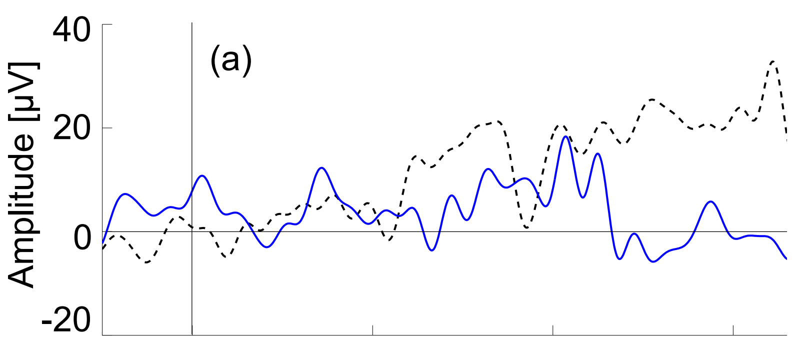
機械学習セミナー

@Life is tech! 大阪オフィス

自己紹介

- 名前：真木勇人 まきはやと(twitter: @mkhyt)
- 所属：奈良先端科学技術大学院大学 (NAIST)
  - 情報科学研究科 D1 知能コミュニケーション研究室
- 専門
  - 信号処理、機械学習
  - 研究テーマ：機械学習を利用した脳情報の分解・解読



今日の目的

- 将来機械学習を使うかもしれないエンジニアに、機械学習の原理、使用上・ビジネス上のポイントを知ってもらう。

機械学習
とは

機械学習 Machine Learning

- いわゆる「人工知能」の基盤技術

音声認識

手書き文字認識

顔画像検出

対話システム（Siri）

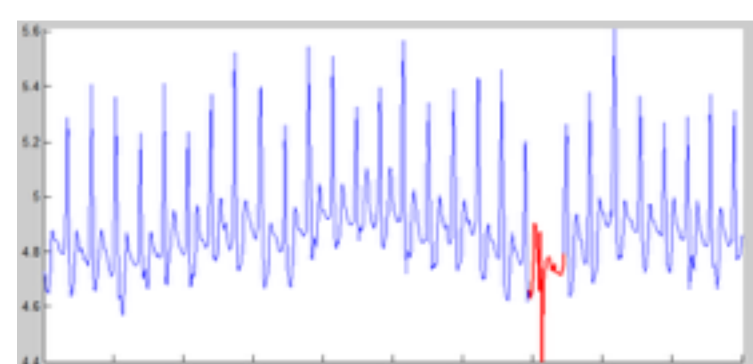
amazon

レコメンドシステム

Google Translate
Break through language barriers.

機械翻訳

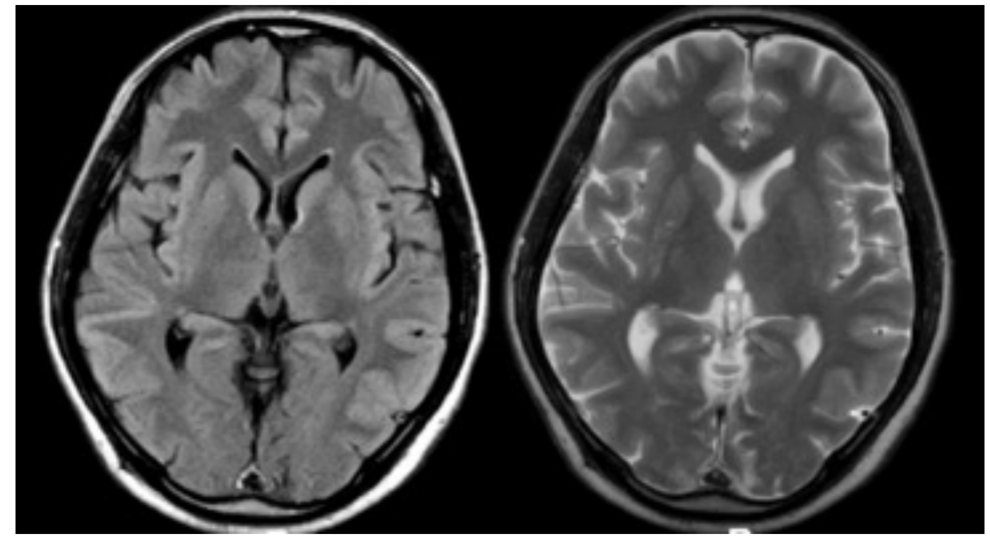
ユーザークラスタリング



異常検出

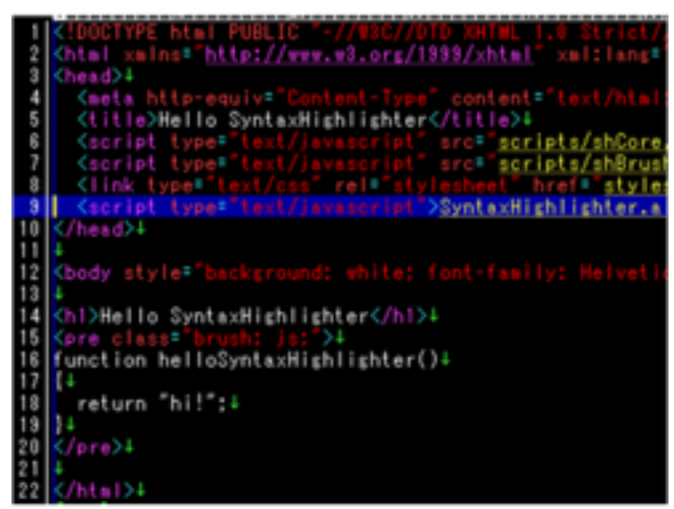
将来的な応用

自動運転

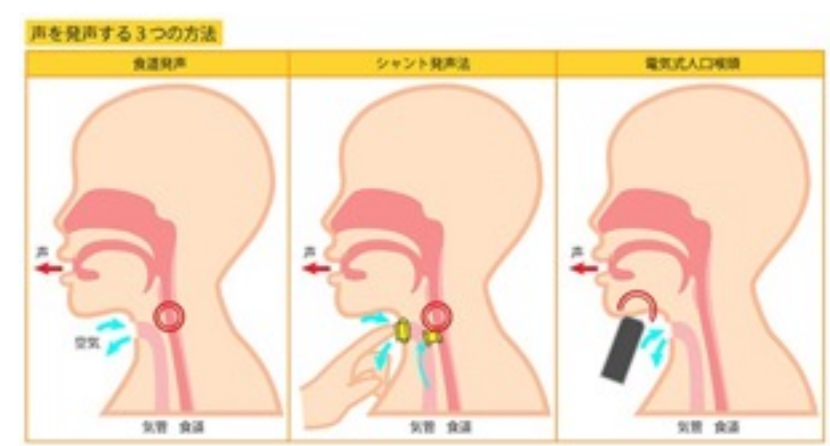


医療画像診断

同時音声翻訳



ソースコード生成



リアルタイム声質変換

機械に

データから法則性(ルール)を

自動的に

学習(発見)させる方法

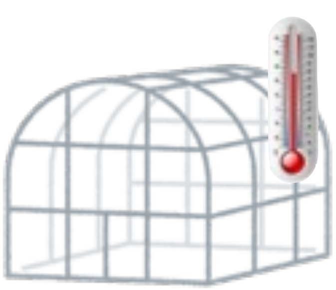
機械学習の
モチベーション

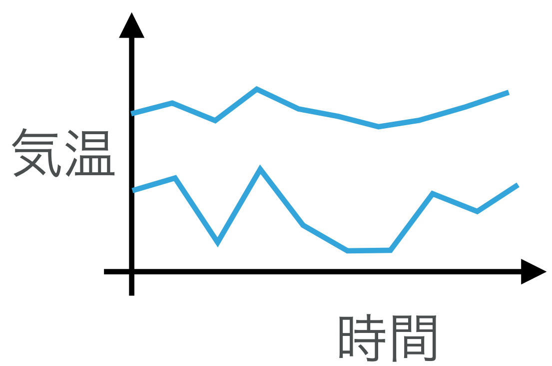
例：ビニールハウスの害虫発生予測

「データサイエンティスト養成読本機械学習入門編」（技術評論社）から改変して引用

- ビニールハウスの内外に温度計・湿度計が1つずつ設置
- 温度と湿度がある条件を満たすと、ビニールハウス内に害虫が発生

生データ



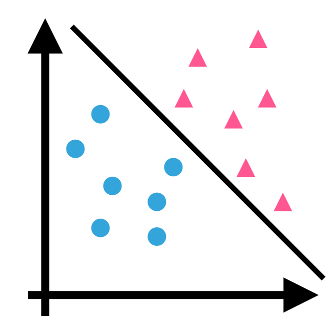


特徴抽出 →

特徴ベクトル

$$\mathbf{x} = \begin{bmatrix} \text{外気温平均} \\ \text{内気温平均} \\ \text{外湿度平均} \\ \text{内湿度平均} \\ \text{当日の季節} \end{bmatrix}$$

$\mathbf{x}$ → 入力 → **予測モデル（ルール）** → 出力 → アラート飛ばす / アラート飛ばさない



ルールをどうやって決める？

人手でルールを決めてみよう

- 過去のデータを眺めてみる（架空のデータ）

夏＝0, 冬＝1　なし＝0, あり＝1

	外気温(℃)	内気温(℃)	外湿度(%)	内湿度(%)	季節	害虫発生
data1	33.5	37.1	70.2	72.4	0	1
data2	35.4	41.5	53.2	55.8	0	0
data3	31.8	35.4	63.3	62.0	0	1
data4	24.7	28.0	68.9	70.0	0	0
data5	10.6	25.2	61.1	63.8	1	1
data6	5.1	22.1	44.5	52.9	1	0
data7	6.3	20.4	70.7	75.3	1	0
data8	12,5	23.6	62.7	77.9	1	1

「内気温が30℃以上」ならアラート？

「夏かつ内気温30℃以上」または「冬かつ内気温25℃以上」ならアラート？？

「夏かつ内気温30℃以上かつ内湿度60%以上」または「冬かつ内気温25℃以上かつ内湿度%60以上」ならアラート？？？

人手でルールを決めてみよう

- 過去のデータを眺めてみる（架空のデータ）

夏＝0, 冬＝1　なし＝0, あり＝1

	外気温(℃)	内気温(℃)	外湿度(%)	内湿度(%)	季節	害虫発生
data1						1
data2						0
data3						1
data4						0
data5	10.6	25.2	61.1	63.8	1	1
data6						0
data7						0
data8						1

高次元かつ大量のデータに対して人手で法則性（ルール）を構築するのは困難

ルールの構築を自動化するのが機械学習(機械にルールを発見させる)

「内気温か

「夏かつ内気温30℃以上」または「冬かつ内気温25℃以上」ならアラート？？

「夏かつ内気温30℃以上かつ内湿度60%以上」または「冬かつ内気温25℃以上かつ内湿度%60以上」ならアラート？？？

機械学習の手法

- **教師あり学習** Supervised Learning
  - 分類 Classification
  - 回帰 Regression
- **教師なし学習** Unsupervised Learning
  - クラスタリング Clustering
  - 次元削減 Dimensionality Reduction
  - 異常検出 Anomaly Detection

教師あり学習

x を入力して、y を予測する

$$\mathbf{y} = f(\mathbf{x})$$

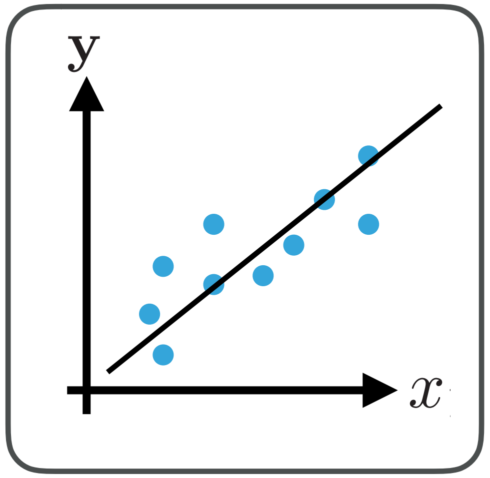
- **トレーニングデータ**（**過去**のデータ）使って、**関数 f (予測モデル)を推定**する問題（関数近似問題）
- **トレーニングデータ: 特徴ベクトルと正解ラベルの事例セット**

x = (x1, x2) = (気温, 湿度) ,　y = 害虫発生あり or 発生なし

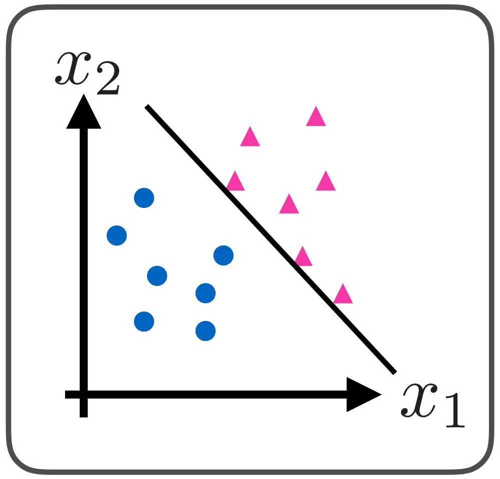
Day1	x = (30, 70)	y = あり
Day2	x = (24, 65)	y = なし
⋮	⋮	⋮

回帰と分類

- **回帰**
  - 予測値が**数値**である問題
    - 例1) 気温からテーマパークの来場者数を予想する
    - 例2) 年齢と喫煙本数から残りの寿命を予想する



- **分類**
  - 予測値が**クラス**である問題
    - 例1) 単語からスパムメールを見分ける
    - 例2) 手書きの文字を認識する



回帰モデルの学習と予測

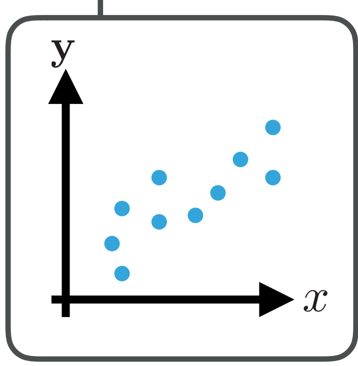
- 例：非雇用率から犯罪発生率を予想する

$\mathbf{x}$ =(非雇用率)

$\mathbf{y}$ =(犯罪発生率)

トレーニングデータ

	特徴ベクトル	正解ラベル
都市1	$\mathbf{x}^{(tr1)}$	$\mathbf{y}^{(tr1)}$
都市2	$\mathbf{x}^{(tr2)}$	$\mathbf{y}^{(tr2)}$
	⋮	⋮

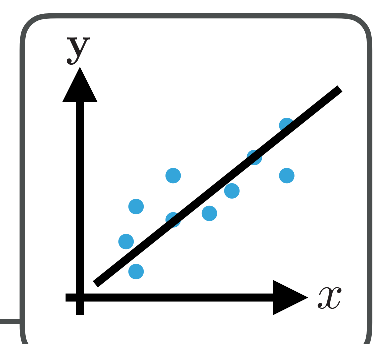


学習アルゴリズム リッジ回帰 ニューラルネットなど

予測モデル f を推定

$$f(\mathbf{x}) = \underline{w_1}\mathbf{x} + \underline{w_0}$$

推定

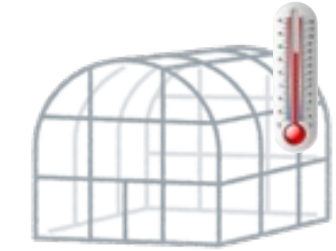


未知のデータ $\mathbf{x}_{\mathrm{new}}$

予測モデル

予測値 $\mathbf{y}_{\mathrm{new}} = f(\mathbf{x}_{\mathrm{new}})$

分類モデルの学習と予測



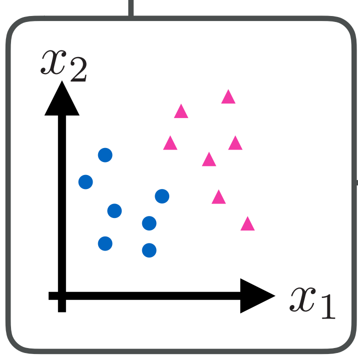
- 例：気温と湿度から害虫の発生を予測する

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \text{温度} \\ \text{湿度} \end{pmatrix}$$

$\mathbf{y}$ = あり or なし

トレーニングデータ

	特徴ベクトル	正解ラベル
Day1	$\mathbf{x}^{(\mathrm{tr1})}$	$\mathbf{y}^{(\mathrm{tr1})}$
Day2	$\mathbf{x}^{(\mathrm{tr2})}$	$\mathbf{y}^{(\mathrm{tr2})}$
	⋮	⋮

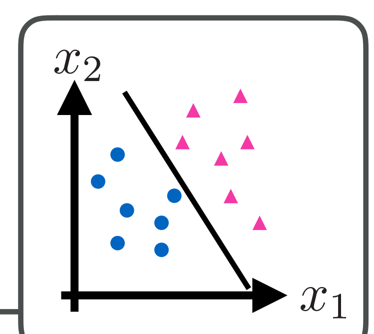


学習アルゴリズム パーセプトロン SVMなど

予測モデル f を推定

$$f(\mathbf{x}) = \underbrace{w_1}_{\text{推定}} x_1 + \underbrace{w_2}_{\text{推定}} x_2 + \underbrace{w_0}_{\text{推定}}$$

推定



未知のデータ $\mathbf{x}_{\mathrm{new}}$

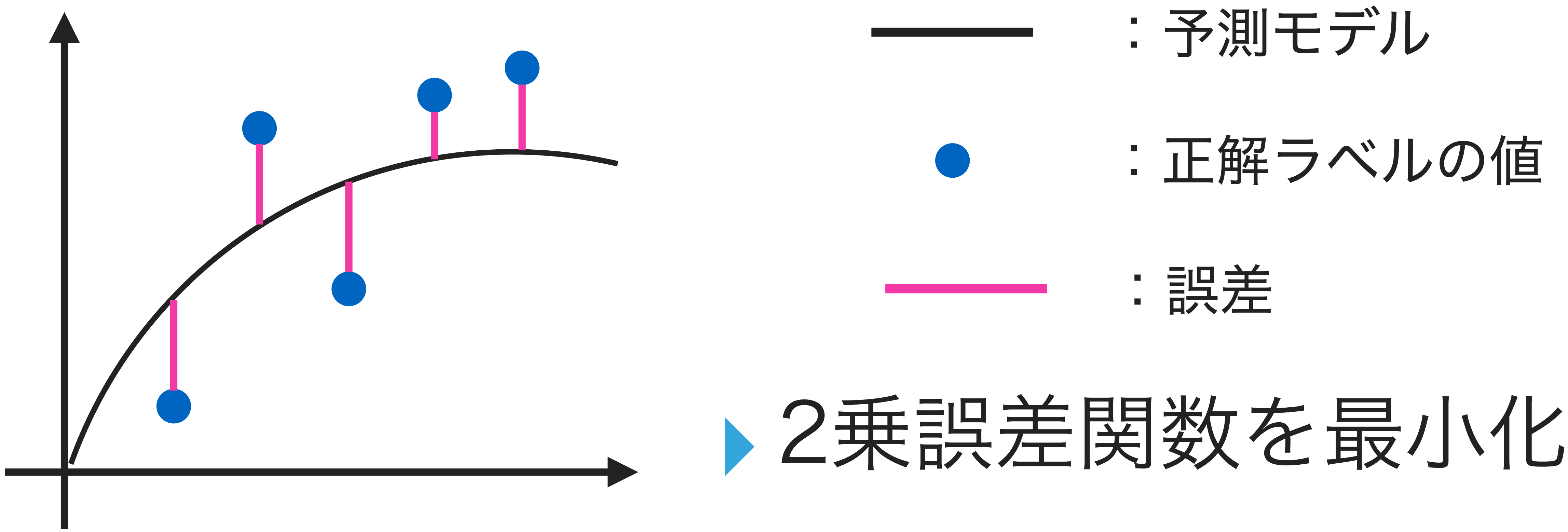
予測モデル

予測値

$$\mathbf{y}_{\mathrm{new}} = \begin{cases} \text{あり}\,(f(\mathbf{x}_{\mathrm{new}}) > 0) \\ \text{なし}\,(f(\mathbf{x}_{\mathrm{new}}) < 0) \end{cases}$$

最小二乗学習

- 多くの機械学習アルゴリズムの原型



- 2乗誤差関数を最小化

$$J(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}\left(f_{\mathbf{w}}(\mathbf{x}_i) - \mathbf{y}_i\right)^2$$

前処理

正規化

- 年収と年齢から、残りの寿命を予想する

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \text{年収} \\ \text{年齢} \end{pmatrix} \begin{matrix} \longleftarrow \\ \longleftarrow \end{matrix} \quad \begin{matrix} \text{200万} \sim \text{3000万} \\ \text{18歳} \sim \text{80歳} \end{matrix}$$

予測モデル： $f(\mathbf{x}) = w_1 x_1 + w_2 x_2 + w_0$

年収が相対的に大きく影響してしまう

- 平均0、分散1になるように、特徴量ごとに正規化

$$x_1' = \frac{x_1 - \mu}{\sigma}$$

μ : x1の平均

σ : x1の標準偏差

ダミー変数

- 年収と居住地から、残りの寿命を予想する

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \text{年収} \\ \text{居住地} \end{pmatrix} \begin{matrix} \longleftarrow \\ \longleftarrow \end{matrix} \begin{matrix} \text{200万 〜 3000万} & \text{数値} \\ \text{関東、関西、中部} & \text{カテゴリ} \end{matrix}$$

予測モデル： $f(\mathbf{x}) = w_1 x_1 + w_2 x_2 + w_0$

- カテゴリ変数を扱えるようにダミー変数を導入

$$f(\mathbf{x}) = w_1 x_1 + \underline{w_2 x_{\text{関東}}} + \underline{w_3 x_{\text{関西}}} + \underline{w_4 x_{\text{中部}}} + w_0$$

該当するところは1、他は0

機械学習を
使いこなす
ために

機械学習のメリット

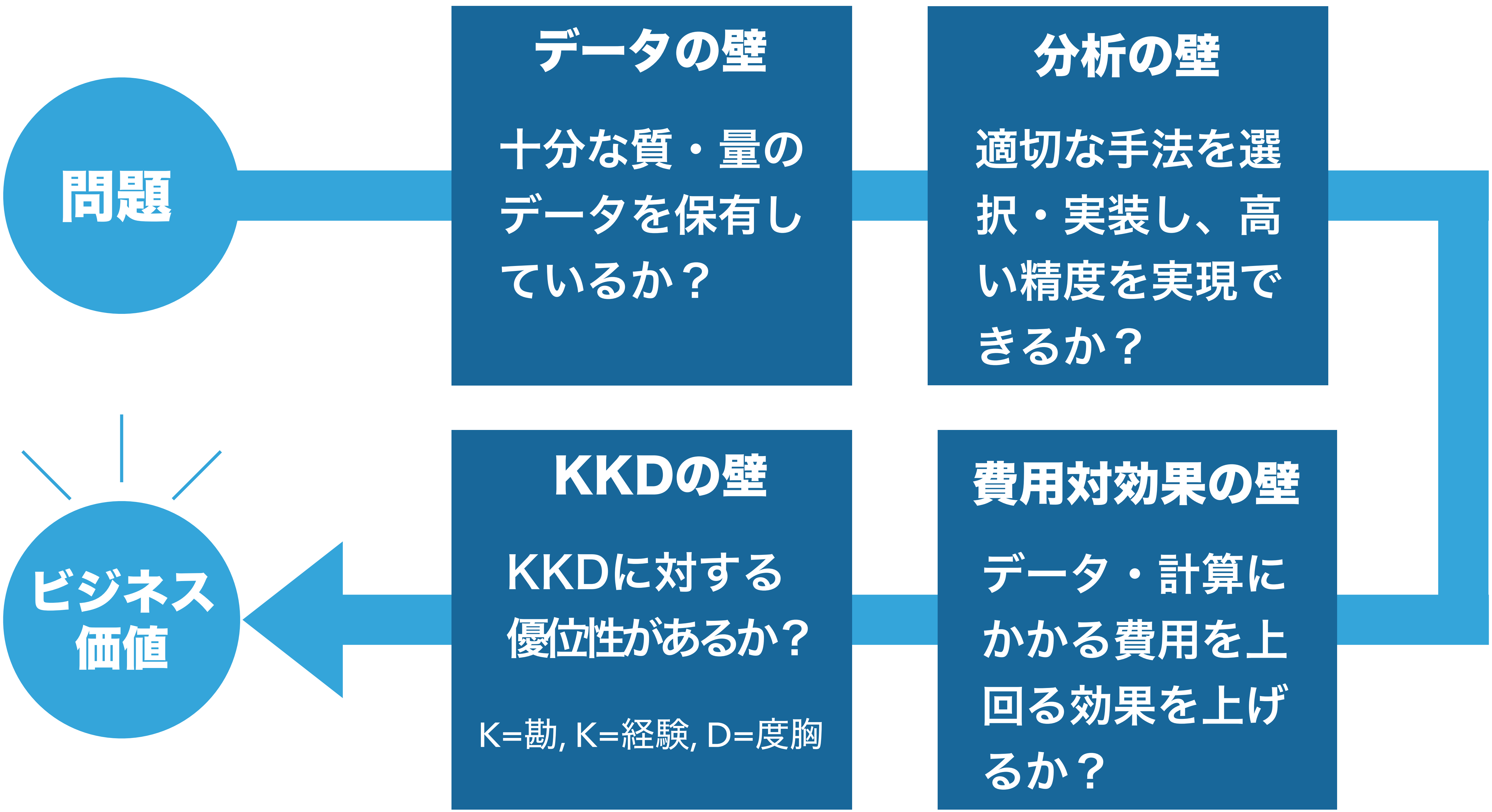
- **メリット**
  - アルゴリズムが汎用的、様々な問題に適用可能
  - (うまく学習すれば)人間を上回る精度・速度を実現可能
  - 人間には扱いきれない高次元・大量なデータを取り扱い可能
  - (場合により)コスト削減

機械学習のデメリット

- **デメリット**
  - 大量かつ良質なデータが必要
    - 欠損値、フォーマット不揃い、網羅性
    - データ前処理ニスト？
  - 計算に長時間または豊富なマシンパワーが必要な場合がある
  - 結果の解釈が容易でない場合がある
  - 特徴量の選択、ハイパーパラメタの調整など、すべて自動になるわけではない

ビジネス利用への4つの壁

- 河本「会社を変える分析の力」講談社　← めっちゃ良い本



おまけ：人工知能は人間を超えるか

- 答え：問題によりけり
  - 画像認識は人間を超えたといわれている
  - 音声認識は人間の方が遥かに優れている
    - 当分超える見込はないと思う（個人の見解）

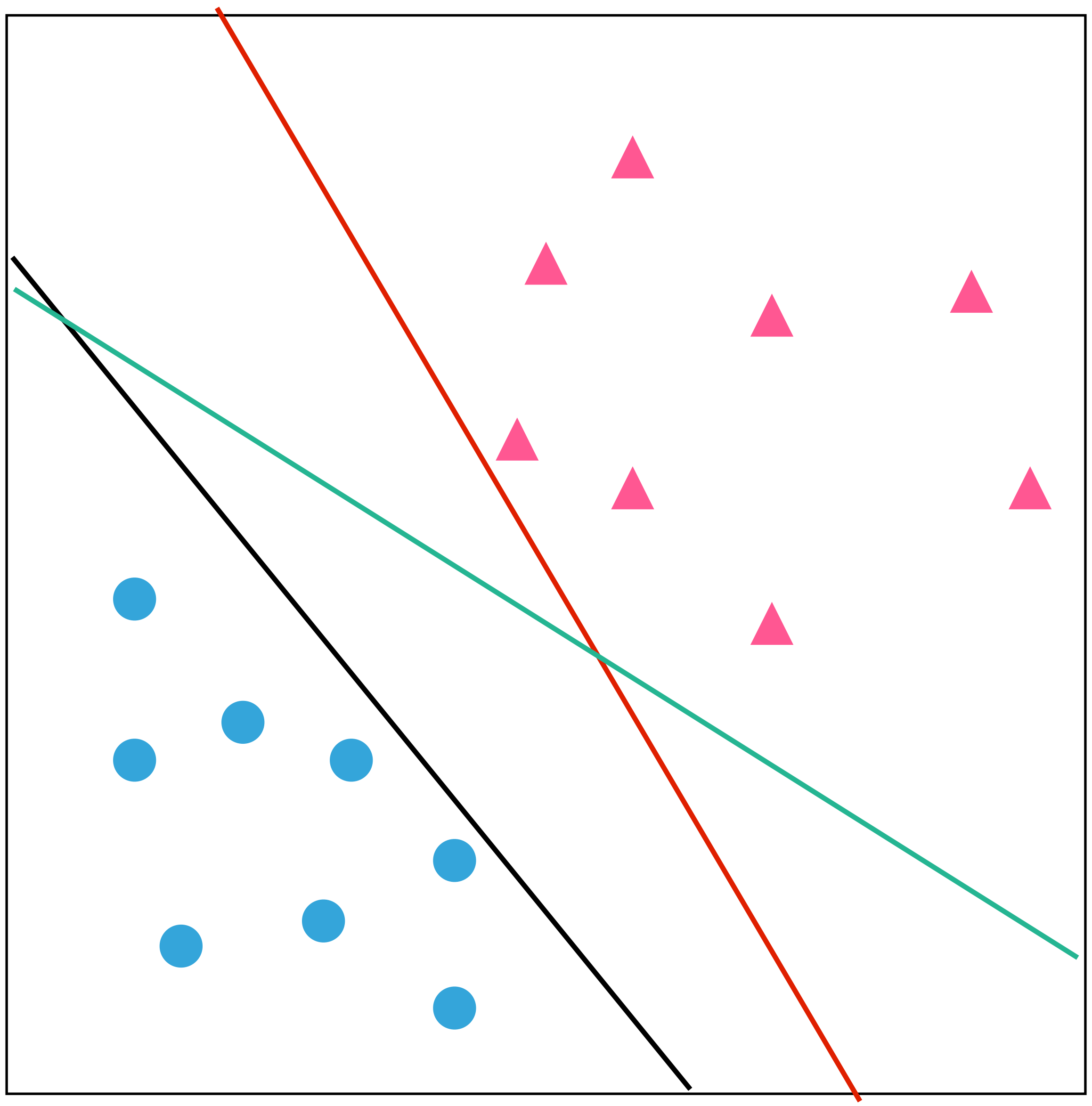
応用編
Support Vector Machine

SVMの戦略

- マージン最大化

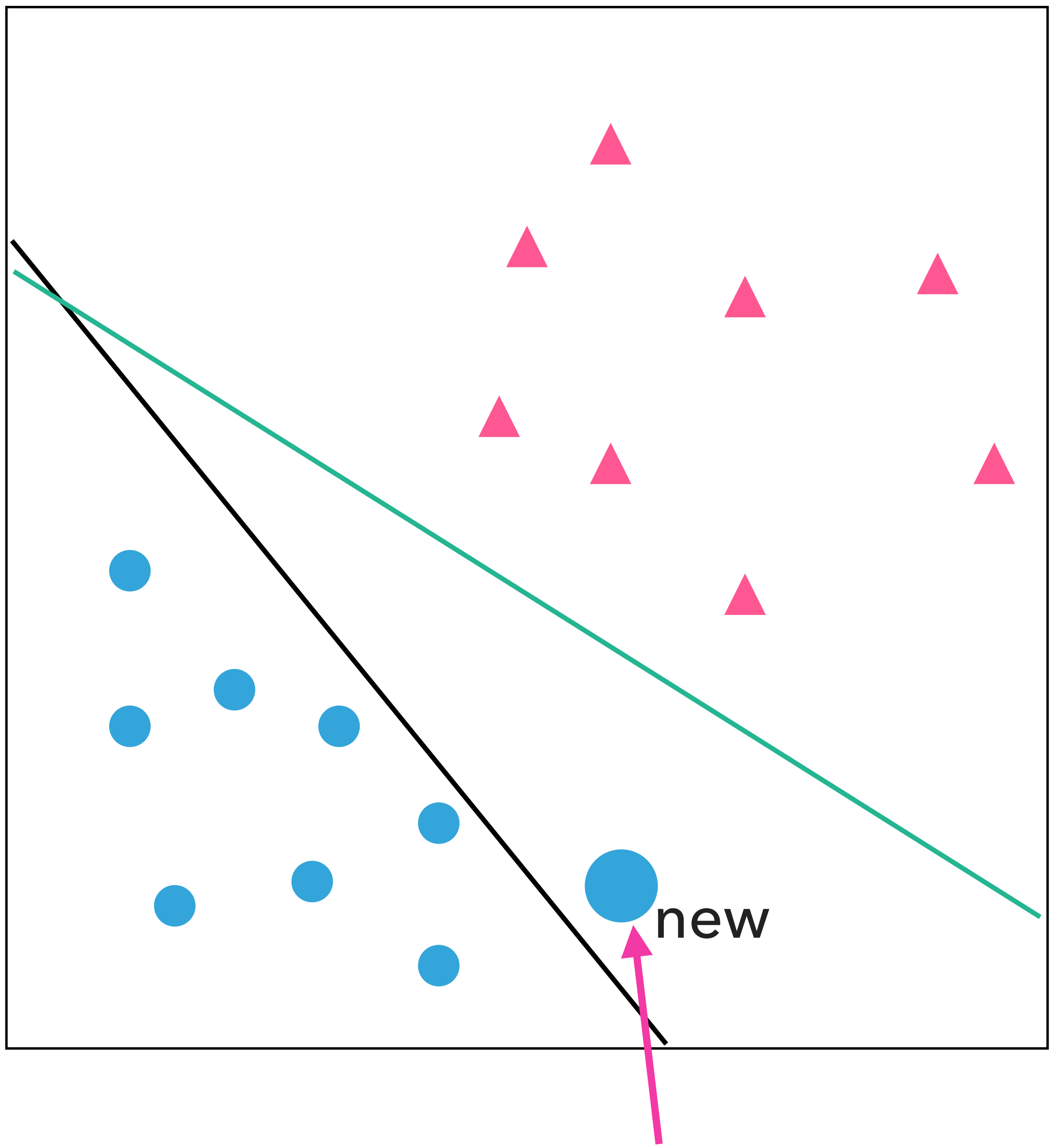
  カーネル法による非線形化

パーセプトロン



- データが線形分離可能なら、必ず決定境界を見つけ出す
- どの決定境界に収束するか不確定(初期値に依存)

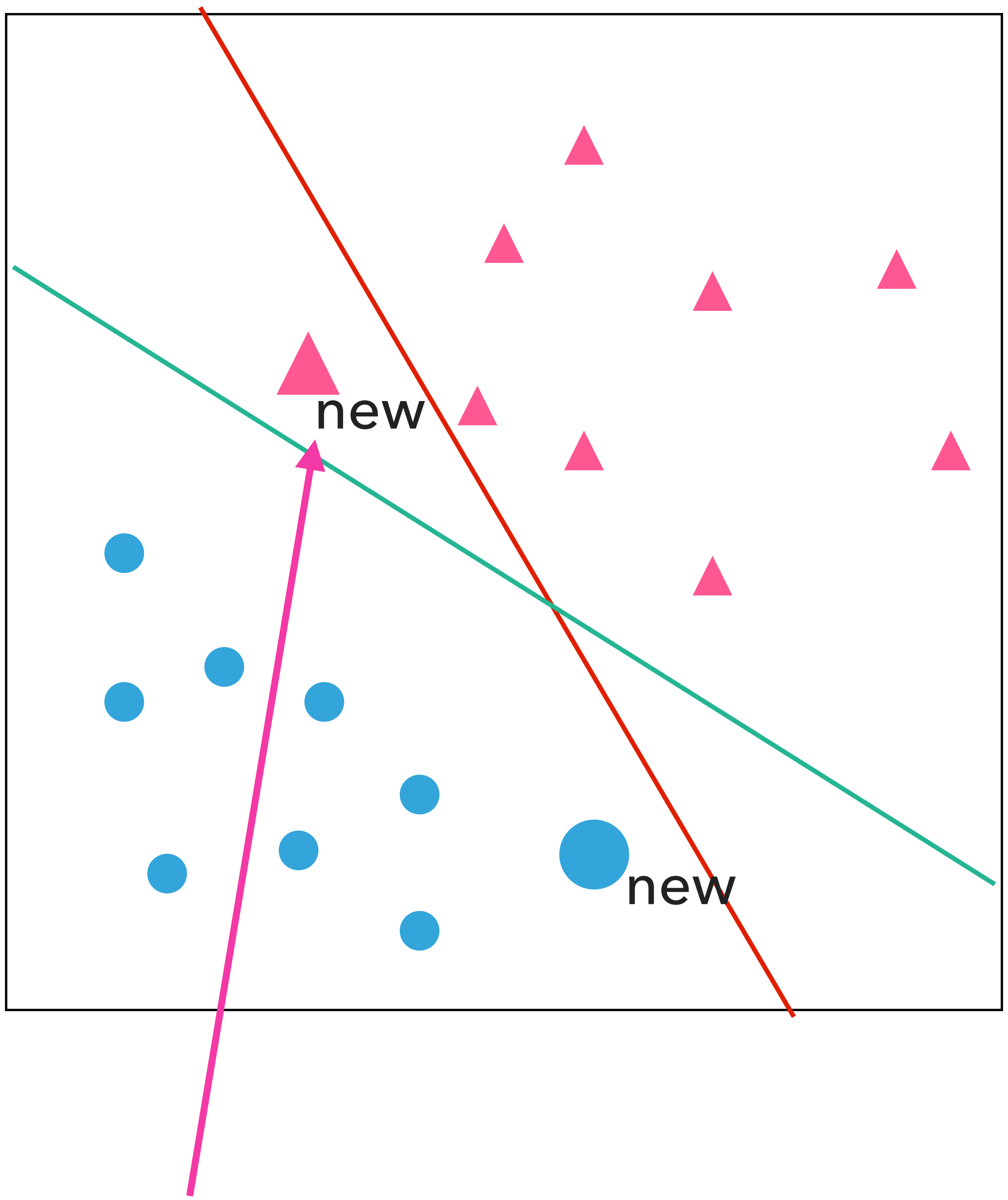
パーセプトロン



- データが線形分離可能なら、必ず決定境界を見つけ出す
- どの決定境界に収束するか不確定(初期値に依存)

黒い決定境界だと誤分類！

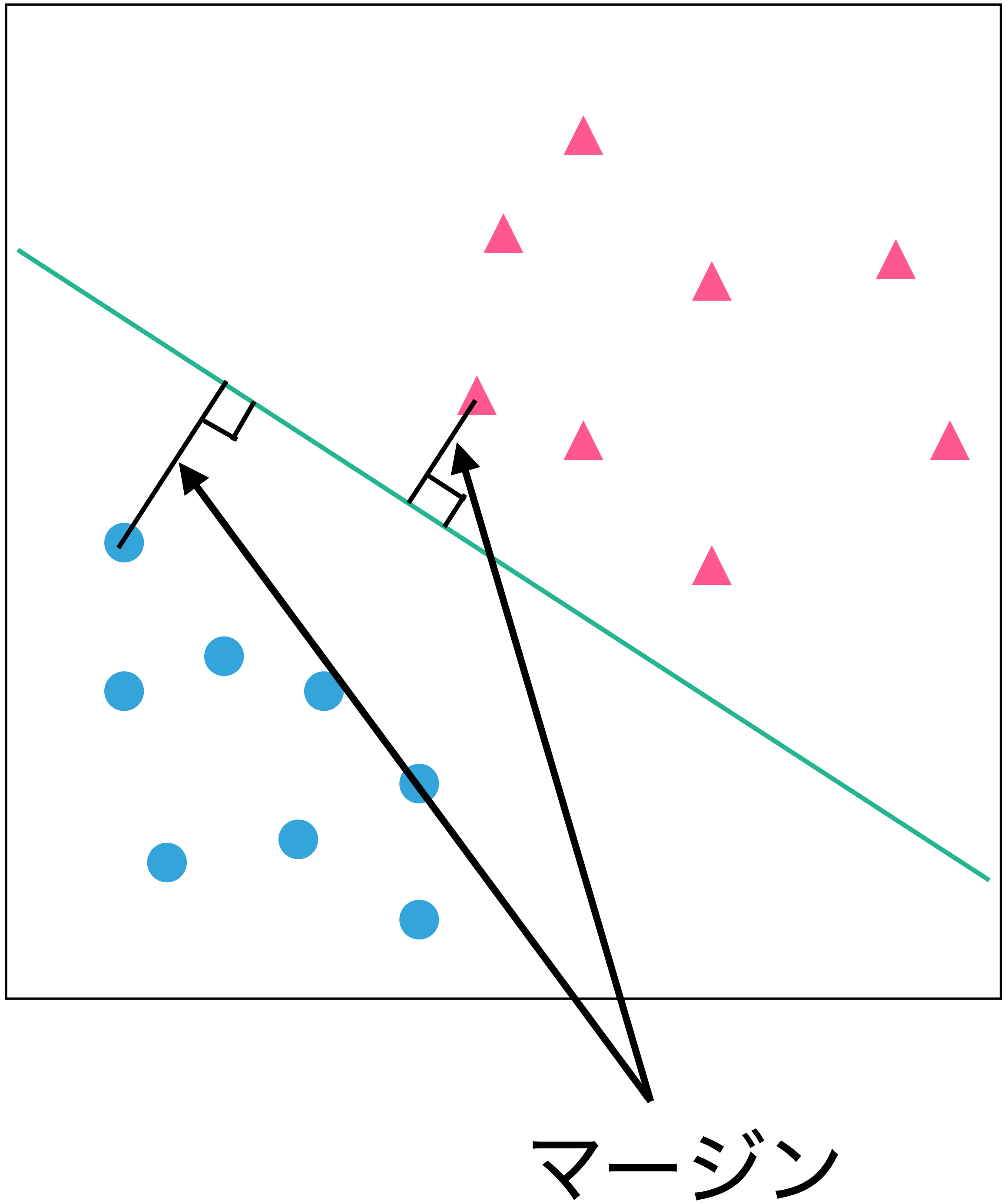
パーセプトロン



赤い決定境界だと誤分類！

- データが線形分離可能なら、必ず決定境界を見つけ出す
- どの決定境界に収束するか不確定(初期値に依存)

パーセプトロン



- データが線形分離可能なら、必ず決定境界を見つけ出す
- どの決定境界に収束するか不確定(初期値に依存)
- マージン最大化！
- 線形分離のみ

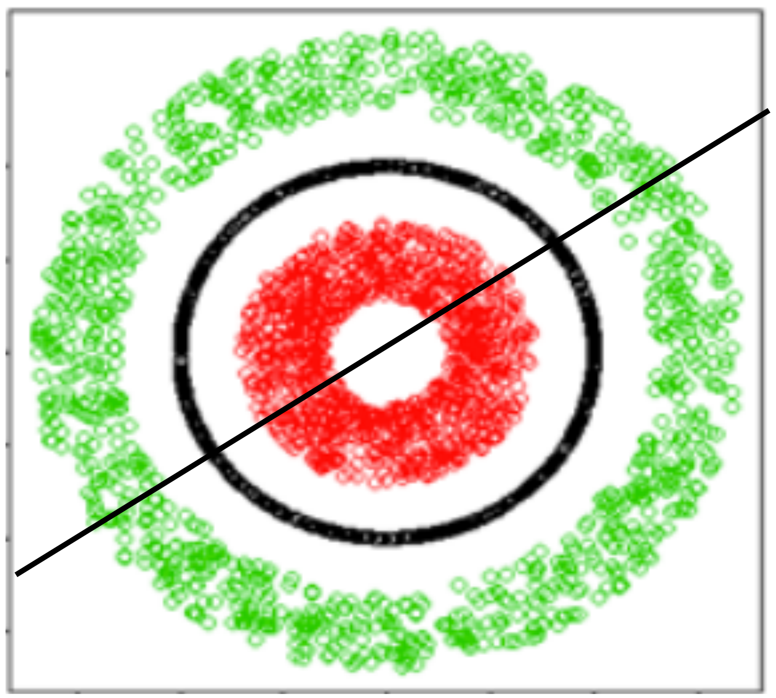
SVMの戦略

- マージン最大化
- カーネル法による非線形化

やや上級者向け

高次元空間への写像

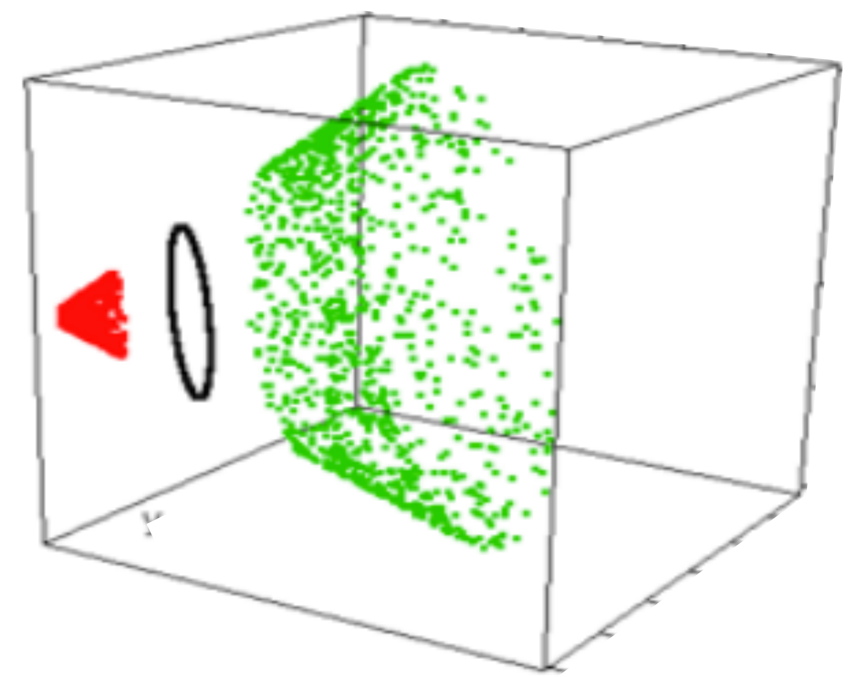
線形分離不能



3次元空間へ写像

写像 ϕ

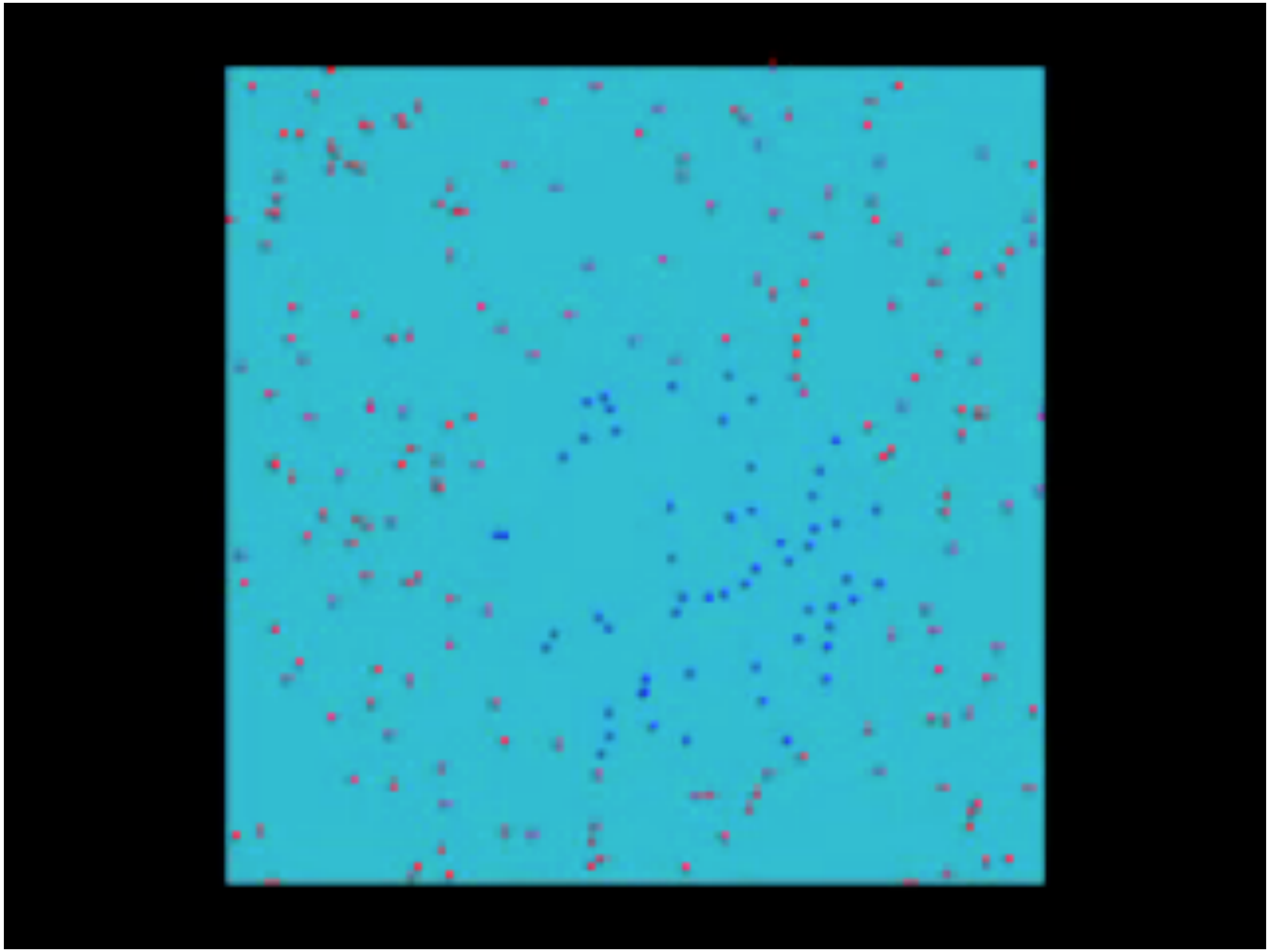
線形分離可能



$$\mathbf{x} = (x_1, x_2)$$

$$\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1, x_2)$$

画像出所：http://sudillap.hatenablog.com/entry/2013/04/08/235610



https://www.youtube.com/watch?v=3liCbRZPrZA&feature=youtu.be

計算量の爆発 (次元の呪い)

- 2次以下の特徴を抽出する写像

$$\mathbf{x} = (x_1, \cdots, x_{1000})$$ 1000次元

$$\downarrow \phi$$

$$\phi(\mathbf{x}) = (x_1^2, \cdots, x_1 x_2, \cdots, 1)$$ 501501次元

	表現力	計算量
低次元	低い	少ない
高次元	高い	多い

高次元の表現力を実現しつつ、**計算は低次元**で行なうアイディア

→ **カーネル法！**

カーネルトリック

- SVMの定式化

$$\text{max.} \quad -\frac{1}{2}\sum_{i,j}\lambda_i\lambda_j y^{(i)}y^{(j)}\left(\phi(\mathbf{x}^{(i)})\cdot\phi(\mathbf{x}^{(j)})\right)+\sum_i \lambda_i$$

$$\text{s.t.} \quad \sum_i \lambda_i y^{(i)} = 0 \quad \lambda_i \geq 0$$

(λ はラグランジュ乗数)

カーネルトリック

- SVMの定式化

高次元化した特徴ベクトル

$$\text{max.} \quad -\frac{1}{2}\sum_{i,j}\lambda_i\lambda_j y^{(i)} y^{(j)} \boxed{\left(\phi(\mathbf{x}^{(i)})\cdot\phi(\mathbf{x}^{(j)})\right)} + \sum_i \lambda_i$$

$$\text{s.t.} \quad \sum_i \lambda_i y^{(i)} = 0 \qquad \lambda_i \geq 0$$

(λ はラグランジュ乗数)

内積さえ計算できればOK
$\phi(\mathbf{x})$ にアクセスする必要無し

内積の性質を満たす関数K（カーネル関数）で置き換えて計算

$$K\left(\mathbf{x}^{(i)}\cdot\mathbf{x}^{(j)}\right) = \left(\phi(\mathbf{x}^{(i)})\cdot\phi(\mathbf{x}^{(j)})\right)$$

カーネル関数 1/2

多項式カーネル $$K\left(\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}\right) = \left(\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)} + r\right)^d$$

↑ 変形

d次以下のすべての項を
持つ特徴ベクトルの内積

カーネル関数 1/2

多項式カーネル $K\left(\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}\right) = \left(\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)} + r\right)^d$

変形

d次以下のすべての項を持つ特徴ベクトルの内積

計算は低次元！

高次元化した $\phi(\mathbf{x})$ ではなく
$\mathbf{x}$ についての式なので

表現力は高次元！

$\phi(\mathbf{x})$ の内積に等しいので

カーネル関数 2/2

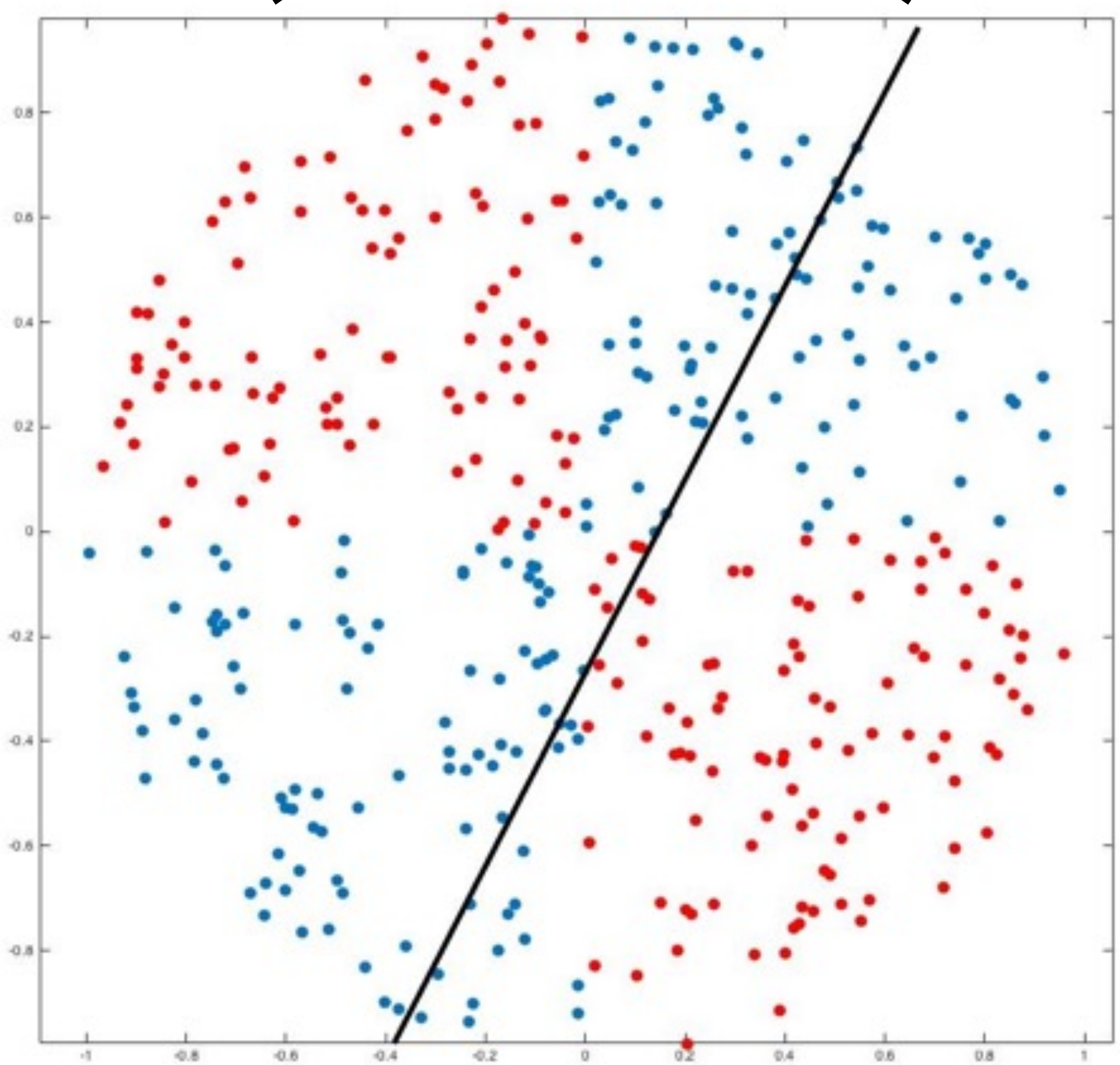
RBFカーネル $$K\left(\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}\right) = \exp\left(-s|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}|^2\right)$$

↑ 変形

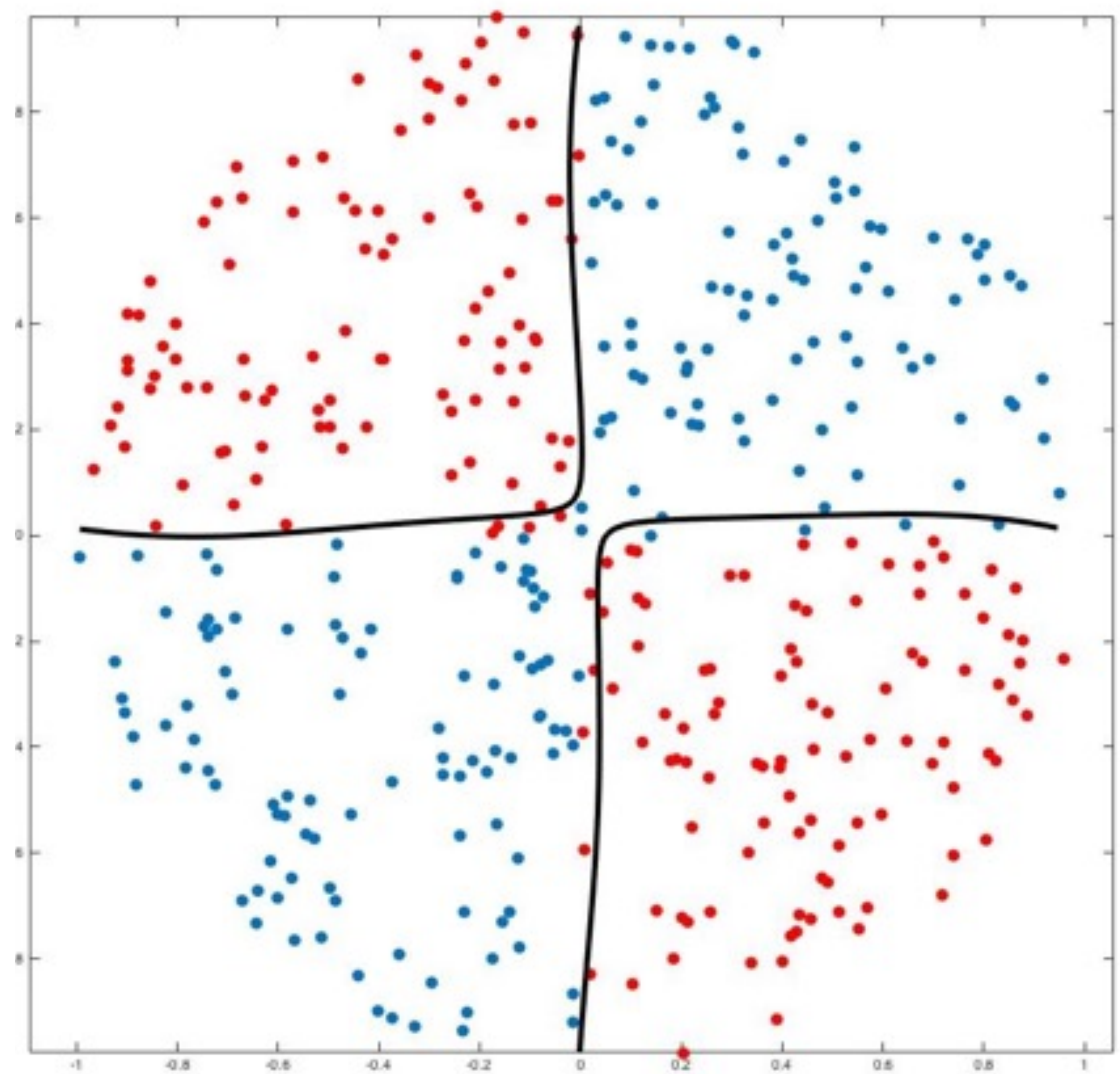
∞次元の特徴ベクトルの内積

カーネル法による決定境界

カーネルなし
(線形カーネル)



RBFカーネル



参考文献

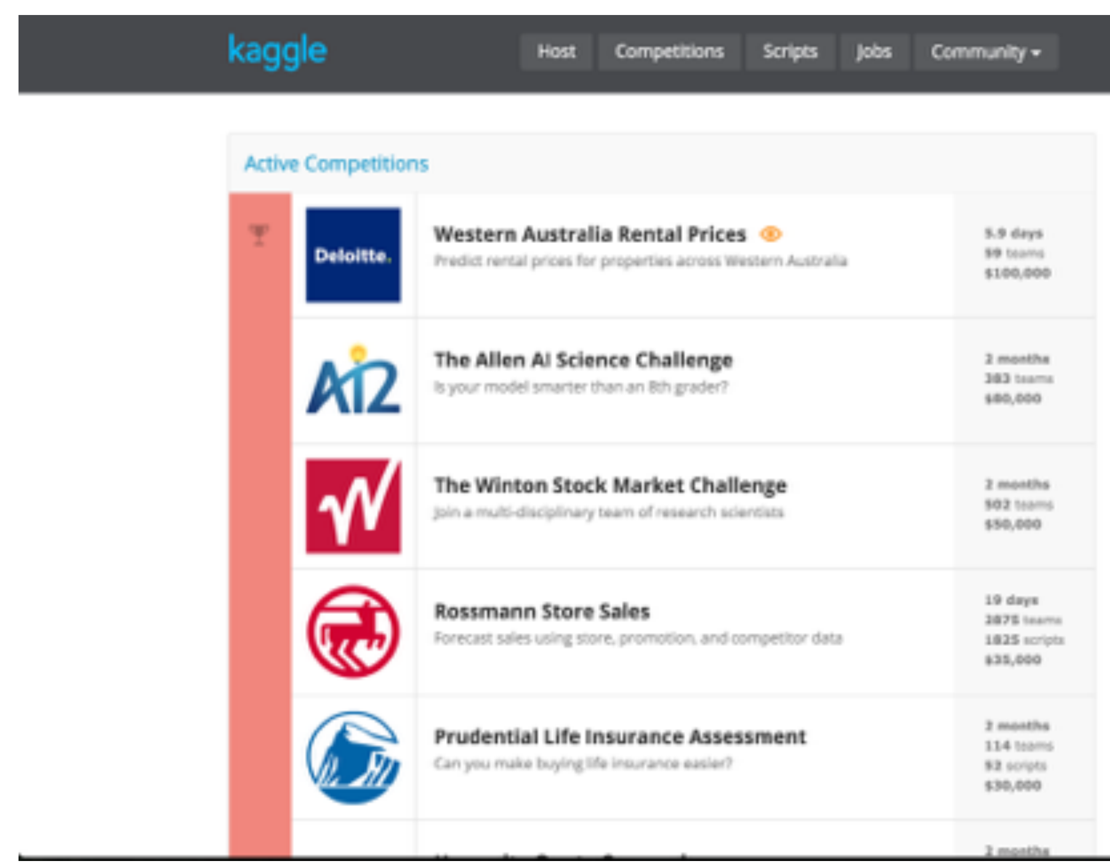
- 比戸 他, データサイエンティスト養成読本 機械学習入門編, 技術評論社
- 河本, 会社を変える分析の力, 講談社
- 杉山, イラストで学ぶ機械学習 最小二乗法による識別モデル学習を中心に, 講談社
- Bishop, Pattern Recognition and Machine Learning, Springer
- 高村, 自然言語処理のための機械学習入門, コロナ社

実践パート

Kaggleに挑戦してみよう

Kaggle

- 世界最大のデータサイエンティストコミュニティ
- データ解析のコンペティションを多数開催
    - 賞金が出る
    - 他企業の協賛コンペ多数、ジョブマッチング



タイタニック生存者予想

- Kaggleのチュートリアルコンペ
- タイタニック搭乗者のプロフィールから、その人が生きて帰ったかどうかを予想する
- トレーニングデータ：891人分
- テストデータ：418人分

データに含まれる情報

- Pclass: 搭乗者のクラス (1st, 2nd, 3rd)
- Name, Sex, Age, Fare(料金)
- SlibSp: 同乗した兄弟または配偶者の数
- Parch: 同乗した親または子供の数
- Ticket: チケット番号
- Cabin: 客室
- Embarked: 出発港 (Cherbourg, Queenstown, Southampton)

Pythonライブラリ

- **numpy, scipy**: 数値計算ライブラリ
- **pandas**: データ解析ライブラリ
- **scikit-learn**: 機械学習ライブラリ
- **matplolib**: グラフ描写ライブラリ
- **IPython**: 対話型シェル

- 1つずつ入れるとめんどいので、**Anaconda**おすすめ
- Kaggleのサイト上でも動かすことができる

コードを書いてみよう

- STEP1
  scikit-learnの使い方を調べ、SVMで学習と予測をおう
  できたら、出力部分のコメントアウトを解除して実行、
  Kaggleに提出して精度を確認

- STEP2
  「Fare」と「Age」をそれぞれ正規化した変数「NorFare」
  と「NorAge」を作ろう
  できたら、FareとAgeの代わりに特徴ベクトルに追加しよう
  再度実行、Kaggleに提出して精度を確認

以上

コンタクト：@mkhyt on twitter