



IBM Tokyo Research Laboratory

機械学習とその応用

censored

かしま ひさし
鹿島 久嗣

日本IBM東京基礎研究所

※ 実際に使ったものから、著作権が怪しい部分を省いたバージョンです

本講義の目的：機械学習（とIBM）の宣伝

- 機械学習が企業のビジネスにおいて差別化要因になることを知ってもらう
 - 「知っておくと得する（≒つぶしがきく）」という気持ちを伝える
- ちょっとやれば機械学習手法を使えそうだという気になってもらう
 - 機械学習の問題設定を知ってもらう
 - 機械学習のモデルを知ってもらう
 - 機械学習のアルゴリズムを知ってもらう
- （話の進み具合によっては）機械学習の先端の研究を垣間見てもらう
 - 構造データの分析（手前味噌）
(分かりやすさを優先するため、厳密さは犠牲にして、単純化して話をする)
- あと、
 - IBM基礎研究所を宣伝する

自己紹介：京大を卒業して、企業研究所で働いています

- 1997年（12年前）に京大・数理工学科卒業、
1999年（10年前）に応用システム科学専攻・修士課程修了
- 同年より日本IBM東京基礎研究所 研究員（現在、主任研究員）
 - 一環して、「機械学習」技術の基礎研究と、そのIBMビジネスへの応用
 - '99～'02年：バイオインフォマティクス
 - '03～'04年：コンピュータシステムの障害解析
 - '05年：研究所運営のサポート
 - '06年～：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
 - '07年～：センサー・データ解析（製造業のお客様）
 - '09年～：特許データの分析、ソーシャルネットワークの分析
 - 平行して
「構造データの機械学習」
の基礎研究
- 2007年に、京大大学院知能情報学専攻 博士課程修了
- 昨年は、ドイツの「マックスプランク研究所」に、3ヶ月間研究滞在

しばらく、IBMの宣伝が続くので、カットしました

プロジェクト内容等については直接お問い合わせください

講義の流れ（前半）：機械学習入門

- 機械学習とは
- 学習問題の区分
 - 教師つき学習と教師無し学習
 - 教師付き学習
 - ロジスティック回帰モデル
 - 教師無し学習
 - 混合分布モデル
- 機械学習の応用
 - 信用リスク推定
 - テキスト分類
 - 画像認識
 - 異常検知
 - クラスタリング
- 学習の定式化
 - 最尤推定
 - 多次元正規分布の最尤推定
- アルゴリズム
 - 勾配法
 - ロジスティック回帰モデルの勾配法
 - EMアルゴリズム
 - 混合正規分布のEM的アルゴリズム
 - 大規模データへの対応：オンラインアルゴリズム
- 機械学習手法の評価方法
 - 訓練データとテストデータ
 - 交差検定（クロスバリデーション）
 - 性能指標
 - テスト尤度、正解率、AUC
 - 過学習
 - 正則化
 - L1正則化とL2正則化
- 拡張
 - 多クラス分類
 - カーネル法
 - ロジスティック回帰のカーネル化
 - リプレゼンタ定理

機械学習とは：データ解析技術の一流派

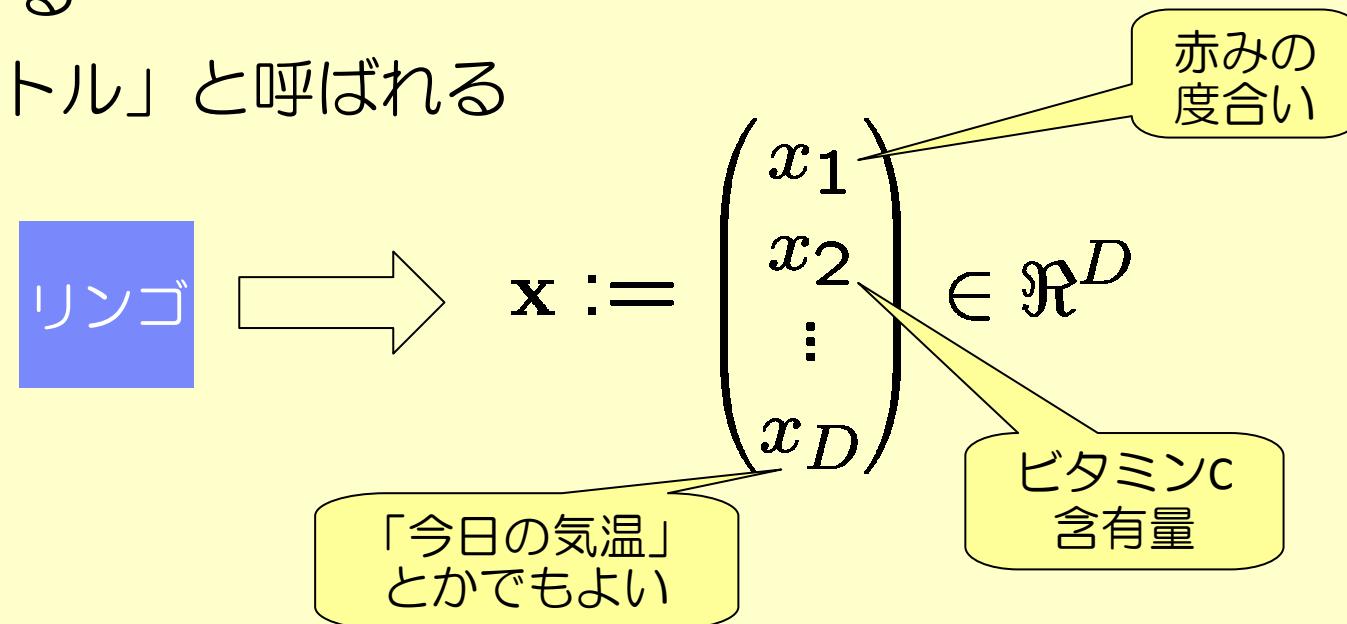
- 機械学習とは、人間のもつ「学習能力」を機械（計算機）にも持たせよう、という研究分野
 - もともとは人工知能の一分野として始まる
 - 論理推論がベース
 - 現在では、「統計的」機械学習が主流（≒機械学習）
 - 遺伝子情報処理、自然言語処理、他、ビジネス分野での成功
- 現在では、データ解析技術一般を指すほかの言葉とあまり変わらない
 - 統計／データマイニング／パターン認識など。
(多少のニュアンスの違いはあるが、基本的に好みの問題)

学習問題の区分：教師付き学習と教師無し学習

- 学習の問題の本質的な部分を、数学的に扱えるように、定式化する必要がある
- 学習者を、入出力のあるシステムであると捉える
- 学習者に対する入力と、それに対する出力の関係をモデル化
 - 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、入力に対する行動
- 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
 - 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つてくる

学習者に入ってくる入力情報を実数値ベクトルとして表現

- 入力信号を、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
 - 「特徴ベクトル」と呼ばれる



- \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、一般的な解はなく、目的にあわせて、人間がデザインする

教師無し学習は、確率分布の推定問題

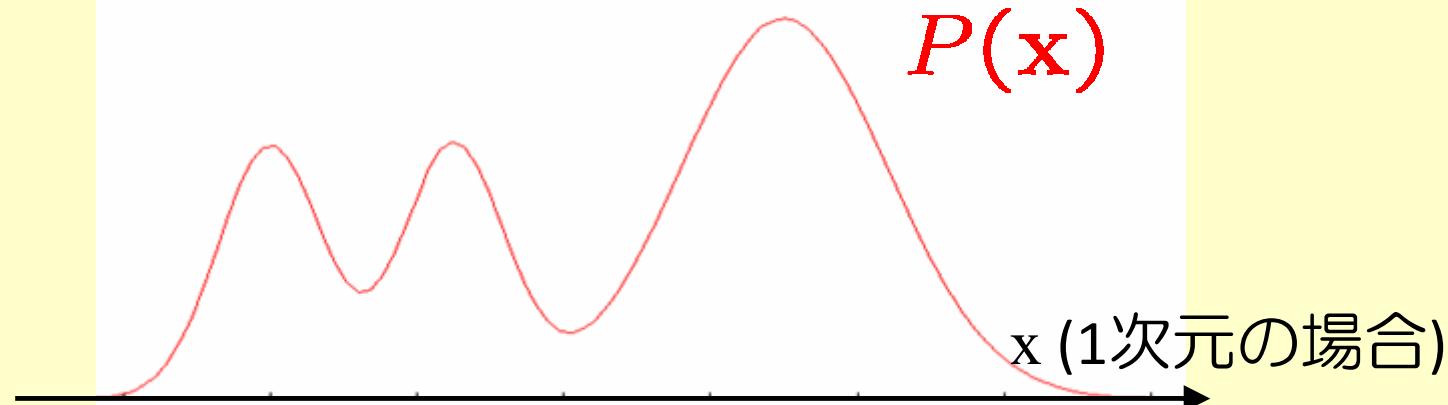
- 教師無し学習：たくさんの入力信号を見ているうちに、どんなものが現れやすいかなどのパターンが分かってくる

- 入力信号 \mathbf{x} : D 次元の実数値ベクトルとして表現

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

- この入力信号 \mathbf{x} 上の確率分布 $P(\mathbf{x})$ を考える

- この形をみるとことで、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



教師無し学習は、確率分布の推定問題

- 目的：訓練データ (N 個の入力信号) から、 $P(\mathbf{x})$ を推定する

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

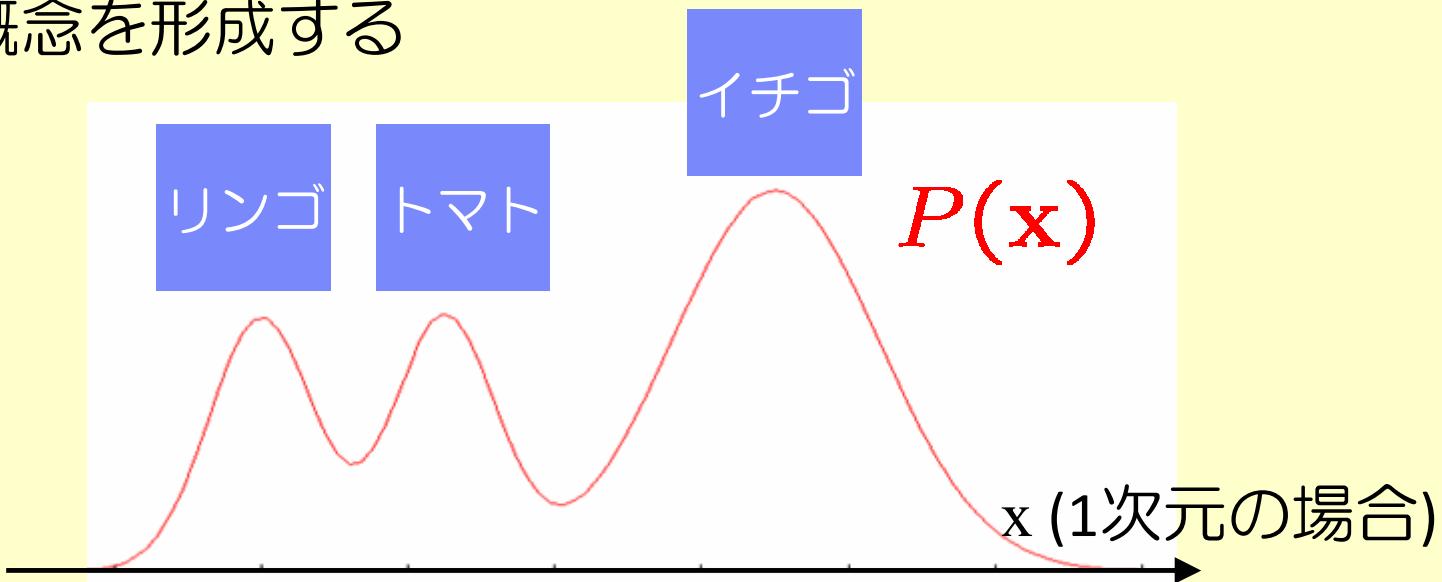
1つめの
データ

2つめの
データ

...

$$\mathbf{x}^{(i)} := \begin{pmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_D^{(i)} \end{pmatrix} \in \Re^D$$

- 「教師無し学習」（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



教師付き学習は、条件付確率分布の推定問題

- 条件付分布 $P(y|x)$: 入力信号 x を条件とした、出力 y の確率分布
 - 入力信号 x は、 D 次元の実数値ベクトル
 - 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：リンゴか否か)
 - > 複数のカテゴリ $\{A, B, C, D, \dots\}$
(例：リンゴかイチゴかトマトか)
 - 実数値 :
- たとえば、 $x := \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{pmatrix}$ がリンゴである確率はいくらか?
という質問に答える

教師付き学習は、条件付確率分布の推定問題

- 目的：訓練データ (N 個の入力信号) から、 $P(y|x)$ を推定する

$$((\underbrace{\mathbf{x}^{(1)}, y^{(1)}}_{\text{1つ目の入出力ペア}}, \underbrace{\mathbf{x}^{(2)}, y^{(2)}}_{\text{2つ目の入出力ペア}}, \underbrace{\mathbf{x}^{(3)}, y^{(3)}}_{\text{3つ目の入出力ペア}}, \dots, (\mathbf{x}^{(N)}, y^{(N)}))$$

- $\mathbf{x}^{(i)}$: i 番目の事例の入力信号ベクトル

- $y^{(i)}$: i 番目の事例に対する正しい出力

(リンゴならば +1, 違うなら -1)

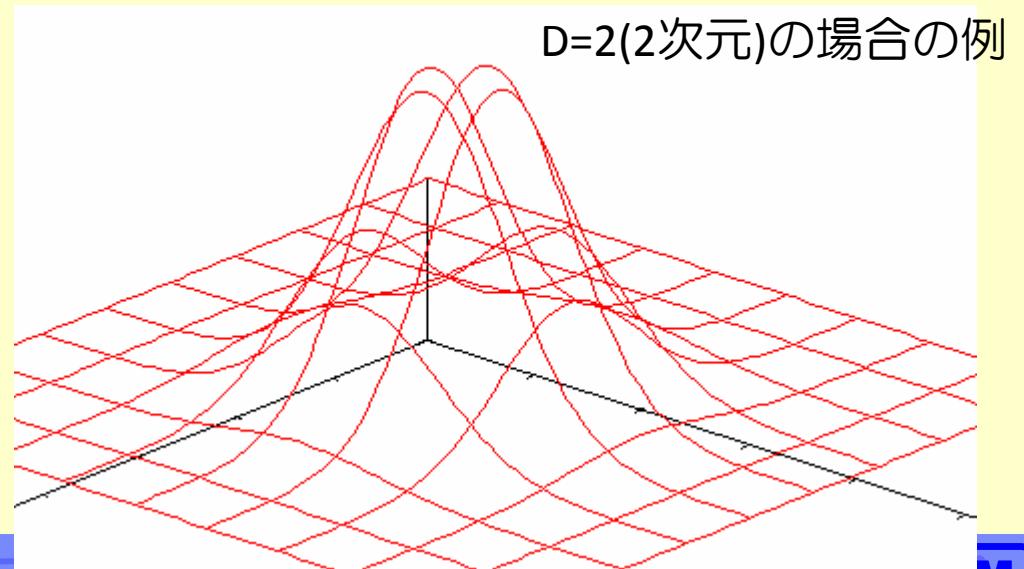
- 「教師付き学習」：与えられた入力信号に対する、あるべき出力を教師信号として、入出力の関係を学習する

教師なし学習モデルの典型：混合正規分布モデル

- 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 $g(\mathbf{x})$ を考える

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

- 1次元の正規分布 $g(\mathbf{x}; \mu, \sigma) := \frac{1}{(2\pi)^{1/2}\sigma} \exp\left(-\frac{(\mathbf{x} - \mu)^2}{2\sigma^2}\right)$ の拡張
- パラメータ
 - $\boldsymbol{\mu}$: 平均 (D 次元)
 - $\boldsymbol{\Sigma}$: 共分散行列 ($D \times D$)
- 単峰なので、表現力が不十分



教師なし学習モデルの典型：混合正規分布モデル

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

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

k 番目の正規分布
の重み

k 番目の正規分布

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

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

- パラメータ

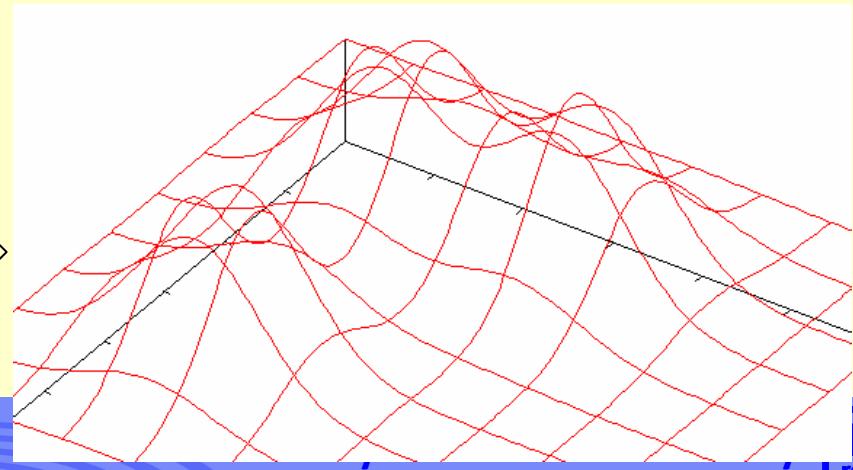
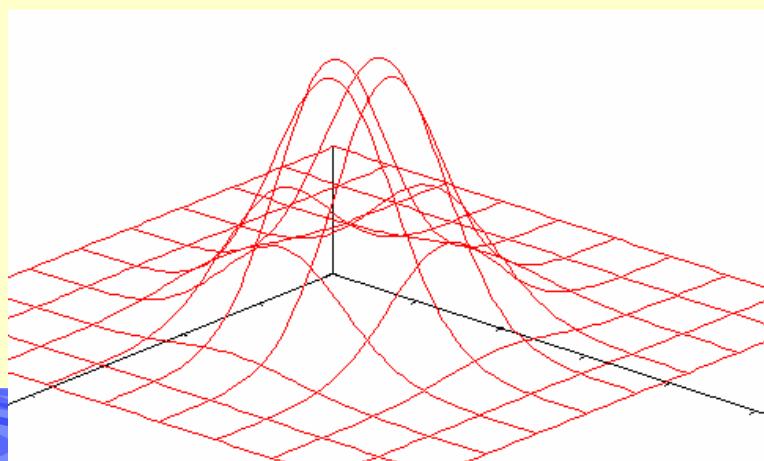
- 混合比パラメータ $\{w^{(k)}\}$
- 各正規分布のパラメータ $\{\boldsymbol{\mu}^{(k)}, \boldsymbol{\Sigma}^{(k)}\}$



教師なし学習モデルの典型：混合正規分布モデル

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

- 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 $g^{(k)}(\mathbf{x}; \boldsymbol{\mu}^{(k)}, \boldsymbol{\Sigma}^{(k)})$ から \mathbf{x} を生成する
- 単一の正規分布より複雑な確率分布を表現できる



教師付き学習モデルの典型：ロジスティック回帰モデル

- 出力が2カテゴリの場合の代表的な条件付確率モデル

$$P(y = +1|\mathbf{x}; \mathbf{w}) := \sigma(\mathbf{w}^\top \mathbf{x}) = \sigma(w_1 x_1 + w_2 x_2 + \dots + w_D x_D)$$

$$P(y = -1|\mathbf{x}; \mathbf{w}) = 1 - \sigma(\mathbf{w}^\top \mathbf{x})$$

- なお、 \mathbf{w} はモデルを定めるパラメータベクトル

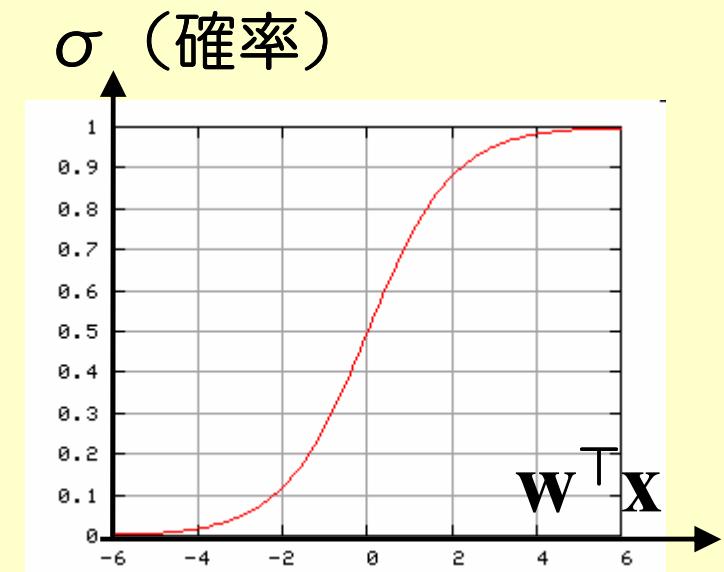
- \mathbf{w} の各次元は \mathbf{x} の各次元の $P(y = +1|\mathbf{x}; \mathbf{w})$ への寄与度

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

- σ はロジスティック関数

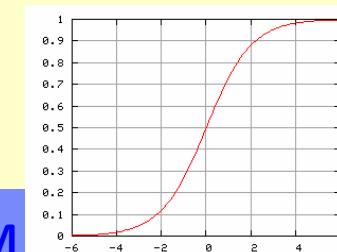
- 連續値を確率値 $[0, 1]$ にマップする

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



ここまでまとめ：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はどういった出力をすればよいかがわかつてくる
 - 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



機械学習の応用

- 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）

教師付き学習の応用例：信用リスク評価

- ある顧客に、融資を行ってよいか
 - 顧客 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)

$$\mathbf{x} := \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{pmatrix}$$

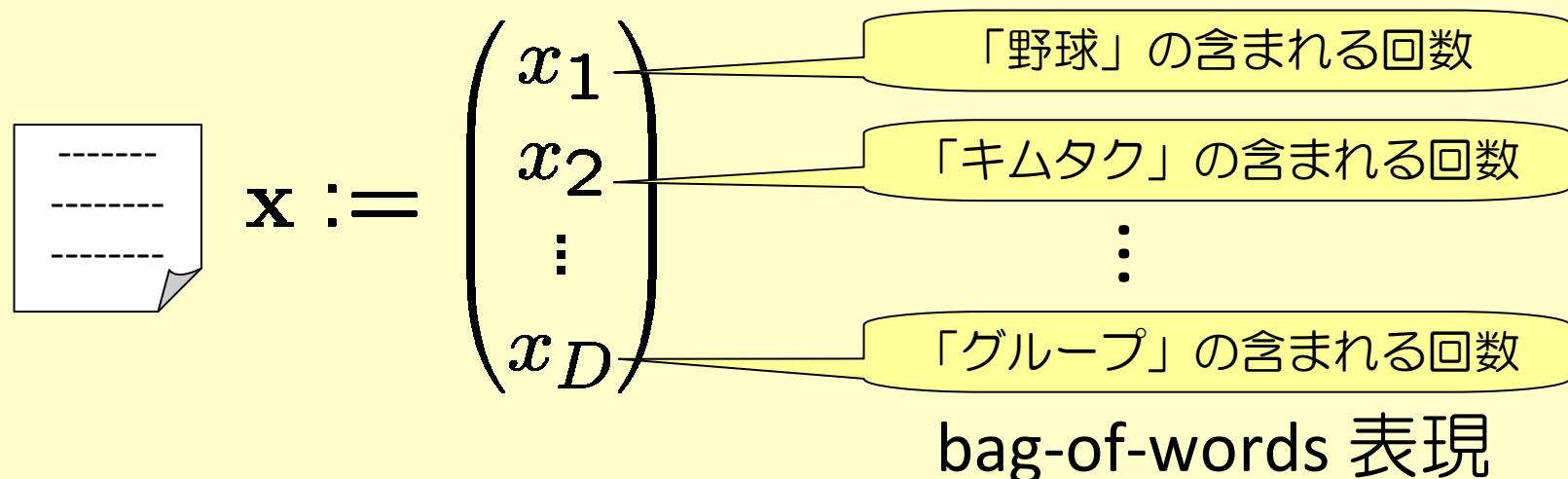
過去に延滞したことがあるか? (1/0)

リボ払い使用率

使用限度額

教師付き学習の応用例：テキスト分類

- 自然言語の文書が、あるカテゴリーに入るかどうか
 - 文書 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)

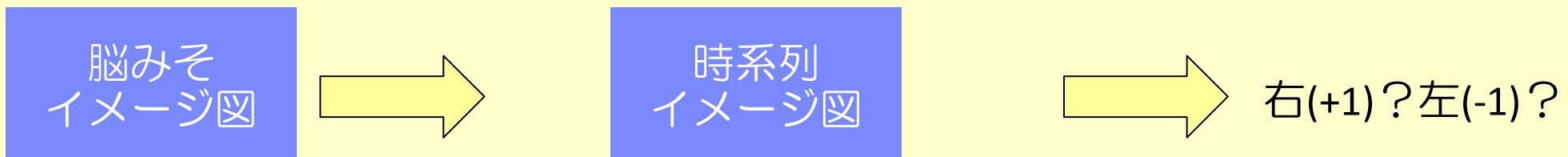


教師付き学習の応用例：画像認識

- 手書き文字認識



- BCI (Brain Computer Interface)

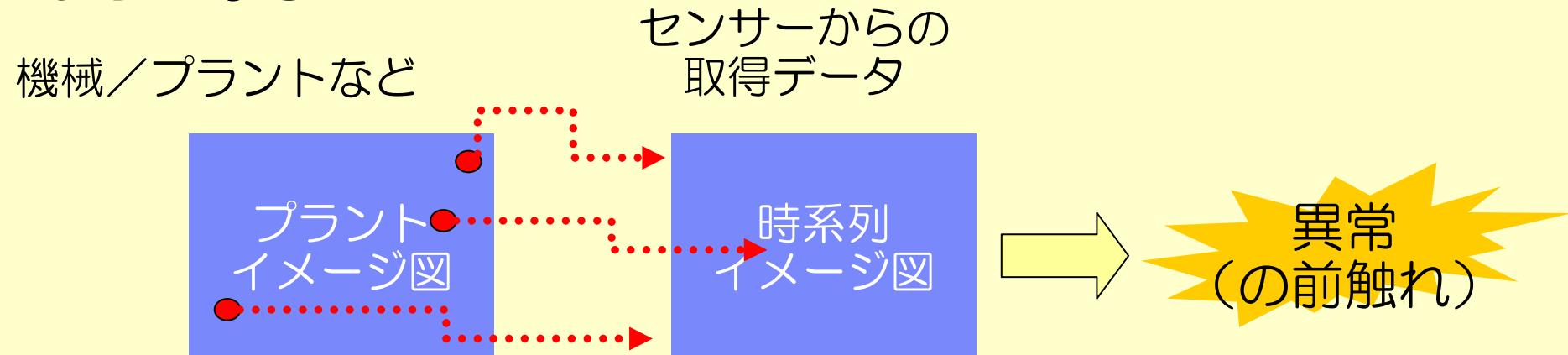


- ほか、顔画像認識や、動画認識



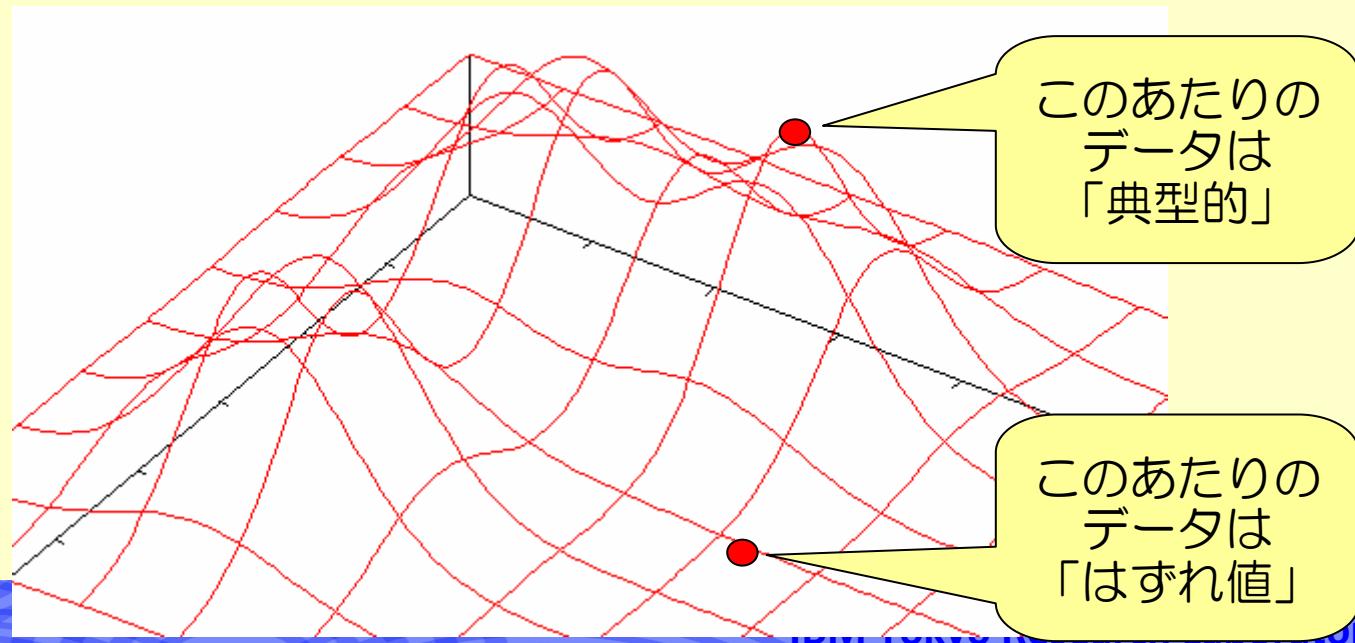
教師なし学習の応用例：異常検知

- 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



教師なし学習の応用例：異常検知

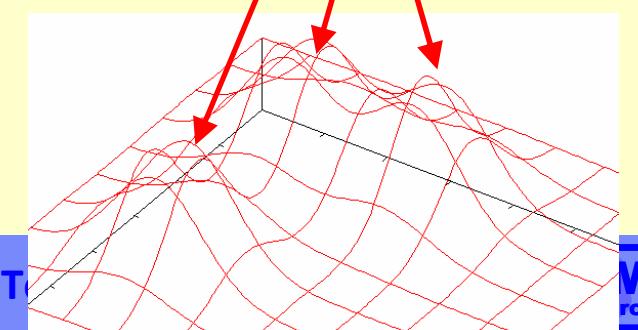
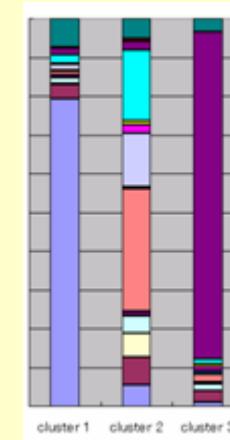
- システムの状態をベクトル x で表現し、教師無し学習による確率分布 $P(x)$ の推定を行う
 - コンピュータ間の通信量、各コマンドやメッセージ頻度
 - 各センサーの計測値の平均、分散、センサー同士の相関
- $P(x)$ の小さいデータ x は「めったに起こらない状態」＝システム異常、不正操作、計測機器故障などの可能性がある



教師なし学習の応用例：クラスタリング

- プロジェクトには様々な職種の人間が様々な配分でかかわる
 - 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

$$\mathbf{x} := \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{pmatrix} \quad \begin{array}{c} \text{PMの働いた時間} \\ \text{コンサルの働いた時間} \\ \vdots \\ \text{SEの働いた時間} \end{array}$$



ここまでまとめ：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
- 紹介しなかった応用
 - 推薦システム
 - ユーザーモデリング
 - 需要予測 (y が実数値)
- 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

学習の定式化

- 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、学習する主体を表現した
 - 対象：ものごとを実数値ベクトルで表現した
 - 主体：その上での確率モデルを考えた
- また、教師付き／教師無しの学習の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

最尤推定：モデル推定問題のもっとも一般的な定式化

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

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

- 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 $P(y = +1 | \mathbf{x}) := \sigma(\mathbf{w}^\top \mathbf{x})$

最尤推定：モデル推定問題のもっとも一般的な定式化

- 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と、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 $L := \sum_{i=1}^N \log P(\mathbf{x}^{(i)})$ 「対数尤度」とよぶ
 - 教師付き学習の場合 $L := \sum_{i=1}^N \log P(y^{(i)}|\mathbf{x}^{(i)})$
- 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

最尤推定の例：多次元正規分布

- 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）

$$P(\mathbf{x}; \boldsymbol{\mu}) := \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp(-(\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}))$$

- 対数尤度は

$$\begin{aligned} L &:= \sum_{i=1}^N \log P(\mathbf{x}^{(i)}; \boldsymbol{\mu}) \\ &= \sum_{i=1}^N (-(\mathbf{x}^{(i)} - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu})) + \text{const.} \end{aligned}$$

- $\boldsymbol{\mu}$ で微分 $\frac{\partial L}{\partial \boldsymbol{\mu}} = \sum_{i=1}^N 2\Sigma^{-1}(\mathbf{x}^{(i)} - \boldsymbol{\mu}) = 2\Sigma^{-1} \sum_{i=1}^N (\mathbf{x}^{(i)} - \boldsymbol{\mu})$

- $\boldsymbol{\mu} := \mathbf{0}$ とおいて解くと、

$$\boldsymbol{\mu} = \frac{\sum_{i=1}^N \mathbf{x}^{(i)}}{N}$$

データの平均に
なった

ここまでまとめ：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る

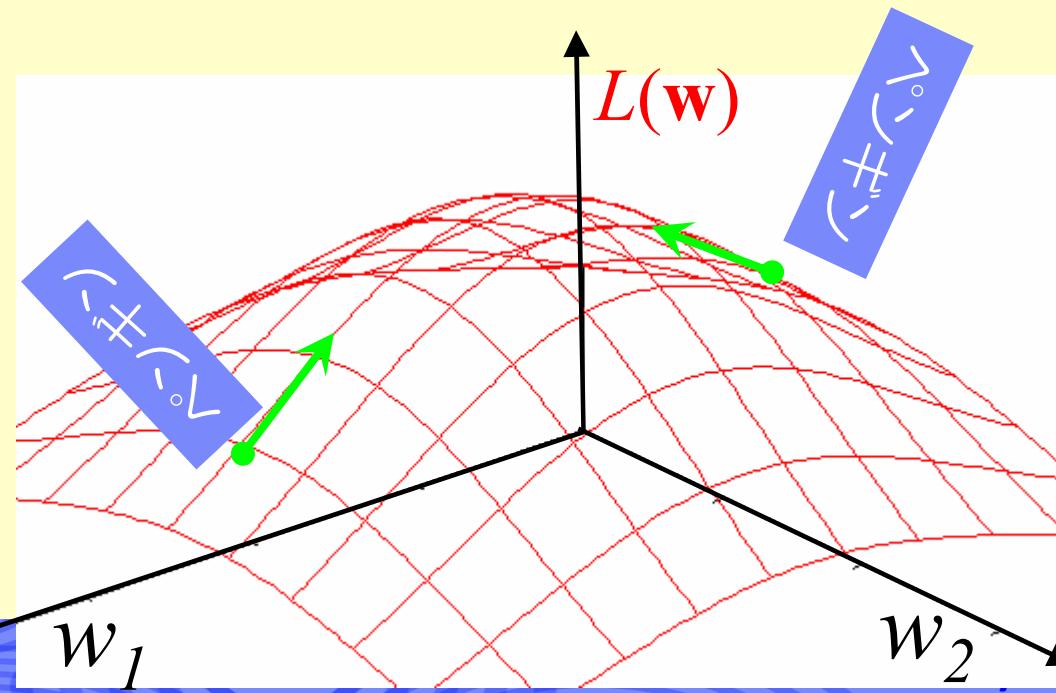
- 最尤推定
「訓練データを、最もよく再現するパラメータが良いパラメータとする」に基づいて学習を行う
- 対数尤度を目的関数として、パラメータについての最大化を行う
- 多次元正規分布の平均パラメータの最尤推定による推定値は、データの平均によってもとまる
 - ちなみに、共分散行列の推定値は、データの共分散行列によつて求まる
- もっと複雑なモデル（混合正規分布、ロジスティック回帰）では、最尤推定はどのように行えばいいだろうか？

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求める

- 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
 - EMアルゴリズム
- 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

勾配法：もっとも基本的な最適化法

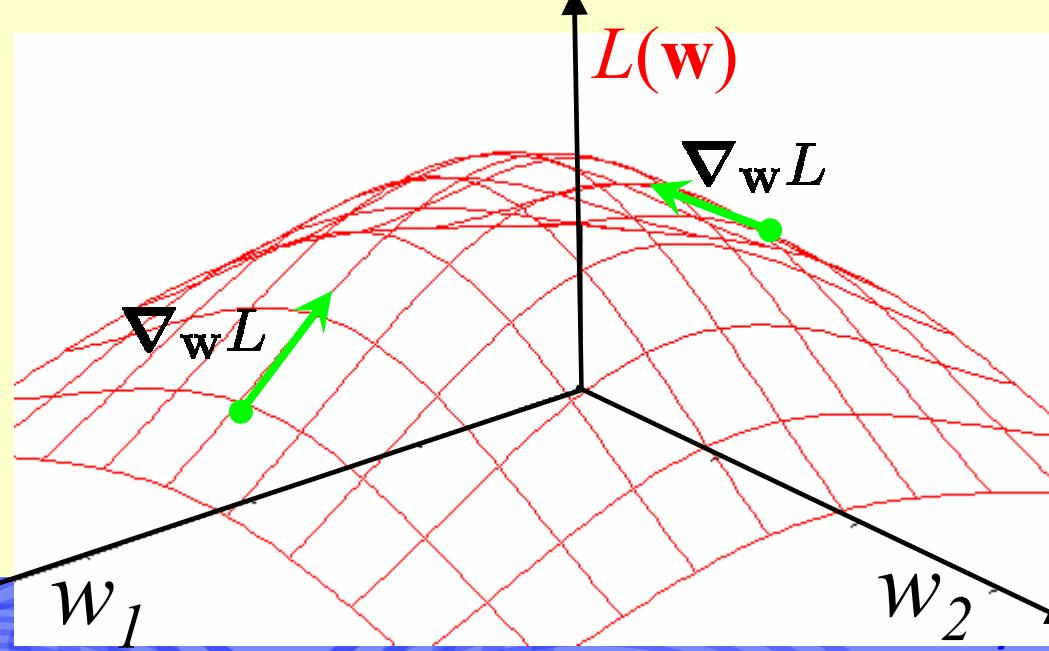
- 山（対数尤度 L ）の頂点を（なるべく速く）目指したい
- もっとも坂が急な方向に向かって（パラメータ上で）1m進む
 - 頂上付近だと、頂上を越えて向こう側にいってしまうことも
 - 実際には歩幅はだんだん小さくする

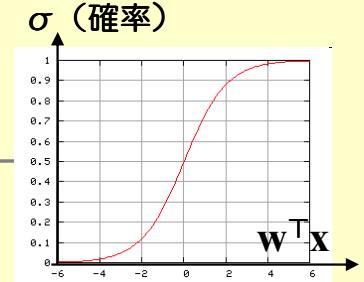


勾配法：もっとも基本的な最適化法

- もっとも急な方向 = 勾配
- 勾配は、目的関数（対数尤度） $L(\mathbf{w})$ のパラメータ \mathbf{w} での偏微分
$$\nabla_{\mathbf{w}} L := \left(\frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, \dots, \frac{\partial L}{\partial w_D} \right)$$
- 勾配の方向に、少し（正の定数 α ）更新する

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





ロジスティック回帰に対する勾配法

- 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クス集合をPos,
カテゴリ-1 のインデクス集合をNegとすると、対数尤度は、

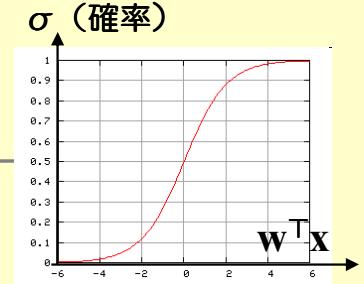
$$\begin{aligned}
 L(\mathbf{w}) &:= \sum_{i=1}^N \log P(y^{(i)} | \mathbf{x}^{(i)}; \mathbf{w}) \\
 &= \sum_{i \in \text{Pos}} \log \sigma(\mathbf{w}^\top \mathbf{x}^{(i)}) + \sum_{i \in \text{Neg}} \log(1 - \sigma(\mathbf{w}^\top \mathbf{x}^{(i)}))
 \end{aligned}$$

- 勾配は、

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

+1カテゴリの
データ

-1カテゴリの
データ



ロジスティック回帰に対する勾配法

- 勾配は、

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

- 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、

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

EMアルゴリズム：混合分布の効率的な推定法

- 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
 - 「隠れ変数」をもつ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
 1. 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
 2. パラメータを固定したときの隠れ変数の推定

混合正規分布のためのEMアルゴリズム

- 混合正規分布 ($\Sigma^{(k)}$ は固定)

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

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

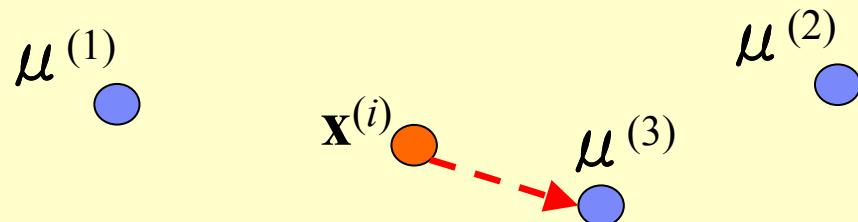
において、対数尤度↓を最大化するパラメータを求めたい

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

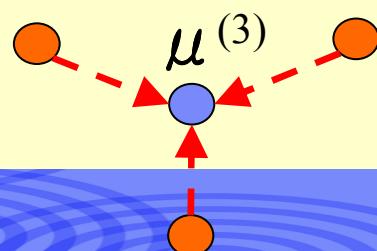
- 煩雑になるので、単純化して、
インチキEMアルゴリズム（K-meansアルゴリズム）を導くこと
にする

K-meansアルゴリズム

- 混合正規分布は、正規分布を一つ選んで、それを使って x を生成していると解釈できる
- 各データ $x^{(i)}$ 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 $\mu = \sum_i x^{(i)} / N$
- 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 $x^{(i)}$ 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



学習アルゴリズムのオンライン化

- 勾配法、EM法、ともに各繰り返しは、(データ数 N) × (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
 - 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1つづつ処理する
 - 人間の学習のイメージにちかい（「だんだん」「試行錯誤」）

ロジスティック回帰の勾配法のオンライン化

- 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

$$\begin{aligned} L^{(i)}(\mathbf{w}) &:= \log P(y^{(i)} | \mathbf{x}^{(i)}; \mathbf{w}) \\ &= \begin{cases} \log \sigma(\mathbf{w}^\top \mathbf{x}^{(i)}) & \text{if } y^{(i)} = +1 \\ \log(1 - \sigma(\mathbf{w}^\top \mathbf{x}^{(i)})) & \text{if } y^{(i)} = -1 \end{cases} \end{aligned}$$

- 勾配の方向にパラメータを少し更新

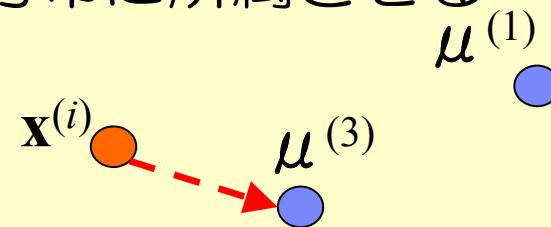
$$\begin{aligned} \mathbf{w}^{\text{NEW}} &\leftarrow \mathbf{w}^{\text{OLD}} + \alpha \nabla_{\mathbf{w}} L^{(i)}(\mathbf{w}) \\ &= \mathbf{w}^{\text{OLD}} + \alpha \begin{cases} \left(1 - \sigma(\mathbf{w}^\top \mathbf{x}^{(i)})\right) \mathbf{x}^{(i)} & \text{if } y^{(i)} = +1 \\ -\sigma(\mathbf{w}^\top \mathbf{x}^{(i)}) \mathbf{x}^{(i)} & \text{if } y^{(i)} = -1 \end{cases} \end{aligned}$$

- 1ステップの計算量は $O(D)$

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：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（おかしなデータを発見）を同時に行う
- 以下のステップを繰り返す

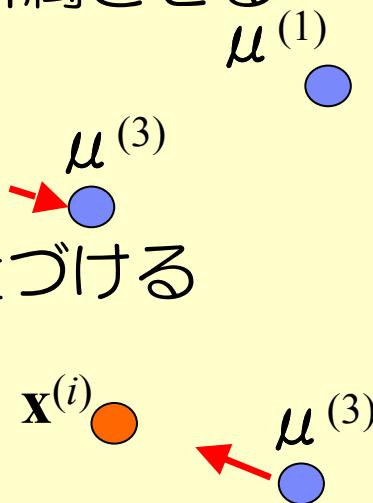
1. $\mathbf{x}^{(i)}$ を、最寄の平均をもつ正規分布に所属させる
(バッチ版と同じ) $\mu^{(1)}$ $\mu^{(1)}$



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

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

– ϵ 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

ここまでまとめ：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？

評価方法：何をもって学習の良し悪しを計るか？

- 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 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 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

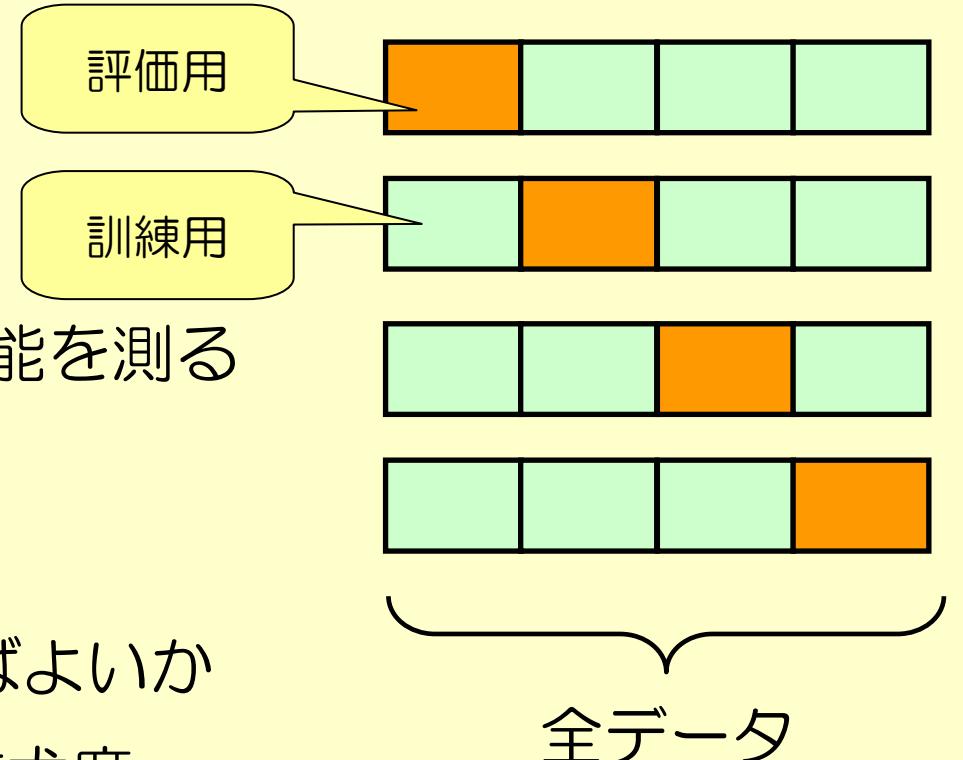
訓練用

評価用

交差検定（クロスバリデーション）

- 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

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

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

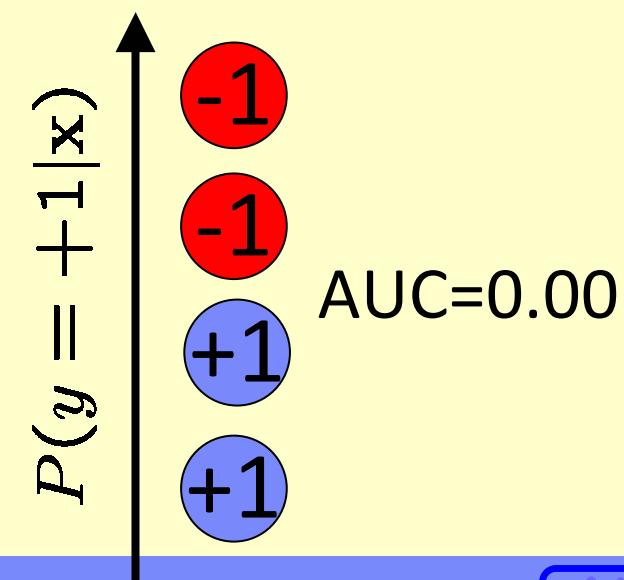
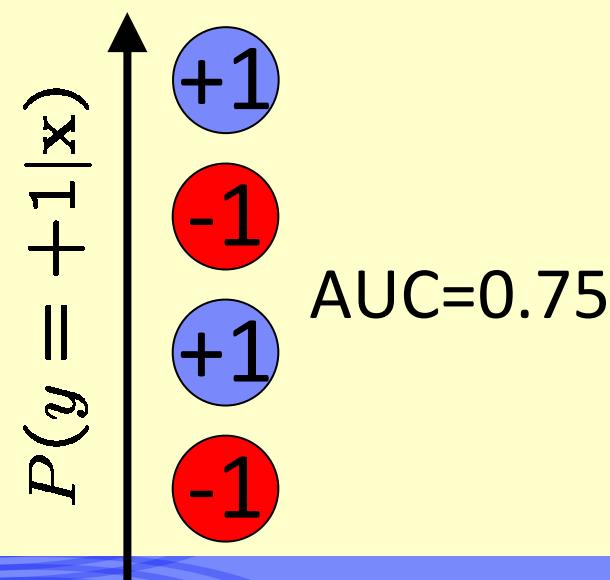
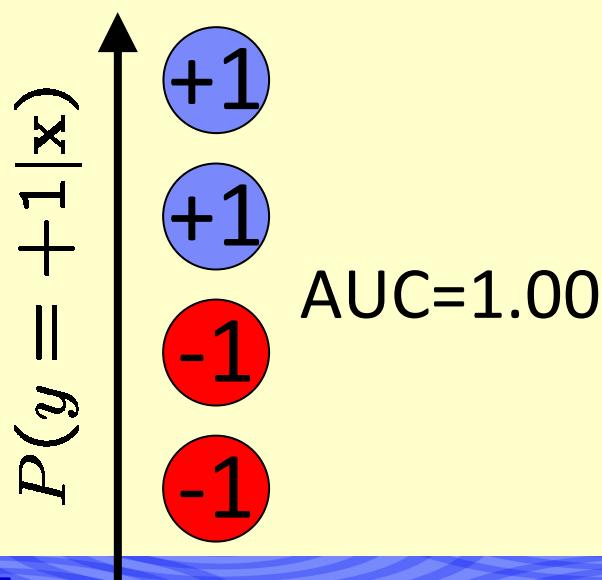
- テストデータと訓練データが同じ分布から出ているのであれば、訓練データに対して高い確率を与えるモデルは、テストデータに対しても高い確率を与えるはず
- もしくは、クラスタリングなどの場合に、正しいクラスラベルが分かっていたりすれば、それとの一致具合も使われる

教師付き学習の場合、正解率のほうがよいけれど…

- 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 $P(y = +1|x)$ の出力を用いて予測したカテゴリが正しいかどうかに興味がある
 - $P(y = +1|x) \geq 0.5$ であれば、カテゴリ +1 と予測
 - $P(y = +1|x) < 0.5$ 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
 - $P(y = +1|x)$ が 0.5 よりも全体的に高め／低めに出る
 - 評価のベースラインがわからない

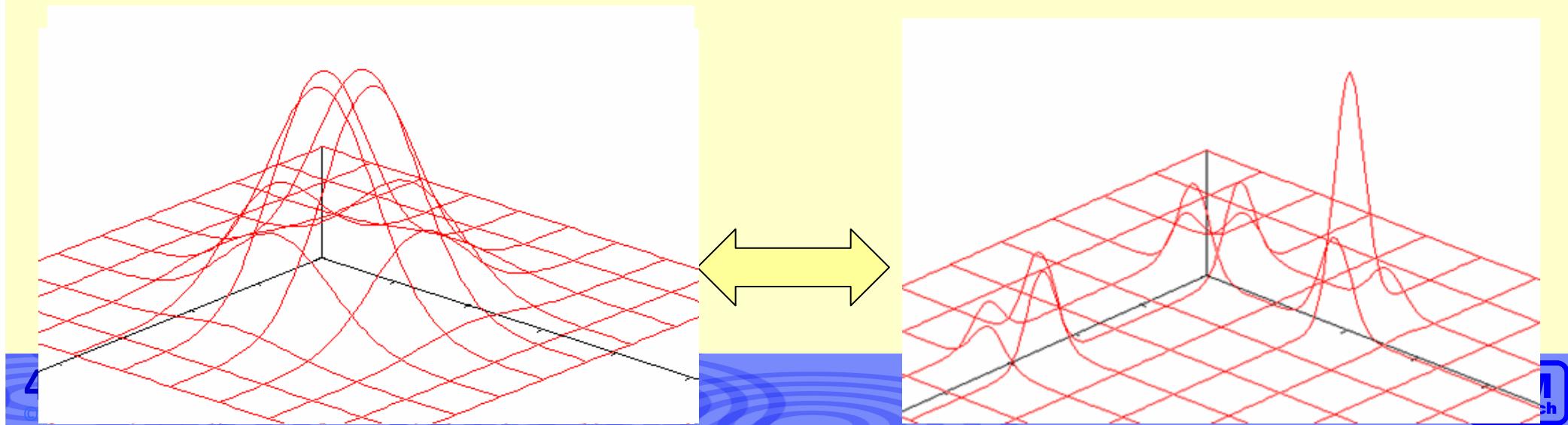
AUC: 閾値に依存しない教師付き学習の定番評価値

- ある閾値を決めたときの正解率ではなく、 $P(y = +1|x)$ 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



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
- データの数に対して、モデルの自由度（パラメータの数）が高すぎると、これに近い現象が起こってしまう
 - とくに x が高次元のデータで起こる



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 \rightarrow 極端なモデル
 - ノルムが小 \rightarrow 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

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

パラメータ 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

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

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

$$\| \mathbf{w} \| := \| \mathbf{w} \|_2^2 = w_1^2 + w_2^2 + \cdots + w_D^2$$

- 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

$$\| \mathbf{w} \| := |\mathbf{w}|_1 := |w_1| + |w_2| + \cdots + |w_D|$$

- これを対数尤度にペナルティ項として加えると

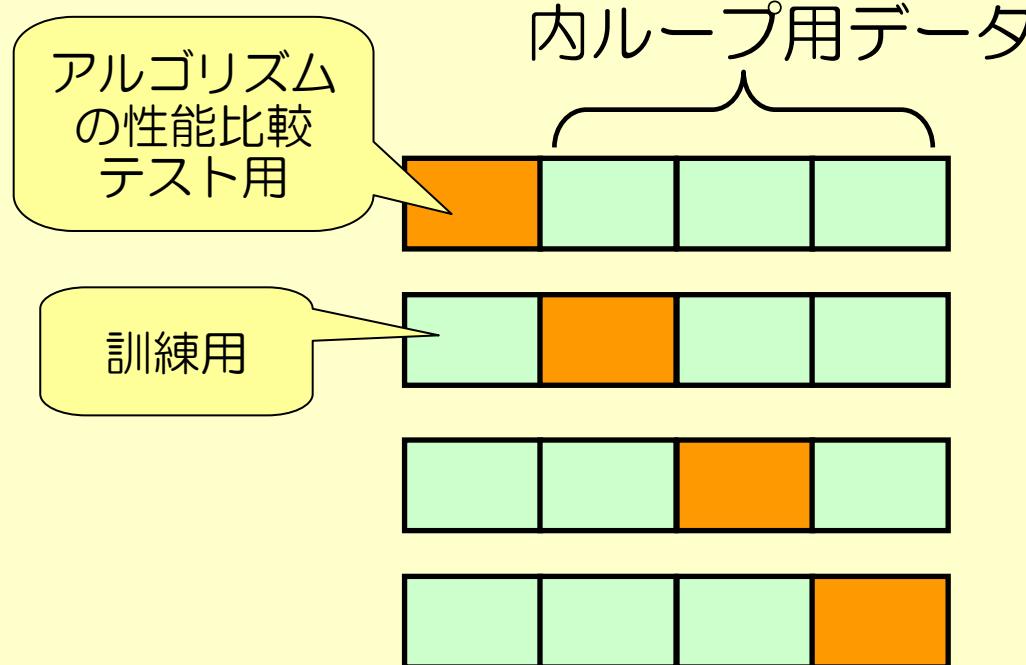
$$L := \sum_{i=1}^N \log P(y^{(i)} | \mathbf{x}^{(i)}; \mathbf{w}) - \lambda |\mathbf{w}|_1$$

- 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

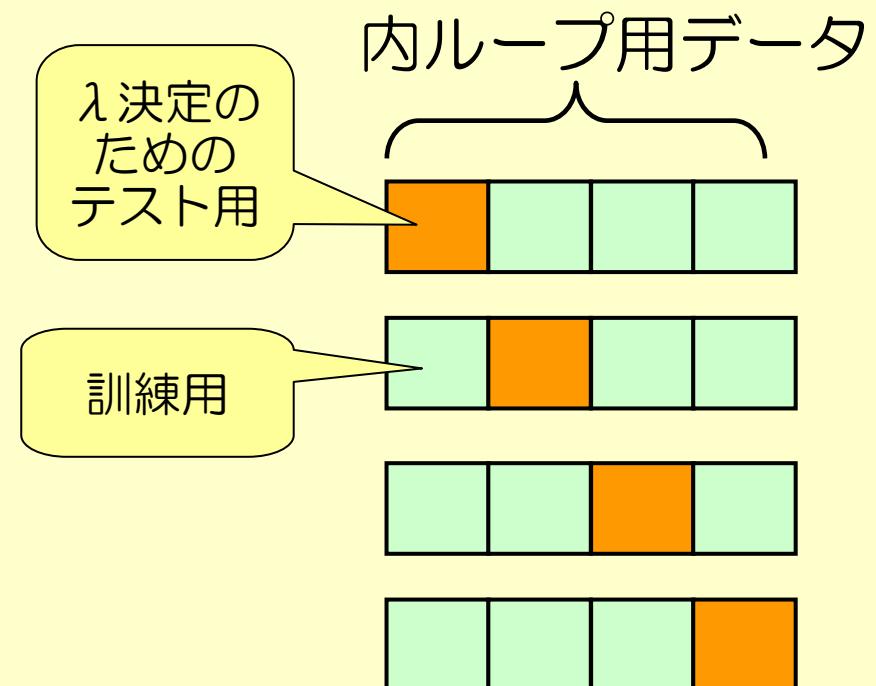
ハイパーパラメータ λ の決定

- 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回の評価を行う交差検定（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

拡張

- 多クラス分類
- カーネル法

多クラスの分類（教師付き学習）

- これまで {+1,-1} の2クラスの分類問題を考えていた
- Kクラスへの対応 {A, B, C, D, E}
 - 多クラスのロジスティック回帰モデルを考える
 - 2クラスに帰着する
 - あるクラスか、そうでないか、という分類問題をK個考える
 - 予測時にもっとも確率の高いクラスに予測する

カーネル法

- ロジスティック回帰のカーネル化
- リプレゼンタ定理

カーネル法とは

- 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
- なぜ?
 - サポートベクトルマシン (SVM) というモデルが各地で大成功を収めた
 - 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
 - 木やグラフなどの複雑な対象を扱うことが出来る（後半）
- 多くのモデルが、カーネル法に変換することが出来る

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

$$P(y = +1|\mathbf{x}) := \sigma(\mathbf{w}^\top \mathbf{x})$$

- 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

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

これを
「カーネル化」
という

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

- ロジスティック回帰モデルを書き直すと

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

- $\langle \cdot, \cdot \rangle$ 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)

カーネル化によって何が起こったか？

内積

- カーネルロジスティック回帰モデル

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

ただし、カーネル関数 $K(\mathbf{x}^{(i)}, \mathbf{x}) := \langle \mathbf{x}^{(i)}, \mathbf{x} \rangle$

- 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 \mathbf{x} の次元よりも小さいオーダーであるときに速い
 - \mathbf{x} が何だかよく分からぬ対象であっても、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

リプレゼンタ定理：なぜ線形結合で表してよいのか？

- 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

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

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

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

- L1正則化（ラッソ正則化）ではこれは保証されない

リプレゼンタ定理の証明

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

- 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 $\mathbf{w}^* = \mathbf{w} + \mathbf{w}'$

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

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

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 ($= \mathbf{0}$) してよい。よって、 $\mathbf{w}^* = \mathbf{w}$

ここまでまとめ：

- 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とくことができる
- 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えるようにする方法
 - 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リプレゼンタ定理
 - パラメータについて線形なモデルに、L2（リッジ）正則化を適用する場合、成立する

講義の流れ（前半）：機械学習入門

- 機械学習とは
- 学習問題の区分
 - 教師つき学習と教師無し学習
 - 教師付き学習
 - ロジスティック回帰モデル
 - 教師無し学習
 - 混合分布モデル
- 機械学習の応用
 - 信用リスク推定
 - テキスト分類
 - 画像認識
 - 異常検知
 - クラスタリング
- 学習の定式化
 - 最尤推定
 - 多次元正規分布の最尤推定
- アルゴリズム
 - 勾配法
 - ロジスティック回帰モデルの勾配法
 - EMアルゴリズム
 - 混合正規分布のEM的アルゴリズム
 - 大規模データへの対応：オンラインアルゴリズム
- 機械学習手法の評価方法
 - 訓練データとテストデータ
 - 交差検定（クロスバリデーション）
 - 性能指標
 - テスト尤度、正解率、AUC
 - 過学習
 - 正則化
 - L1正則化とL2正則化
- 拡張
 - 多クラス分類
 - カーネル法
 - ロジスティック回帰のカーネル化
 - リプレゼンタ定理

参考書：

- C.M.ビショップ著 「パターン認識と機械学習 - ベイズ理論による統計的予測【上】 【下】」（シュプリンガー・ジャパン）
 - 現代的な統計的機械学習を網羅した教科書
 - 鹿島も訳者として参加
 - 原著「Pattern Recognition and Machine Learning」
by C.M. Bishop
- 赤穂昭太郎著「カーネル多変量解析」（共立出版）
 - カーネル法に特化した教科書



本講義の目的：機械学習（とIBM）の宣伝

- 機械学習が企業のビジネスにおいて差別化要因になることを知ってもらう
 - 「知っておくと得する（≒つぶしがきく）」という気持ちを伝える
- ちょっとやれば機械学習手法を使えそうだという気になってもらう
 - 機械学習の問題設定を知ってもらう
 - 機械学習のモデルを知ってもらう
 - 機械学習のアルゴリズムを知ってもらう
- 機械学習の先端の研究を垣間見てもらう
 - 構造データの分析（手前味噌）

（分かりやすさを優先するため、厳密さは犠牲にして、単純化して話をする）

あと、

- IBM基礎研究所を宣伝する