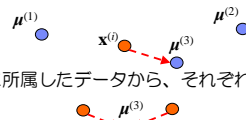
- 煩雑になるので、単純化して、
インチキEMアルゴリズム (K-meansアルゴリズム) を導くことにする

49

THE UNIVERSITY OF TOKYO

K-meansアルゴリズム：隠れ変数を考えることで、簡単な繰り返しアルゴリズムになります

- 混合正規分布は、正規分布の一つ選んで、それを使って \mathbf{x} を生成していると解釈できる
- 各データ $\mathbf{x}^{(i)}$ が、 K 個の正規分布のどれからでてきたのかはわからない。もしわかっていれば、平均によって正規分布のパラメータが推定できた $\mu = \sum_i \mathbf{x}^{(i)} / N$
- そこで、以下のステップを収束するまで繰り返す
 - 各データ $\mathbf{x}^{(i)}$ を、最寄の平均をもつ正規分布に所属させる



- 各正規分布に所属したデータから、それぞれの平均を新たに求める

50

THE UNIVERSITY OF TOKYO

学習アルゴリズムのオンライン化：
大規模なデータを扱うときに有効です

- 勾配法、EM法、ともに各繰り返しは、(データ数 N) \times (次元数 D) に比例した時間がかかる
- しかし、
 - データ数が非常に大きいときには、結構時間がかかる
 - 実際には、本当にキッチリ最適化する必要も無い
(モデルもホントかどうか...)
 - 時間とともにデータが到来するような場合もある
 - 時間とともに、正解のモデルも変化するかもしれない
- そこで、オンライン学習 (逐次学習) アルゴリズム：
訓練データをひとつずつ処理する
 - 人間の学習のイメージにちかい (「だんだん」「試行錯誤」)

51

THE UNIVERSITY OF TOKYO

ロジスティック回帰の勾配法のオンライン化：
ひとつのデータに対しての対数尤度の勾配を用います

- データ $(\mathbf{x}^{(i)}, y^{(i)})$ について注目して最適化を行う
- 1つのデータに注目したときの対数尤度

$$L^{(i)}(\mathbf{w}) := \log P(y^{(i)} | \mathbf{x}^{(i)}; \mathbf{w})$$

$$= \begin{cases} \log \sigma(\mathbf{w}^T \mathbf{x}^{(i)}) & \text{if } y^{(i)} = +1 \\ \log(1 - \sigma(\mathbf{w}^T \mathbf{x}^{(i)})) & \text{if } y^{(i)} = -1 \end{cases}$$

- 勾配の方向にパラメータを少し更新

$$\mathbf{w}^{\text{NEW}} \leftarrow \mathbf{w}^{\text{OLD}} + \alpha \nabla_{\mathbf{w}} L^{(i)}(\mathbf{w})$$

$$= \mathbf{w}^{\text{OLD}} + \alpha \begin{cases} (1 - \sigma(\mathbf{w}^T \mathbf{x}^{(i)})) \mathbf{x}^{(i)} & \text{if } y^{(i)} = +1 \\ -\sigma(\mathbf{w}^T \mathbf{x}^{(i)}) \mathbf{x}^{(i)} & \text{if } y^{(i)} = -1 \end{cases}$$

- 1ステップの計算量は $O(D)$

52

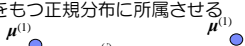
THE UNIVERSITY OF TOKYO

EMアルゴリズム (K-means) のオンライン化：
オンライン異常検知を行うためには必須です

- オンライン異常検知：データが時々刻々流れてくる中で
 - モデル推定 (モデルを逐次的に更新)
 - 異常検知 (おかしいデータを発見) を同時に行う

- 以下のステップを繰り返す

- $\mathbf{x}^{(i)}$ を、最寄の平均をもつ正規分布に所属させる (バッチ版と同じ)



- その最寄の平均を $\mathbf{x}^{(i)}$ に (ちょっと) 近づける

$$\mu^{\text{NEW}} \leftarrow (1 - \epsilon) \mu^{\text{OLD}} + \epsilon \mathbf{x}^{(i)}$$

$-\epsilon$ は正の小さい値

ここで、最寄の平均への距離が大きければ異常データと判断

53

THE UNIVERSITY OF TOKYO

これまでのまとめ：最尤推定のための数値計算アルゴリズム (勾配法とEMアルゴリズム)

- 最尤推定を数値的に行うためのアルゴリズム
 - 勾配法
 - EMアルゴリズム
- 大規模データを効率的に行うための方法としてオンライン学習アルゴリズム
 - ロジスティック回帰のオンライン化
 - K-meansアルゴリズムのオンライン化
 - オンライン異常検知
- 実際に学習してみたものの、その結果の良し悪しはどのように判断したらよいだろうか？

54

THE UNIVERSITY OF TOKYO

機械学習手法の評価

55

THE UNIVERSITY OF TOKYO

評価方法：何をもって学習の良し悪しを計るか？
まだ見ぬデータに対する性能が真の性能であると考えます

- 実際に学習してみたものの、その結果の良し悪しはどのように判断したらよいだろうか？
- モデルは、まだ見ぬデータに対してうまく働く必要がある
 - 教師無し学習：未知の入力 \mathbf{x} に対して高い確率を割り当てる
 - 教師付き学習：カテゴリ y が未知の入力 \mathbf{x} に対して正しいカテゴリを振る
- 訓練データとテストデータに分けて評価を行う
 - 訓練データ：モデルをつくるためのデータ
 - テストデータ：モデルの性能を評価するためのデータ
(=将来のデータとして、まだ見ていないことにする)

全データ 訓練用 評価用

56

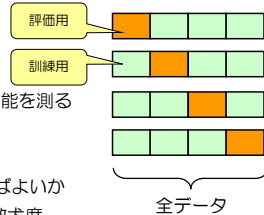
THE UNIVERSITY OF TOKYO

交差確認法（クロスバリデーション）：
まだ見ぬデータに対する性能を評価することができます

- 全体を K 等分し、
 - そのうち $K-1$ 個を訓練用に
 - 1個を評価用に使う

を K 回繰り返し、その平均的な性能を測る

- では、性能を測る指標として、
具体的にどのような指標を使えばよいか
 - 教師無し学習：（テスト）対数尤度
 - 教師付き学習：正解率、AUC



57

THE UNIVERSITY OF TOKYO

教師無し学習の場合、テストデータに対する対数尤度

- テストデータに対する対数尤度

$$\mathcal{L} := \sum_{\mathbf{x} \in \text{test set}} \log P(\mathbf{x}^{(i)})$$

- テストデータと訓練データが同じ分布から出ているのであれば、訓練データに対して高い確率を与えるモデルは、テストデータに対しても高い確率を与えるはず
- もしくは、クラスタリングなどの場合に、正しいクラスラベルが分かっていたりすれば、それとの一致具合も使われる

58

THE UNIVERSITY OF TOKYO

教師付き学習の場合、尤度より正解率のほうが評価値として好ましいが、評価値が閾値に依存するという問題があります

- 教師付き学習の場合にも、対数尤度を使ってよい
- が、本当はモデル $P(g = +1|\mathbf{x})$ の出力を用いて予測したカテゴリが正しいかどうかに関心がある
 - $P(g = +1|\mathbf{x}) > 0.5$ であれば、カテゴリ +1 と予測
 - $P(g = +1|\mathbf{x}) < 0.5$ であれば、カテゴリ -1 と予測
- 正解率 := (テストデータ中の正解数) / (テストデータの数)
- しかし、
 - 精度が閾値に依存してしまう
 - 特に、カテゴリのバランスが悪いときに
 - $P(g = +1|\mathbf{x})$ が 0.5 よりも全体的に高め／低めに出るので評価のベースラインがわからない

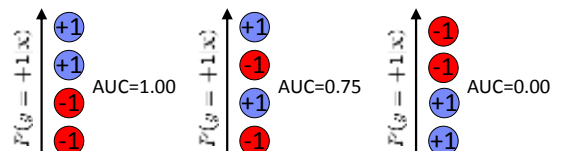
59

THE UNIVERSITY OF TOKYO

AUC: 閾値に依存しない教師付き学習の定番評価値

- ある閾値を決めたときの正解率ではなく、
 $P(g = +1|\mathbf{x})$ の相対的な順序に依存した指標
- AUC (Area Under the Curve) とは：
 - あるカテゴリ +1 の $\mathbf{x}^{(i)}$ をランダムに選び
 - あるカテゴリ -1 の $\mathbf{x}^{(j)}$ をランダムに選んだとき
 - $P(y^{(i)} = +1|\mathbf{x}^{(i)}) > P(y^{(j)} = +1|\mathbf{x}^{(j)})$ であるような確率

ちなみに、金融では、ほぼ同様の指標がAR値と呼ばれる

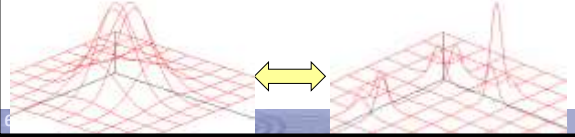


60

THE UNIVERSITY OF TOKYO

過学習：訓練データに適応しすぎると、性能が悪くなる現象

- （教師付き学習において）訓練データそのものを覚えてしまえば、訓練データに関しては100%正解できる
 - しかし、本当は、訓練データに含まれていないデータに対して正解したい
- 訓練データに拘りすぎると、性能が悪くなる現象「過学習」がおこる
 - とくに \mathbf{x} が高次元のデータを扱うときに起こる
 - データの数に対して、モデルの自由度（パラメータの数）が大きすぎると、おこりやすい



正則化：訓練データへの過適合を防ぐ方法

- 尤度だけではなく「関数の滑らかさ」を表す項を目的関数に加える
- 具体的には、パラメータのノルム $\|\mathbf{w}\|$ （ベクトルの大きさ）を使うことが多い
 - ノルムが大→極端なモデル
 - ノルムが小→滑らかなモデル

ロジスティック回帰（教師付き学習）の場合

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

$$L := \sum_{i=1}^N \log P(y^{(i)} | \mathbf{x}^{(i)}; \mathbf{w}) - \lambda \|\mathbf{w}\|$$

パラメータ \mathbf{w} に依存することを明示的にするためにこのように書く

ノルムがペナルティ項として加わる

λ は適当な正の定数（対数尤度とのバランスをとる）

l2-正則化（リッジ正則化）：もっとも一般的な正則化法

- ノルムとして2-ノルムを用いる

$$\|\mathbf{w}\|_2 := \|\mathbf{w}\|_2^2 = w_1^2 + w_2^2 + \dots + w_D^2$$
- これを対数尤度にペナルティ項として加えると

$$L := \sum_{i=1}^N \log P(y^{(i)} | \mathbf{x}^{(i)}; \mathbf{w}) - \lambda \|\mathbf{w}\|_2^2$$
- もっとも一般的に用いられる正則化法
- λ の決め方は後述

l1-正則化（ラッソ正則化）：スパース（疎）な解を得られる正則化法

- ノルムとして1-ノルムを用いる

$$\|\mathbf{w}\|_1 := |\mathbf{w}|_1 = |w_1| + |w_2| + \dots + |w_D|$$
- これを対数尤度にペナルティ項として加えると

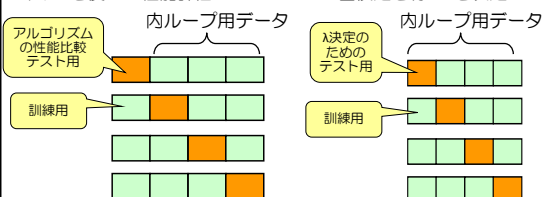
$$L := \sum_{i=1}^N \log P(y^{(i)} | \mathbf{x}^{(i)}; \mathbf{w}) - \lambda |\mathbf{w}|_1$$
- 得られる \mathbf{w} が疎になることが知られている
 - \mathbf{w} の要素の多くが0になる
- \mathbf{x} の次元が高いときに有効
 - テキスト分類など

ハイパーパラメータ λ の決定：交差検定を利用することで決定できます

- 交差検定によって決定する
- 複数の学習アルゴリズムの比較には、交差検定の2重ループ

外ループでは、内ループで決定された λ を使って性能評価

内ループでは、さらに交差検定を行い λ を決定



ここまでのまとめ：性能評価方法と過学習の問題、過学習を避けるための正則化法

- 性能評価の方法として、訓練データとテストデータに切り分けて複数の評価を行う交差確認法（クロスバリデーション）
- 評価値としては、
 - 教師無し学習：テスト尤度
 - 教師付き学習：正解率、AUC
- 訓練データに適合しすぎて、性能が悪化する過学習の問題
- 過学習の解決法として、パラメータのノルムをペナルティ項として目的関数に加える正則化法
 - l2正則化（リッジ正則化）：世界標準
 - l1正則化（ラッソ正則化）：次元削減の効果あり

発展的な話題

少し発展的な話題を2つ：

- 多クラス分類
 - 2クラス分類の一般化
- カーネル法
 - 近年注目されている学習手法

発展的な話題 1： 多クラス分類（教師付き学習）

- これまでは2クラス分類問題を考えていた
 - ロジスティック回帰の出力は $y \in \{+1, -1\}$ の2値
- K クラスへの対応をするにはどうしたらよいだろうか
 - たとえば $\{A, B, C, D, E\}$ の5クラス
 - 例：文字認識、文書分類
- 主に2つのアプローチがとられる：
 - 2クラス分類問題に帰着する
 - 多クラス版の分類モデルを直接考える

多クラス分類に対する最もシンプルなアプローチは 「2クラス分類への帰着」です

- すでに2クラス分類問題については解法が分かっているから、2クラス分類に帰着できれば色々都合がよいはず
- アプローチ1: 1対多(one-versus-rest)方式
 - あるクラスが、そうでないか、という分類問題を K 個考える
 - K 個のモデルができる
 - 予測時にはもっとも確率の高いクラスに予測する
- アプローチ2: 1対1(one-versus-one)方式
 - 全ての2つのクラスの組について、2クラス分類問題を考える ($K(K-1)$ 個の2値分類問題)
 - $K(K-1)$ 個のモデルができる
 - 予測時には、それぞれのクラスへの所属確率の和で多数決をとる

1対多方式と1対1方式の中間的な2クラス分類帰着方法として「誤り訂正出力符号方式」があります

- 1対多方式では表現力不足、1対1方式ではモデルが多く（クラス数の2乗）なりすぎる場合がある
- アプローチ1.5：誤り訂正出力符号 (error correcting output code; ECOC) 方式
 - クラスを適当に2グループに分けた、2クラス分類問題を複数作る
 - 予測時には、予測時には、それぞれのクラスへの所属確率の和で多数決をとる

クラス	2値分類問題					
	1	2	3	4	5	6
A	1	1	1	1	1	1
B	1	-1	1	-1	-1	-1
C	-1	-1	-1	1	-1	1
D	-1	1	1	-1	-1	1

「誤り訂正出力符号方式」のポイント、うまくクラスが分離するようにグループ分けをするところだ

- 分類がうまくいくためにはグループ分けの表の行（「符号」とよぶ）がうまく分離している（ハミング距離が離れている）とよい

クラス	2値分類問題						クラス間のハミング距離				
	1	2	3	4	5	6	A	B	C	D	
A	1	1	1	1	1	1	A	0	4	3	D
B	1	-1	1	-1	-1	-1	B		0	4	3
C	-1	-1	-1	1	-1	1	C			0	3
D	-1	1	1	-1	-1	1	D				0

- 「符号」をうまく設計するところがポイント

多クラス分類を直接行うモデル： 多クラスロジスティック回帰

- ロジスティック回帰を多クラス版に拡張する
- 2クラスのロジスティック回帰

$$P(y = +1|x; w) := \frac{1}{1 + \exp(-w^T x)}$$

- パラメータ w は、 x の各次元のクラス+1への寄与分を表す
- 多クラスのロジスティック回帰

$$P(y = c|x; \{w_c\}_{c \in y}) := \frac{\exp(w_c^T x)}{\sum_c \exp(w_c^T x)}$$

実数値を正の値にマップ

足して1(確率)になるよう正規化

- クラス集合 \mathcal{Y} の各クラス c に対してパラメータベクトル w_c をもつ
- パラメータ w_c は、 x の各次元のクラス c への寄与分を表す

73

THE UNIVERSITY OF TOKYO

発展的な話題 2: カーネル法

- ロジスティック回帰のカーネル法化
- 表現定理：カーネル法の正当化

74

THE UNIVERSITY OF TOKYO

カーネル法は、非線形モデルを線形モデルのように扱える 近年話題の手法です

- ここ10年くらい機械学習の世界で研究が進んでいるモデル
- 理由は：
 - サポートベクトルマシン (svm) というモデルが各地で大成を収めた
 - データの見方を「特徴空間ビュー」から「類似度ビュー」に変換することで...
 - 高次元のデータに対しても適用できる(次元数→データ数)
 - 非線形なモデルの学習が行える
 - 木やグラフなどの非ベクトル的な対象を扱うことができる
- 多くのモデルが、カーネル法に変換することが出来る

75

THE UNIVERSITY OF TOKYO

ロジスティック回帰のカーネル化

- ロジスティック回帰モデル

$$P(y = +1|x) := \sigma(w^T x)$$

- 仮定：パラメータが、入力ベクトルの線形結合で表せるとする

$$w := \sum_{i=1}^N \alpha^{(i)} x^{(i)}$$

- $\alpha := (\alpha^{(1)}, \alpha^{(2)}, \dots, \alpha^{(N)})$ が新たなパラメータ

これを「カーネル化」という

- ロジスティック回帰モデルを書き直すと

$$P(y = +1|x) := \sigma \left(\sum_{i=1}^N \alpha^{(i)} \langle x^{(i)}, x \rangle \right) := \sigma \left(\sum_{i=1}^N \alpha^{(i)} K(x^{(i)}, x) \right)$$

- $\langle \cdot, \cdot \rangle$ は内積

- $K(\cdot, \cdot) := \langle \cdot, \cdot \rangle$ をカーネル関数と呼ぶ(内積を置き換えただけ)

76

THE UNIVERSITY OF TOKYO

カーネル化によって何が起ったか？ 高速化と非線形化

- カーネルロジスティック回帰モデル

$$P(y = +1|x) := \sigma \left(\sum_{i=1}^N \alpha^{(i)} K(x^{(i)}, x) \right) \quad \text{ただし、カーネル関数} \quad K(x^{(i)}, x) := \langle x^{(i)}, x \rangle$$

内積

- カーネル化によって
 - モデルのパラメータが D 個(次元数)から N 個(データ数)になった
 - データアクセスがカーネル関数(内積)を通じてのみ行われるようになった
- つまり
 - $N < D$ のときに速い。特に、カーネル関数の計算が x の次元よりも小さいオーダーであるときに速い
 - x が何だかよく分からない対象であっても、類似度らしきものがカーネル関数として与えられてさえいれば一応動く

77

THE UNIVERSITY OF TOKYO

表現定理： なぜパラメータを線形結合で表してよいのか？に答えます

- カーネル化は、パラメータが入力ベクトルの線形結合で表されるという仮定

$$w := \sum_{i=1}^N \alpha^{(i)} x^{(i)}$$

に基づくが、果たしてこれは正しいのか？

- 答え：l2正則化(リッジ正則化)ならば正しい
 - リプレゼンタ定理(表現定理)によって保証される
 - ちなみに、正則化の項は

$$\|w\|_2^2 = \sum_{i=1}^N \sum_{j=1}^N \alpha^{(i)} \alpha^{(j)} K(x^{(i)}, x^{(j)})$$

- l1正則化(ラッソ正則化)ではこれは保証されない
 - ひとつのアドホックな解決法としては $\|\alpha\|_1$ を使う

78

THE UNIVERSITY OF TOKYO

表現定理の証明

$$\mathbf{w} := \sum_{i=1}^N y^{(i)} \mathbf{x}^{(i)}$$

- 目的関数 $L := \sum \log P(y^{(i)} | \mathbf{x}^{(i)}; \mathbf{w}) - \lambda \|\mathbf{w}\|_2^2$ を最大化するパラメータを \mathbf{w}^* と置く
- \mathbf{w}^* を線形結合で表現できる部分 \mathbf{w} とそれ以外の部分 (どの $\mathbf{x}^{(i)}$ とも直交する) \mathbf{w}' に分ける $\mathbf{w}^* = \mathbf{w} + \mathbf{w}'$

$$\begin{aligned} L &:= \sum_{i=1}^N \log P(y^{(i)} | \mathbf{w}^{*\top} \mathbf{x}^{(i)}) - \lambda \|\mathbf{w}^*\|_2^2 \\ &= \sum_{i=1}^N \log P(y^{(i)} | \mathbf{w}^\top \mathbf{x}^{(i)}) - \lambda (\|\mathbf{w}\|_2^2 + \|\mathbf{w}'\|_2^2) \end{aligned}$$

1) パラメータと入力ベクトルの積に依存することを明示

2) \mathbf{w}' は入力ベクトルとはすべて直交するので、 \mathbf{w}' に依存する部分が消える

3) これは対数尤度とは関係ないから、勝手に最小化 (= 0) してよい。
よって、 $\mathbf{w}^* = \mathbf{w}$

79

THE UNIVERSITY OF TOKYO

カーネル関数：データ間の類似度を定義する関数 これがあればカーネル法は動く

- カーネル関数とは、2つの特徴ベクトルの内積

$$K(\mathbf{x}, \mathbf{x}') := \langle \mathbf{x}, \mathbf{x}' \rangle$$
 - 内積 = ある種の類似度と考えることができる
- カーネル法は、内積 (カーネル関数) として解釈できる類似度関数が与えられていれば動く
- 適切なカーネル関数さえ定義できれば、
 - (暗に) 高次元の特徴ベクトル
 - 文字列、木、グラフなどの非ベクトル型データ
 に対しても適用可能

80

THE UNIVERSITY OF TOKYO

高次元空間におけるカーネル関数の例：多項式カーネル

- カーネル関数を2つの特徴ベクトル $\mathbf{x} = (x_1, x_2)^\top$ と $\mathbf{x}' = (x'_1, x'_2)^\top$ の内積とする

$$K(\mathbf{x}, \mathbf{x}') := \langle \mathbf{x}, \mathbf{x}' \rangle = x_1 x'_1 + x_2 x'_2$$
- このカーネル関数を2乗したカーネル関数を考えてみると

$$\begin{aligned} K^2(\mathbf{x}, \mathbf{x}') &= \langle \mathbf{x}, \mathbf{x}' \rangle^2 \\ &= x_1^2 x_1'^2 + x_2^2 x_2'^2 + 2x_1 x_2 x'_1 x'_2 \\ &= \langle (x_1^2, x_2^2, \sqrt{2}x_1 x_2)^\top, (x_1'^2, x_2'^2, \sqrt{2}x'_1 x'_2)^\top \rangle \end{aligned}$$
- 特徴の組み合わせの項 $\sqrt{2}x_1 x_2$ が登場し、特徴ベクトルの次元が3になったにもかかわらず、計算量は元々の次元数(2)に依存
- 一般に、 d 乗することで d 個の特徴の組み合わせを実現できる

81

THE UNIVERSITY OF TOKYO

非ベクトル型データに対するカーネル関数

- 文字列、木、グラフなど、予め特徴ベクトルで表されていないようなデータを扱う方法は自明でない
 - いままでの議論は「特徴ベクトル」ありきであった
- どのようにしたらよいか？
 - カーネル法なら、とりあえずカーネル関数 (= 2つの非ベクトル型データの間の類似度) さえ定義できれば動くはず

82

THE UNIVERSITY OF TOKYO

ここまでのまとめ：

- 多クラス分類：
 - 2クラスの分類を組み合わせれば、多クラスの分類問題を解ける
 - 1対多方式、1対1方式、誤り訂正出力符号方式
 - 多クラス用の分類モデルを直接設計することもできる
 - 多クラスロジスティック回帰
- カーネル法：
 - データの見方を「特徴空間ビュー」から「類似度ビュー」に変換することで、高次元のデータを扱うようにする方法
 - 高次元のデータに対しても適用できる (次元数 → データ数)
 - カーネルの定義によっては非線形なモデルの学習が行える
 - カーネル化を正当化するための表現定理
 - パラメータについて線形なモデルに、 L_2 (リッジ) 正則化を適用する場合、成立する

83

THE UNIVERSITY OF TOKYO

構造をもったデータ

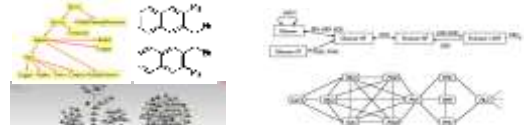
84

THE UNIVERSITY OF TOKYO

世の中には様々な「構造をもったデータ」があります

- 構造を持ったデータとは、データの構成要素とそれらの間の関係によって記述されるデータ：

- 配列：DNA、タンパク質、自然言語、イベント列、時系列
- 木構造：HTML/XML、RNA構造、構文解析木、系統樹、ディレクトリ
- グラフ/ネットワーク構造：化合物、画像、Web、社会ネットワーク、生体ネットワーク、...



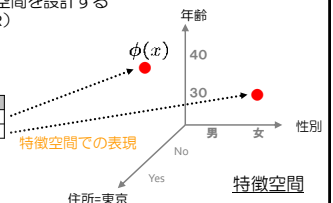
85

従来の機械学習手法ではベクトル型のデータを前提としており構造をもったデータをうまく扱うことが出来ません

- 従来のデータ解析手法では、データ x が特徴空間中のベクトル $\phi(x)$ として表されていることを前提とする
- 配列/木/グラフ/ネットワークなどの構造をもったデータでは一般的な特徴空間の構成は自明でなく、直接適用できない
- ドメインごとに独自に特徴空間を設計する
(例：化合物におけるQSAR)

従来：ベクトル型のデータ

観測番号	観測氏名	年齢	性別	住所	...
0001	O O	40代	男性	東京都	...
0002	X X	30代	女性	大阪府	...



86

THE UNIVERSITY OF TOKYO

「構造」には内部構造（グラフ）と外部構造（ネットワーク）の2種類があります

- 内部構造（グラフ）：

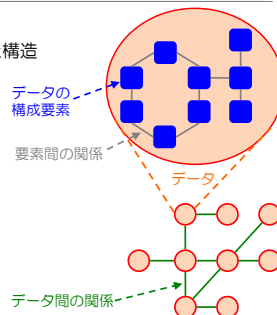
データ内の要素の関連を表した構造

- 半構造データ（HTML/XML）
- DNA配列
- 化合物

- 外部構造（ネットワーク）：

データ間の関連を表した構造

- Web/文献の参照ネットワーク
- 社会ネットワーク
- 遺伝子/タンパク質/薬剤の相互作用ネットワーク



87

THE UNIVERSITY OF TOKYO

内部構造（グラフ）の解析

88

THE UNIVERSITY OF TOKYO

内部構造の扱いについて、これからお話しすること

- （内部）構造データを扱う学習が、なぜ難しいか？
- 構造データ解析に便利な枠組み：カーネル法
- 順序木を扱うためのカーネル法
- グラフを扱うためのカーネル法

89

THE UNIVERSITY OF TOKYO

教師付き(分類)学習を前提にお話します

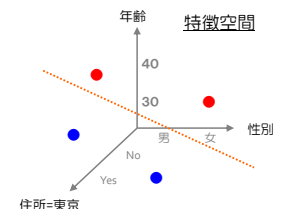
- 教師付き学習は、+1 か -1 の正解ラベルのついた訓練データ $\{(x^{(1)}, +1), (x^{(2)}, +1), (x^{(3)}, -1), (x^{(4)}, +1), \dots\}$

をもとに、ラベル未知のデータ x のラベルを予測する条件付き確率分布 $P(y|x; w)$ を学習する

- w はモデルパラメータ

- ロジスティック回帰モデル

$$P(y = +1|x; w) \equiv \sigma(w^T x)$$

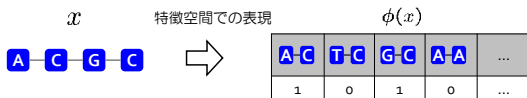


90

THE UNIVERSITY OF TOKYO

ひとつの自然なアプローチは「部分構造」を用いた特徴空間の構成ですが、計算量的な問題を抱えています

- 構造データの性質は、その「部分構造」が担っていると考える
 - 配列データの性質は、部分配列が担う（＝マルコフモデル）
- 各部分構造の有無や出現回数を用いて、特徴ベクトル表現する
- しかし、全ての部分構造をテーブル要素の候補にするときりが無い
 - グラフには、指数個の部分グラフが含まれるので、全てを数え上げている場所と時間はない（これが本質的な問題）

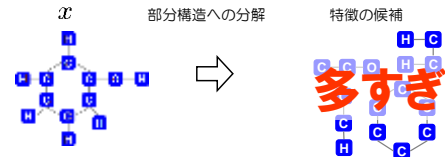


91

THE UNIVERSITY OF TOKYO

「部分構造多すぎ問題」に対するアプローチはいくつもあります

- グラフの部分構造は非常に多い
- 考えうるアプローチ：
 - 部分構造のサイズを限定する ⇒ マルコフモデル
 - 限られた重要なものだけを数え上げる ⇒ パターンマイニング
 - そもそもベクトル表現に持ち込まない ⇒ カーネル法



92

THE UNIVERSITY OF TOKYO

カーネル法

93

THE UNIVERSITY OF TOKYO

カーネル法は、データアクセスが特徴空間の次元に依存しないため、非ベクトルデータの扱いに向いています

- カーネル法：カーネル関数（＝特徴空間における内積）によってのみデータアクセスを行う学習器のクラス

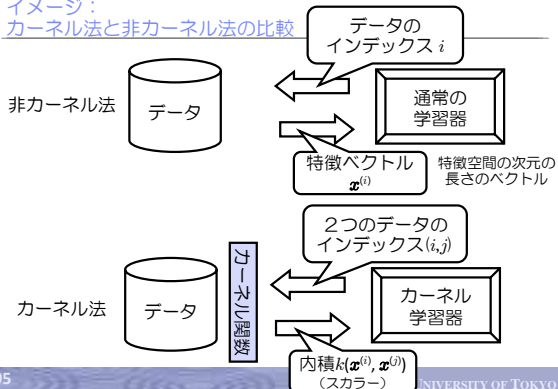
$$h(x) = \text{sign} \left(\sum_{i=1}^N \alpha_i K(x^{(i)}, x) \right)$$
 （カーネル予測器：パラメータ α ）

$$k(x, x') \equiv \langle x, x' \rangle$$
 （カーネル関数：類似度）
 - 特徴ベクトル x へのアクセスは常にカーネル関数 k を経由する
- ポイント：特徴ベクトル x が陽に現れないため、特徴ベクトルを陽に構成する必要が無い
 - データアクセス部分が次元に依存しない
- x や x' が非ベクトルデータでも、（内積であるような）適当な類似度 $k(x, x')$ を定義すればそれがカーネル関数として使える！

94

THE UNIVERSITY OF TOKYO

イメージ：
カーネル法と非カーネル法の比較



95

THE UNIVERSITY OF TOKYO

結局のところ、カーネル関数の設計がポイントです

- カーネル法とは：
 - モデル

$$h(x) = \text{sign} \left(\sum_{i=1}^N \alpha_i K(x^{(i)}, x) \right)$$
 - カーネル関数（＝2つのデータの類似度）

$$k(x, x') \equiv \langle x, x' \rangle$$
 が分離している機械学習法
- カーネル法は、良いカーネル関数さえ定義できれば、あとはおまかせで動く → ポイントはカーネル関数の設計
- 構造データに対する場合も自然に扱うことができる
 - 配列に対するカーネル関数
 - 木に対するカーネル関数
 - グラフに対するカーネル関数
 - ...

96

THE UNIVERSITY OF TOKYO

構造カーネル法

97

THE UNIVERSITY OF TOKYO

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



98

THE UNIVERSITY OF TOKYO

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 (注: 重なりがあってもよい)
- K_S : 部分構造間のカーネル関数



99

THE UNIVERSITY OF TOKYO

(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

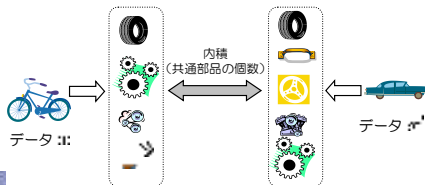


100

THE UNIVERSITY OF TOKYO

構造データのカーネル関数計算は、共通部分構造の数を数える問題に帰着されます

- 2つの構造データのカーネル関数 (= 部分構造で定義された特徴ベクトルの内積) を、特徴ベクトルを明示的に作らずに計算する
 - 対象とする構造データに対して、適切な部分構造を定義する
 - 木なら部分木、グラフならパス
 - カーネル関数の計算は、共通部分構造の個数を数える問題になる
 - 再帰構造を利用して、内積だけを効率的に計算

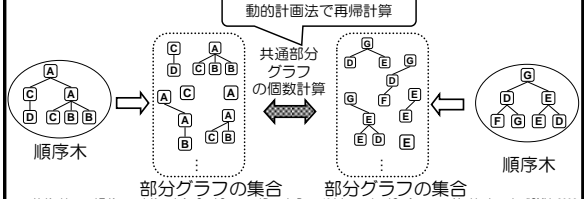


101

THE UNIVERSITY OF TOKYO

順序木に対するカーネル関数では、部分グラフを部分構造として用います

- 順序木カーネルでは、部分構造を部分グラフによって定義する
- 難しいところ: 部分木は指数個ありうる
 - 解決法: 数え上げの計算を動的計画法によって再帰計算することで計算可能になる



102

THE UNIVERSITY OF TOKYO

H. Kashima and T. Koyanagi: Kernels for Semi-Structured Data, In Proc. 19th International Conference on Machine Learning (ICML), 2002
重田 久雄, 坂本 比呂志, 小柳 光生: 半構造データに対するカーネル関数の設計と解析, 人工知能学会論文誌, Vol.21, No.1, 2006

順序木カーネルの計算のキモは、ノード対ごとのカーネル関数への分解です

- 特徴ベクトルを、部分構造の集合として部分木をつかって構成
 - 特徴ベクトル $\mathbf{x} \equiv (\text{①の出現回数}, \text{②の出現回数}, \dots)$
 - 特徴ベクトル $\mathbf{x}' \equiv (\text{①の出現回数}, \text{②の出現回数}, \dots)$
 - 内積 $k(\mathbf{x}, \mathbf{x}') \equiv (\mathbf{x}, \mathbf{x}')$ を計算したい

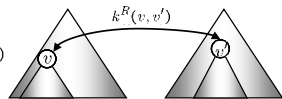
- カーネル関数 k は v と v' を根に持つ共通部分木の個数 k_R の和としてかくことができる

$$k(\mathbf{x}, \mathbf{x}') = \sum_{v \in V} \sum_{v' \in V'} k^R(v, v')$$

$$k^R(v, v') = \sum_{s \in S_v(x)} \sum_{s' \in S_{v'}(x')} \delta(s, s')$$

v を根に持つ部分木の集合

s と s' が同じレベルなら 1



103

THE UNIVERSITY OF TOKYO

ノード対ごとのカーネル関数は、再帰によって効率的に計算できます

- k^R は「 (v, v') についてボトムアップ」& 「 (v, v') の子について左から右」の2重ループの動的計画法によって $O(|V||V'|)$ で再帰計算

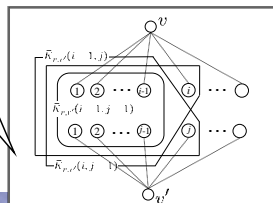
$$k^R(v, v') = \delta(\ell(v), \ell(v')) \cdot \bar{k}_{v,v'}^R(\text{ch}(v), \text{ch}(v'))$$

$$\bar{k}_{v,v'}^R(i, j) = \bar{k}_{v,v'}^R(i-1, j) + \bar{k}_{v,v'}^R(i, j-1) - \bar{k}_{v,v'}^R(i-1, j-1) + k^R(\text{ch}(v, i), \text{ch}(v', j))$$

(v, v') について

(v, v') の子同士について

(v, v') の子同士のマッチング数を全て数える

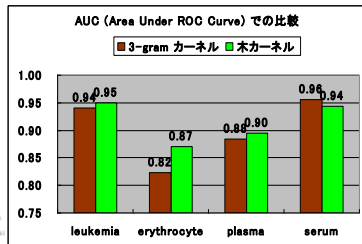
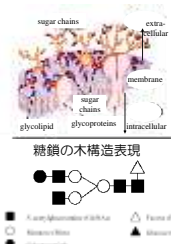


104

THE UNIVERSITY OF TOKYO

応用：木カーネルによる糖鎖の構造分類

- 糖鎖の構造から組織を予測する問題に適用、単純な方法（3-gramに基づく方法）を若干上回る性能が得られた



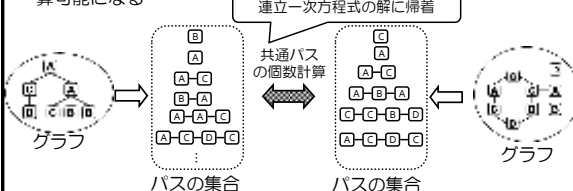
105

THE UNIVERSITY OF TOKYO

グラフに対するカーネル関数では、パスを部分構造として用います

- グラフカーネルでは、部分構造を、グラフ上のランダムウォークによって取り出されるパスとして定義する

- 難しいところ：パスは無限個ありうる
 ← 解決法：数え上げの計算を連立一次方程式に帰着することで計算可能になる



H. Kashino, K. Tsuda and A. Inokuchi: Marginalized Kernels Between Labeled Graphs. In Proc. 20th International Conference on Machine Learning (ICML), 2003
 H. Kashino, K. Tsuda and A. Inokuchi: Marginalized Kernels Between Labeled Graphs. In Kernel Methods in Computational Biology, MIT Press, 2004

106

THE UNIVERSITY OF TOKYO

グラフカーネル計算のキモもまた、ノード対ごとのカーネル関数への分解です

- ランダムウォークによる無限個のパスによって特徴ベクトルを構成
 - $\mathbf{x} \equiv (\text{①-②の出現回数} \times \lambda^2, \text{②-③の出現回数} \times \lambda^3, \dots)$
 - $\mathbf{x}' \equiv (\text{①-②の出現回数} \times \lambda^2, \text{②-③の出現回数} \times \lambda^3, \dots)$
 - 発散しないように、パスの長さによって指数的に減衰される
 - 内積 $k(\mathbf{x}, \mathbf{x}') \equiv (\mathbf{x}, \mathbf{x}')$ を計算したい

- カーネル関数 k は v と v' で終わる共通パスの個数 k_V の和としてかくことができる

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

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

ノード v で終わるパスの集合

107

THE UNIVERSITY OF TOKYO

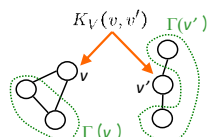
ノード対ごとのカーネル関数の計算は、連立方程式に帰着され効率的に行えます

- $k_V(v, v')$ は近隣ノード同士の k_V の和によって再帰的に書ける

$$K_V(v, v') = \lambda^2 \delta(\ell(v), \ell(v')) \left(1 + \sum_{\tilde{v} \in \Gamma(v)} \sum_{\tilde{v}' \in \Gamma(v')} \lambda^2 K_V(\tilde{v}, \tilde{v}') \right)$$

ノード v の隣接ノード集合

- これは連立方程式であるため多項式時間で解くことができる
 → 通常疎であるため、高速に解ける

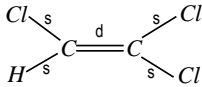


108

THE UNIVERSITY OF TOKYO

応用：グラフカーネルによる化合物の毒性予測

- 指数時間の他手法（パターンマイニング）に匹敵する性能が得られた



化合物のグラフ表現

パターンマイニング

Subtype	MSI	PSI	MMI	PII
0.0%	98.0%	97.0%	97.0%	98.0%
0.1%	91.0%	91.0%	91.0%	91.0%
0.2%	58.0%	55.0%	60.0%	60.0%
0.5%	58.0%	55.0%	60.0%	60.0%
1.0%	58.0%	55.0%	60.0%	60.0%
2.0%	41.0%	35.0%	50.0%	41.0%

Subtype	MSI	PSI	MMI	PII
0.1%	62.0%	61.0%	60.0%	60.0%
0.2%	62.0%	61.0%	60.0%	60.0%
0.3%	62.0%	61.0%	60.0%	60.0%
0.4%	62.0%	61.0%	60.0%	60.0%
0.5%	62.0%	61.0%	60.0%	60.0%
0.6%	62.0%	61.0%	60.0%	60.0%
0.7%	62.0%	61.0%	60.0%	60.0%
0.8%	62.0%	61.0%	60.0%	60.0%
0.9%	62.0%	61.0%	60.0%	60.0%

グラフカーネル

109

THE UNIVERSITY OF TOKYO

これらの研究はその後、さまざまな発展を遂げています
最近では（ほぼ）線形時間で動くようになっています

- 順序木カーネル：

- 曖昧マッチング[久保山ら, 2006]、部分構造を限定した高速化 [Kuboyama et al., 2006, 2007][Vishwanathan et al., 2003] 他
- 糖鎖構造分類への応用 [Kuboyama et al., 2006] 他

- グラフカーネル：

- ランダムウォークの設計による高精度化 [Mahe et al., 2004]、行列計算を利用した各種高速化 [Vishwanathan et al., 2006]、循環パターン [Horvath et al., 2004]、最短経路 [Borgwardt et al., 2005]、ハッシュ [Shervashidze et al., 2009][Hido et al. 2009]などによる近似を用いた高速化（線形時間）、ハイパーグラフへの一般化[Wachman et al., 2007] 他
- タンパク質立体構造分類 [Borgwardt et al., 2005]、画像検索への応用[Philipp-Foliguet et al., 2008] 他

110

THE UNIVERSITY OF TOKYO

ここまでのまとめ：

- 構造データ：
 - 内部構造（グラフ）と外部構造（ネットワーク）
- （グラフ）構造をもつデータに対するカーネル法アプローチ
 - 畳み込みカーネル
 - 木カーネル、グラフカーネル

111

THE UNIVERSITY OF TOKYO