

総合科目 一般F「数理工学のすすめ」  
機械学習入門

かしま ひさし  
鹿島 久嗣

情報理工学系研究科  
数理情報学専攻

2011/5/11

kashima@mist.i.~



---

本日の資料は以下のURLからダウンロードできます

<http://goo.gl/d9yxG>

この講義シリーズのWebページは

<http://goo.gl/ZcLlb>

## レポート課題（4/27+5/11分）

---

次の2つの課題のうちいずれか：

1. 機械学習／データマイニング関連の図書で興味をもった部分を調べてまとめよ

参考図書の例)

- 山西健司：「データマイニングによる異常検知」共立出版（2009）
- 山西健司：「情報論的学習理論」共立出版（2010）
- C.M.ビショップ：「パターン認識と機械学習」（上）（下）シュプリンガー・ジャパン（2008－2009）

2. 機械学習／データマイニングの現実的な応用例や応用可能性について論文または書物を調べてまとめよ

- ただし、引用あるいは参考とした資料は明記すること

## 今日は機械学習の初歩的な手法を具体的に紹介します

---

- 機械学習：教師つき学習と教師なし学習
- 教師つき学習の例：パーセプトロン
- 教師なし学習の例： $K$ -平均クラスタリング

---

## 機械学習：教師つき学習と教師なし学習

## 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？

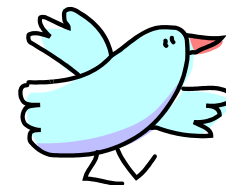
## 部分文字列に注目してみると...

- 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

「あり」のグループには  
鳥の名前が含まれている



## 仲間はずれさがし：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
たらばがに
毛がに
えび



## グループ分けしてみると...

- 「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	やわらかい	くも たこ	いか
	かたい	たらばがに やどかり	毛がに えび

グループ1 (くも, たこ)

グループ2 (やわらかい, かたい)

グループ3 (毛がに, えび)

- 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

## 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」と「教師なし学習」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
- 仲間はずれ探しの場合：
  - 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  
(この場合、グループに入らないメンバーをみつける)
- これらはそれぞれ機械学習の2大タスク「教師つき学習」と「教師なし学習」に対応している

## 教師つき学習は、入出力関係の推定問題です

---

- 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」か「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、ヒントなしではこれではできない...  
そこでヒントとして過去の事例（＝訓練データ）を利用できる
  - 「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

## 教師なし学習は、入力データのグループ分けを行います

---

- 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、グループ分けくらいしかできない
  - 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無い)
      - 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

# 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「人間のもつ”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）

# 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

- 機械学習とは、本来  
「人間のもつ”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
- 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、どういう入力のお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

---

## 機械学習問題の定式化

## 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

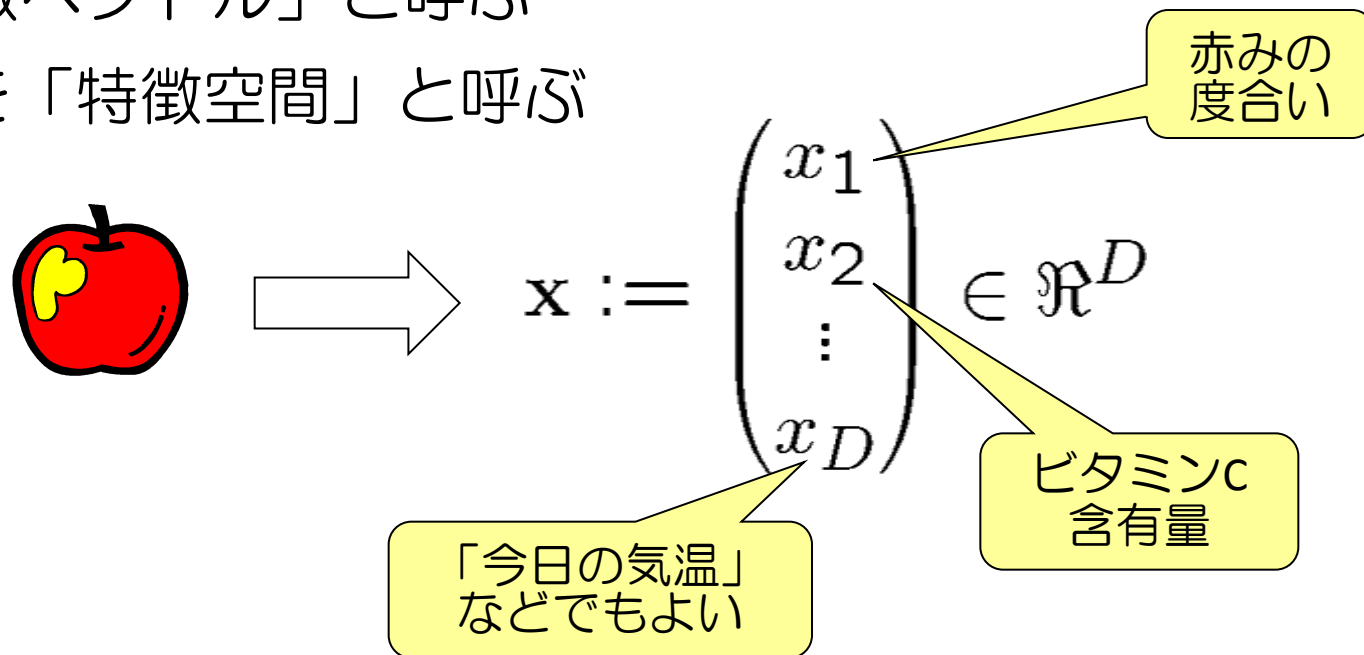
- 学習機能を計算機上に実現するために、まず、学習問題を数理的にとらえる必要がある
- まずは、入力をどう数理的（＝計算機可読な形式）に表現するか？
  - 「やどかり」「ねこ」「りんご」は計算機上でどのように扱うか？
- 出力については比較的自明
  - 「あり」を+1、「なし」を-1と割り当てる



## 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  $x$  として表現する
  - $x$  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  $x$  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

# 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

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

- $\mathbf{x}^{(i)}$ :  $i$  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、入出力の関係を学習する

# 教師なし学習では、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

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

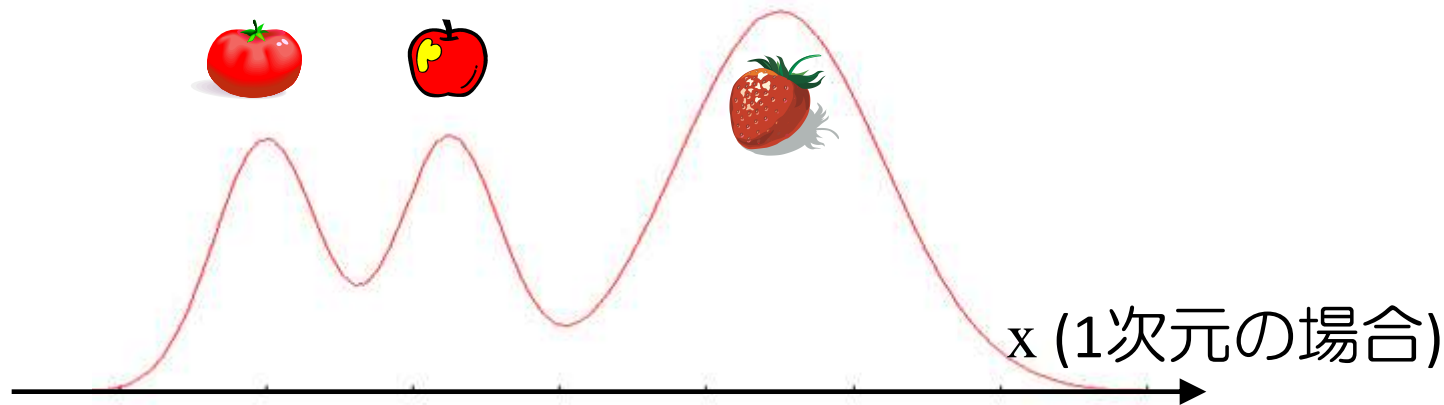
$$\mathbf{x}^{(i)} := \begin{pmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_D^{(i)} \end{pmatrix} \in \mathfrak{R}^D$$

1つめの  
データ

2つめの  
データ

...

- これらをいくつかのグループに分ける
- 教師なし学習は、（大げさにいえば）明示的に指定されることなしに、“概念”を形成するプロセスを表している



---

## 教師つき学習法：パーセプトロン

## 線形モデル：もっともシンプルな出力予測モデル

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

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

- $\text{sign}()$  は引数が0以上なら+1、0未満なら-1を返す関数
- $\mathbf{w} = (w_1, w_2, \dots, w_D)^\top$  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

# 学習とは、訓練データからパラメータベクトル $w$ を決定することです

---

- パラメータ $w$  がきまるとモデル $f$  がきまる

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

- 訓練データから $w$ を決定するのが「学習」

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

- 基本的には、訓練データの入出力を再現できるように $w$ を調整する
  - 出力が $y = +1$ のデータについては $\mathbf{w}^\top \mathbf{x} > 0$ となるように
  - 出力が $y = -1$ のデータについては $\mathbf{w}^\top \mathbf{x} < 0$ となるように
  - まとめてかくと $y \mathbf{w}^\top \mathbf{x} > 0$

# パーセプトロン：シンプルな逐次学習型アルゴリズム

---

- パーセプトロンとは：
  - 制約  $y^{(i)} w^\top x^{(i)} > 0$  (for all  $i$ )を実現するアルゴリズム
  - 訓練データを1つずつ処理する逐次学習型のアルゴリズム
- 学習の各ステップにおいて、1つの訓練データ( $x^{(i)}, y^{(i)}$ )を選び、これを用いてモデルパラメータ  $w$  の更新を行う
- ( $x^{(i)}, y^{(i)}$ ) について既に制約が満たされている場合には何も行わない
- 制約が満たされない場合に限って以下の更新式を用いてパラメータの更新を行う

$$w^{\text{新}} \leftarrow w^{\text{旧}} + y^{(i)} x^{(i)}$$

- パラメータの更新は制約を満たす方向に行われる（後述）

## パーセプトロンのパラメータ更新則は制約を満たそうとする方向に行われます

---

- 更新後のパラメータを用いて、  
入力  $\mathbf{x}^{(i)}$  に対する予測  $\mathbf{w}^\top \mathbf{x}^{(i)}$  を計算してみると：

$$\mathbf{w}^{\text{新}} \mathbf{x}^{(i)} = \mathbf{w}^{\text{旧}} \mathbf{x}^{(i)} + y^{(i)} \mathbf{x}^{(i)\top} \mathbf{x}^{(i)}$$

- 正しいクラスが  $y^{(i)} = +1$  の場合には  $\mathbf{w}^{\text{新}} \mathbf{x}^{(i)}$  が  $\mathbf{w}^{\text{旧}} \mathbf{x}^{(i)}$  よりも  $\mathbf{x}^{(i)\top} \mathbf{x}^{(i)}$  (必ず0以上の値をとる) だけ大きくなる
  - $\mathbf{w}^{\text{新}} \mathbf{x}^{(i)}$  が0より大きくなる方向にパラメータを調整
- 一方、正しいクラスが  $y^{(i)} = -1$  の場合には同じ量だけ小さくなる
  - $\mathbf{w}^{\text{新}} \mathbf{x}^{(i)}$  が0より小さくなる方向にパラメータを調整
- つまり、制約を満たそうとする方向にパラメータが更新されている



## パーセプトロンは、制約を満たすパラメータを有限回のパラメータ更新で見つけることができます（もしあれば）

- パーセプトロンアルゴリズムのパラメータ更新回数についての定理
- 2つの条件が満たされていると仮定する：
  - [条件1] 全てのデータに対し制約  $y^{(i)} \mathbf{w}^{*\top} \mathbf{x}^{(i)} > 0$  を満たす、ある理想的なパラメータ  $\mathbf{w}^*$  が存在して：
    - ある正の定数  $\gamma > 0$  について  $y^{(i)} \mathbf{w}^{*\top} \mathbf{x}^{(i)} > \gamma$  が成立する
    - $\mathbf{w}^*$  の2-ノルムは  $\|\mathbf{w}^*\|_2^2 = 1$  である
  - [条件2] 全ての訓練データについて特徴ベクトル  $\mathbf{x}^{(i)}$  のノルムが  $\|\mathbf{x}^{(i)}\|_2^2 < R^2$  である
- このとき、パーセプトロンアルゴリズムが全ての制約を満たすパラメータを発見するまでのパラメータ更新回数  $k$  は高々： $\left(\frac{R}{\gamma}\right)^2$   
(予測の符号を間違える回数)

※ 過去スライド使いまわしの都合上  $w^{\text{新}}$  と  $w^{\text{NEW}}$  が混ざっていますがご容赦ください (同じものです)

## 証明 (前半)

- 今  $k$  回目のパラメータ更新 ( $k$  回目の制約不成立) が起こったとする
- 更新後のパラメータ  $w^{\text{新}}$  と、理想的なパラメータ  $w^*$  との近さを見るために、その内積  $w^{\text{新}\top} w^*$  を調べると:

$$\begin{aligned} w^{\text{新}\top} w^* &= w^{\text{旧}\top} w^* + y^{(i)} x^{(i)\top} w^* && : \text{更新式の定義より} \\ &> w^{\text{旧}\top} w^* + \gamma && : \text{仮定 } y^{(i)} w^{*\top} x^{(i)} > \gamma \text{ より} \\ &> k\gamma && : \text{パラメータ更新回数が } k \text{ 回であることより} \end{aligned}$$

- パラメータ更新のたびに内積が少なくとも  $\gamma$  ずつ大きくなっていく

- 両辺を  $\|w^{\text{新}}\|_2$  で割ると: 
$$\left( \frac{w^{\text{NEW}}}{\|w^{\text{NEW}}\|_2} \right)^\top w^* > \frac{k\gamma}{\|w^{\text{NEW}}\|_2}$$

– 左辺にある2つのベクトルの2-ノルムは共に1

– その内積は1以下であるから 
$$\frac{k\gamma}{\|w^{\text{NEW}}\|_2} \leq 1$$

## 証明 (後半)

- あとは  $\|\mathbf{w}^{\text{新}}\|_2$  を評価すればよい。
- 更新式  $\mathbf{w}^{\text{新}} \leftarrow \mathbf{w}^{\text{旧}} + y^{(i)} \mathbf{x}^{(i)}$  より：

$$\begin{aligned}\|\mathbf{w}^{\text{新}}\|_2^2 &= \|\mathbf{w}^{\text{旧}}\|_2^2 + \|\mathbf{x}^{(i)}\|_2^2 + 2 y^{(i)} \mathbf{w}^{\text{旧}\top} \mathbf{x}^{(i)} \\ &\leq \|\mathbf{w}^{\text{旧}}\|_2^2 + R^2 + 2 y^{(i)} \mathbf{w}^{\text{旧}\top} \mathbf{x}^{(i)} && \text{: 条件2 } \|\mathbf{x}^{(i)}\|_2^2 < R^2 \\ &\leq \|\mathbf{w}^{\text{旧}}\|_2^2 + R^2 && \text{: 制約の不成立の仮定より}\end{aligned}$$

- パラメータの更新回数が  $k$  回であることから、この不等式を繰り返し適用することにより  $\|\mathbf{w}^{\text{新}}\|_2^2 \leq kR^2$  すなわち  $\|\mathbf{w}^{\text{新}}\|_2 \leq \sqrt{k} R$
- これを、前頁で導いた不等式と組み合わせ、 $\gamma$  について解くと：

$$\frac{k\gamma}{\|\mathbf{w}^{\text{NEW}}\|_2} \leq 1 \qquad k \leq \left(\frac{R}{\gamma}\right)^2$$

## データを完璧に分類できる超平面が存在しない場合でも、パーセプトロンの性能はよいことが知られています

- 定理では、制約  $y^{(i)} \mathbf{w}^{*\top} \mathbf{x}^{(i)} > 0$  が成立すること、すなわち、全ての訓練データを間違いなく判別することのできる超平面  $\mathbf{w}^{*\top} \mathbf{x}^{(i)} > 0$  が存在することを仮定していた
- 現実的には真実の  $f$  は線形モデルよりももっと複雑であったり、あるいはデータにノイズが含まれていたりなどの事情により、この制約は必ず満たされるとは限らない
- このような場合にも、「最も制約を破らないようなパラメータ  $\mathbf{w}^*$ 」と比較して、パーセプトロンアルゴリズムが誤った予測をする回数が小さく抑えられることがわかっている

$$k \leq \left( \frac{R+D}{\gamma} \right)^2$$

- ここで  $D$  は  $\mathbf{w}^*$  の制約破りっぷりを表す

---

## 教師つき学習の応用例

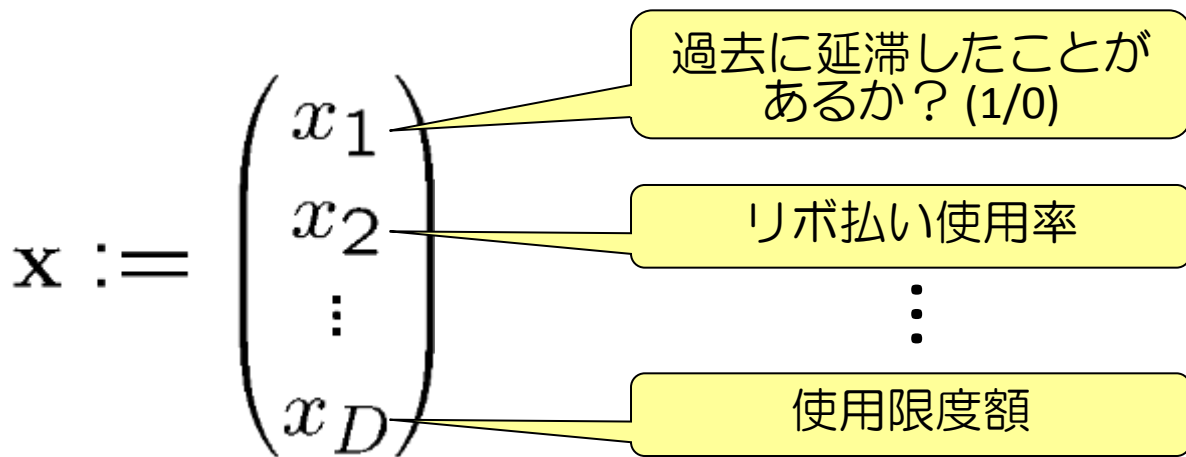
## 教師つき学習の応用例

---

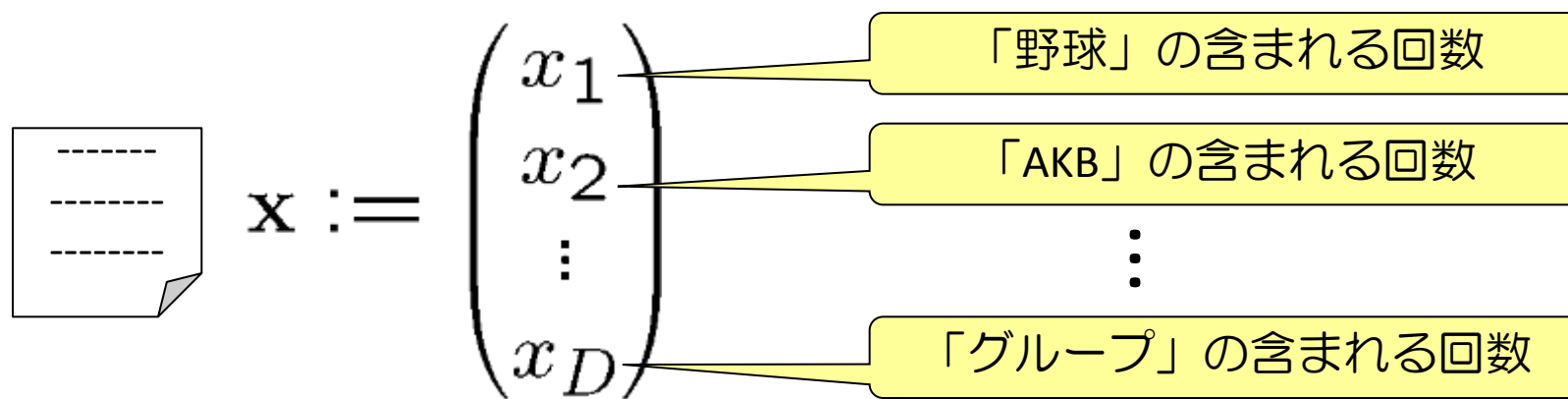
- 信用リスク評価
- テキスト分類
- 画像認識

「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  $x$  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)



- 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)



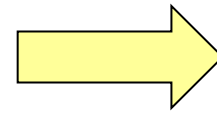
bag-of-words 表現



「これ、何て書いてあるの？」 「いま何考えてる？」

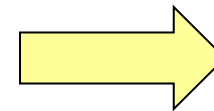
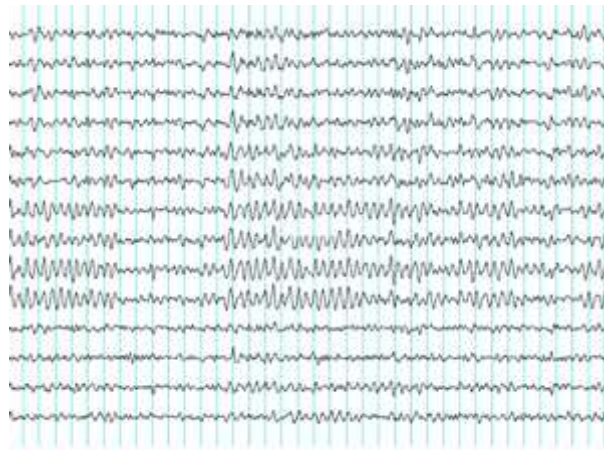
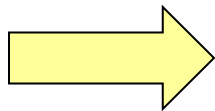
## ■ 手書き文字認識

7210414959  
0690159734  
9665407401  
3134727121  
1742351244



{ ある文字か(+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

---

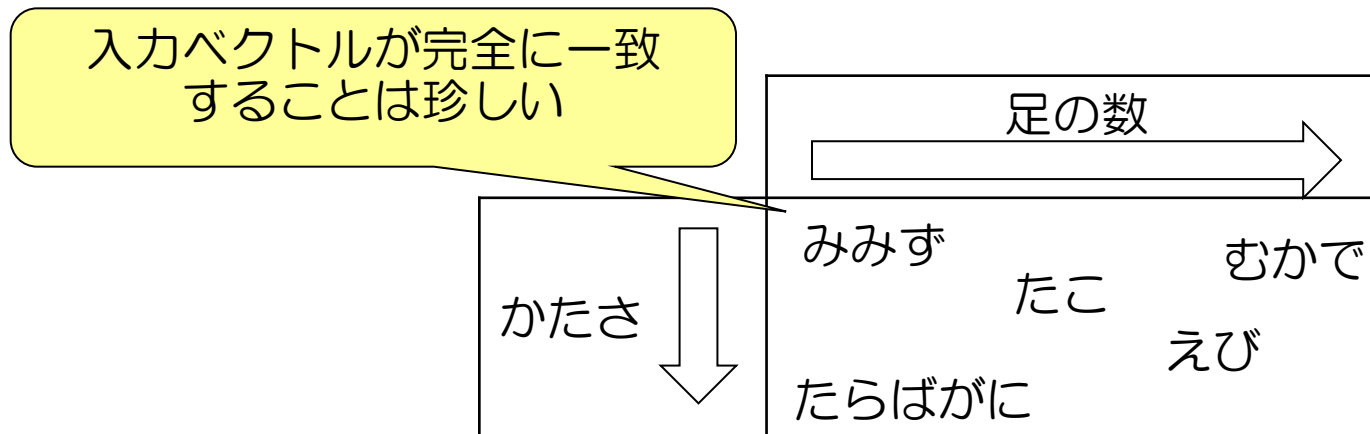
## 教師なし学習法： $k$ -平均クラスタリング

教師なし学習では入力データを $K$ 個のグループに分けますがデータは完全に一致することは珍しいので工夫が必要です

- $N$ 個の入力ベクトル  $\{x^{(1)}, x^{(2)}, \dots, x^{(N)}\}$  を  $K$  個のグループに分ける
- 先の例では完全に一致するデータがあったのでグループ分けは自明

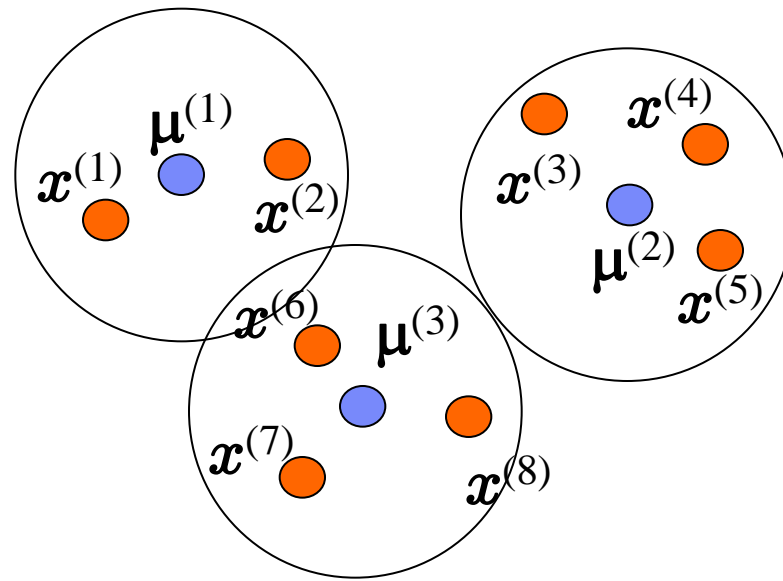
		足の数	
		8本	10本
かたさ	やわらかい	くも たこ	いか
	かたい	たらばがに やどかり	毛がに えび

- 通常はそうではないので、グループ分けは自明でない



ひとつのアプローチは、グループごとの代表点を考え、代表点への距離でグループ所属をはかることです

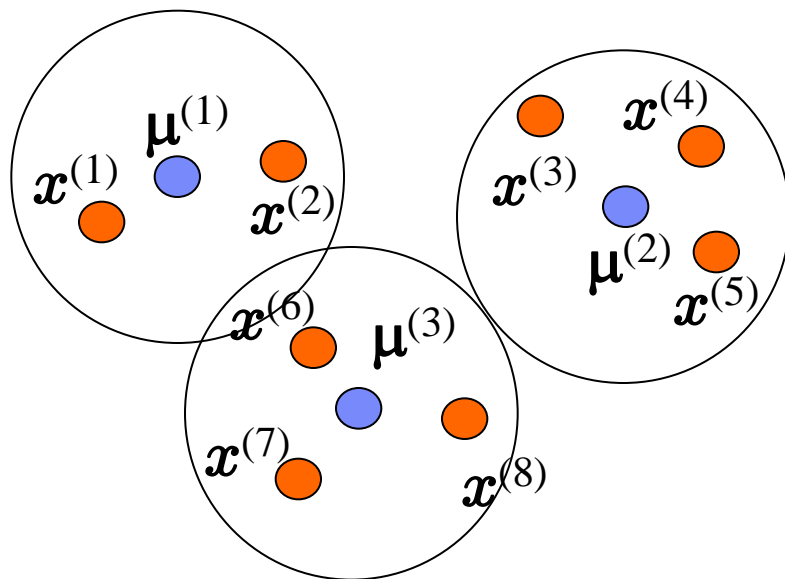
- $K$ 個のグループそれぞれの代表点 $\{\mu^{(1)}, \mu^{(2)}, \dots, \mu^{(K)}\}$ を考える
- 代表点に近い入力データは、そのグループに属するとする



- 代表点への「近さ」（距離）はどう定義するか？
  - 距離関数  $d(\mu^{(k)}, x^{(i)})$  を目的によって適切に定義する

## 距離関数を適切に定義する必要があります

- 代表点に近い入力データは、そのグループに属するとする

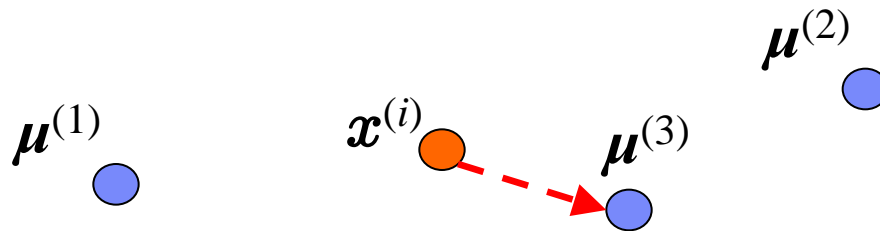


- 代表点への「近さ」（距離）はどう定義するか？
- 距離関数  $d(\mu^{(k)}, x^{(i)})$  を目的によって適切に定義する必要がある
  - たとえばユークリッド距離  $d(\mu^{(k)}, x^{(i)}) = \|\mu^{(k)} - x^{(i)}\|_2^2$
- 距離関数の定義によって結果が変わってくる

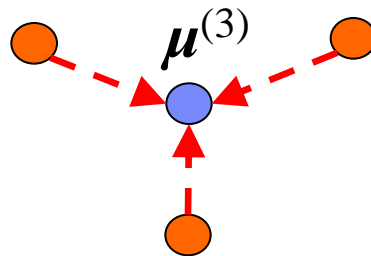
# K-meansアルゴリズム：グループ割り当てと代表点推定を交互に行うアルゴリズムです

- 以下のステップを収束するまで繰り返す

1. 各データ $x^{(i)}$ を、最寄の代表点 $\mu^{(k)}$ に割り当てる



2. 各代表点に所属したデータの平均として代表点を新たに求める  
(ユークリッド距離の場合)



---

## 教師なし学習の応用例

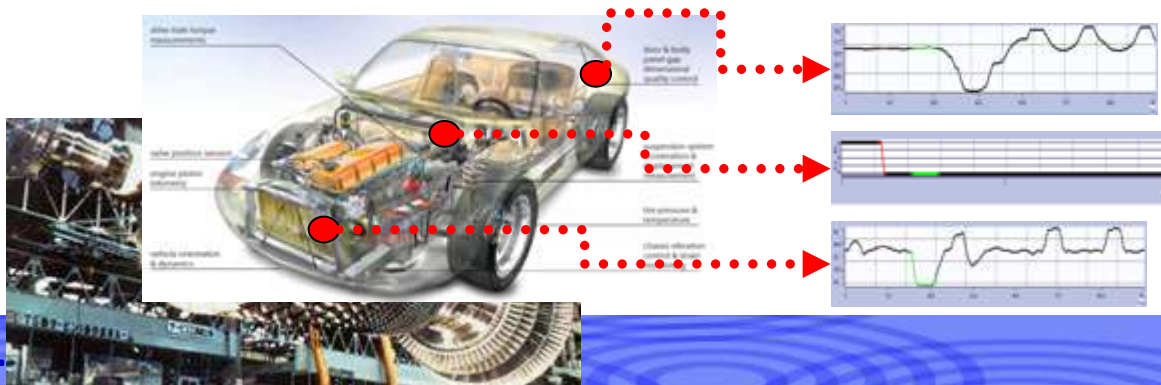
# 教師なし学習の応用例：異常検知

「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

機械／プラントなど

センサーからの  
取得データ



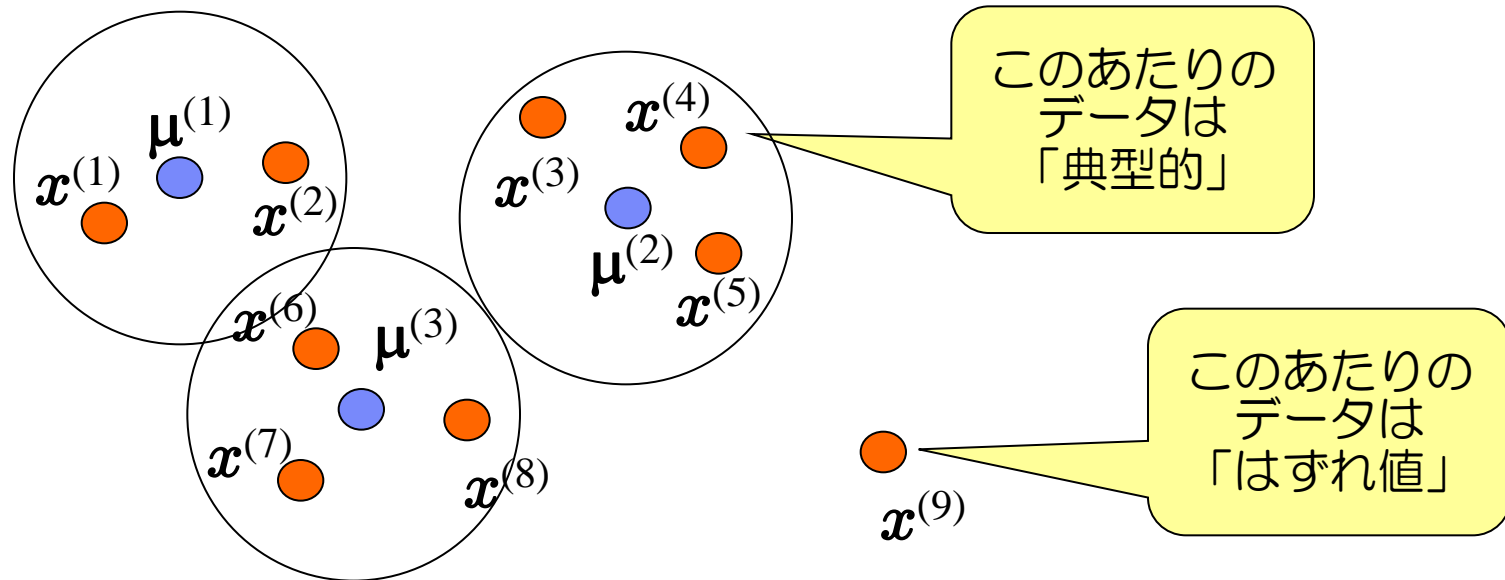
異常  
(の前触れ)



# 教師なし学習の応用例：異常検知

## グループに属さないデータ＝異常 と考えます

- 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グループ分けを行う
  - 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
- 代表点から遠い  $x$  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



---

おわりに

## 今日は機械学習の初歩的な手法を紹介しました

---

- 機械学習：教師つき学習と教師なし学習
- 教師つき学習の例：パーセプトロン
- 教師なし学習の例： $K$ -平均クラスタリング

## 機械学習は、簡単だけどつぶしの効く、お得な技術です

---

- データあるところには、機械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
- 一方で、まだまだビジネスの現場において、機械学習技術は十分に入り込んでいない。
- さまざまな領域でデータ収集インフラが整った今、今後は蓄積されたデータを、どう価値にかえていくかが課題となっている
- まさに今、もっとも旬な、つぶしの効く技術
- その他のモデル：回帰、混合分布、カーネル法、グラフィカルモデル
- その他の応用：推薦システム、ユーザーモデリング、需要予測