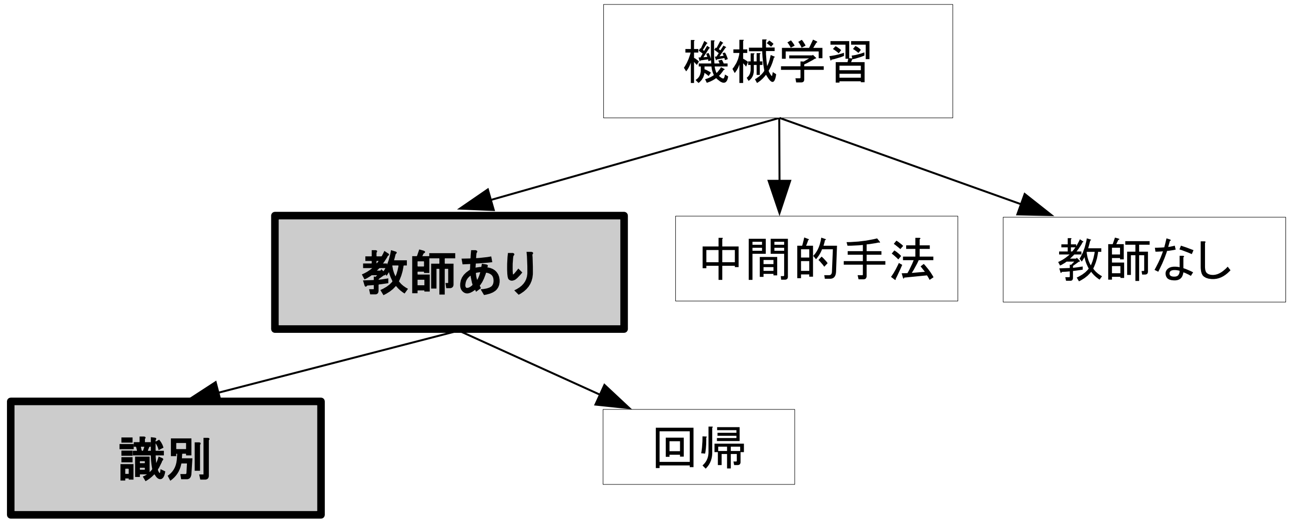
Section 2

- 識別2（4〜7章）
- 機械学習ライブラリの紹介



- カテゴリ特徴

年齢 = 若年
眼鏡 = 近視
乱視 = なし
涙量 = 正常

→ 推薦レンズ = ソフト

- 数値特徴

気温 = 20
湿度 = 60
降水確率 = 0

→ Play = Yes

4.1 統計的識別とは

max $f(x)$: $f(x)$ の最大値
argmax $f(x)$: $f(x)$ が最大となる x

- 最大事後確率則による識別

$$C_{MAP} = \arg\max_i P(\omega_i|\boldsymbol{x})$$

$\boldsymbol{x}$ ：特徴ベクトル

$\omega_i \quad (1 \leq i \leq c)$ ：クラス

  - データから直接的にこの確率を求めるのは難しい

- ベイズの定理 $P(A|B) = \dfrac{P(B|A)P(A)}{P(B)}$

$$\begin{aligned} C_{MAP} &= \arg\max_i P(\omega_i|\boldsymbol{x}) \\ &= \arg\max_i \frac{P(\boldsymbol{x}|\omega_i)P(\omega_i)}{P(\boldsymbol{x})} \\ &= \arg\max_i P(\boldsymbol{x}|\omega_i)P(\omega_i) \end{aligned}$$

4.1 統計的識別とは

- 事前確率 $P(\omega_i)$
  - 特徴ベクトルを観測する前の、各クラスの起こりやすさ
- 事前確率の最尤推定

$$P(\omega_i) = \frac{n_i}{N}$$

N: 全データ数、n_i: クラス ω_i のデータ数

4.1 統計的識別とは

- 尤度 $P(\boldsymbol{x}|\omega_i)$
  - 特定のクラスから、ある特徴ベクトルが出現する尤もらしさ
- *d* 次元ベクトルの場合の最尤推定
  - 値の組合せがデータ中に出現しないもの多数

Viewer

Relation: weather.symbolic

No.	1: outlook Nominal	2: temperature Nominal	3: humidity Nominal	4: windy Nominal	5: **play** Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Undo OK Cancel

Weka の weather.nominal データ

3×3×2×2=36 種類の組合せ

4.2.2 ナイーブベイス識別

- ナイーブベイズの近似
  - 全ての特徴が独立であると仮定

$$P(\boldsymbol{x}|\omega_i) = P(x_1, \ldots, x_d|\omega_i)$$
$$\approx \prod_{j=1}^{d} P(x_j|\omega_i)$$

$$C_{NB} = \arg\max_i P(\omega_i) \prod_{j=1}^{d} P(x_j|\omega_i)$$

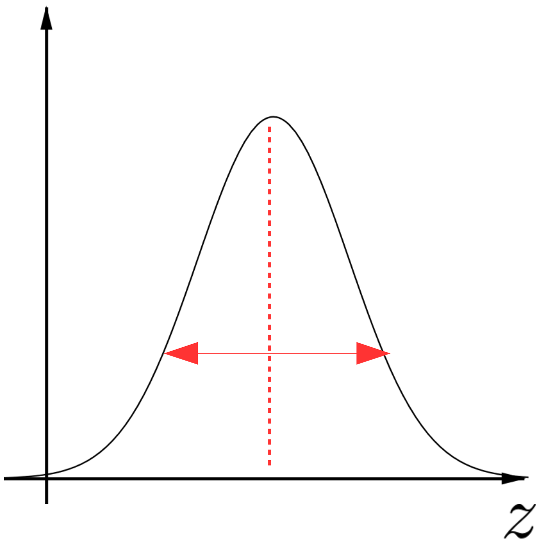
5.2 数値特徴に対するベイズ識別

5.2.1 数値特徴に対するナイーブベイズ識別

$$C_{NB} = \arg\max_i P(\omega_i) \prod_{j=1}^{d} p(x_j|\omega_i)$$

- 確率密度関数 $p(x_j|\omega_i)$ の推定
  - 正規分布を仮定

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(z-\mu)^2}{2\sigma^2})$$



  - 平均 μ と分散 σ を最尤推定
    - それぞれ、学習データの平均と分散になる

5.3.1 識別モデルの考え方

- 事後確率を直接求める

正例で大きな値

負例で大きな値

あまり関係ない

$$\boldsymbol{x} = (x_1, \quad x_2, \quad x_3, \quad \ldots, \quad x_d)$$

w_1 正の大きな値

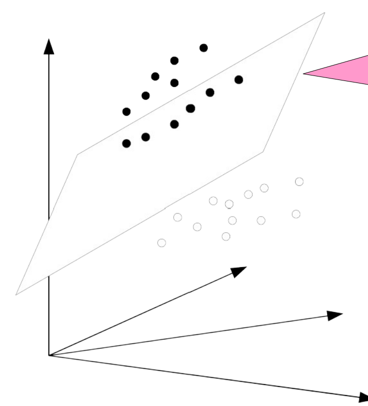
w_2 負の大きな値

w_3 小さな値

正例／負例の閾値

$$w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_dx_d + w_0$$

この値が正なら正例、負なら負例となるように重み w を学習する



この平面を求めていることになる

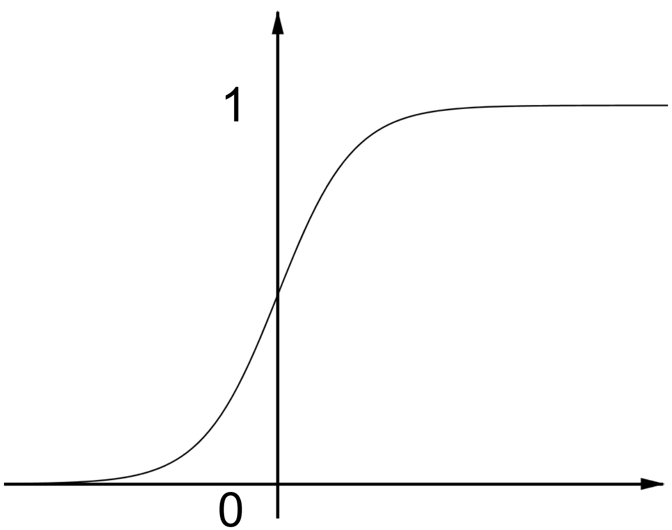
確率と対応づけるには？

5.3.1 識別モデルの考え方

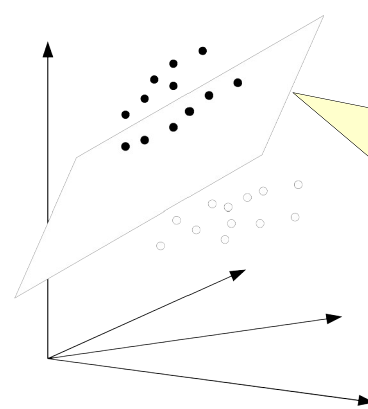
- ロジスティック識別
  - 入力が正例である確率

$$P(\oplus|\boldsymbol{x}) = \frac{1}{1 + \exp(-(\boldsymbol{w} \cdot \boldsymbol{x} + w_0))}$$

-∞ 〜 +∞ の値域を持つものを、順序を変えずに 0 〜 1 にマッピング



シグモイド関数



この平面上は
$\boldsymbol{wx}$+w_0=0
⇔ P(+|x)=0.5

5.3.2 ロジスティック識別器の学習

- 最適化対象＝モデルが学習データを生成する確率

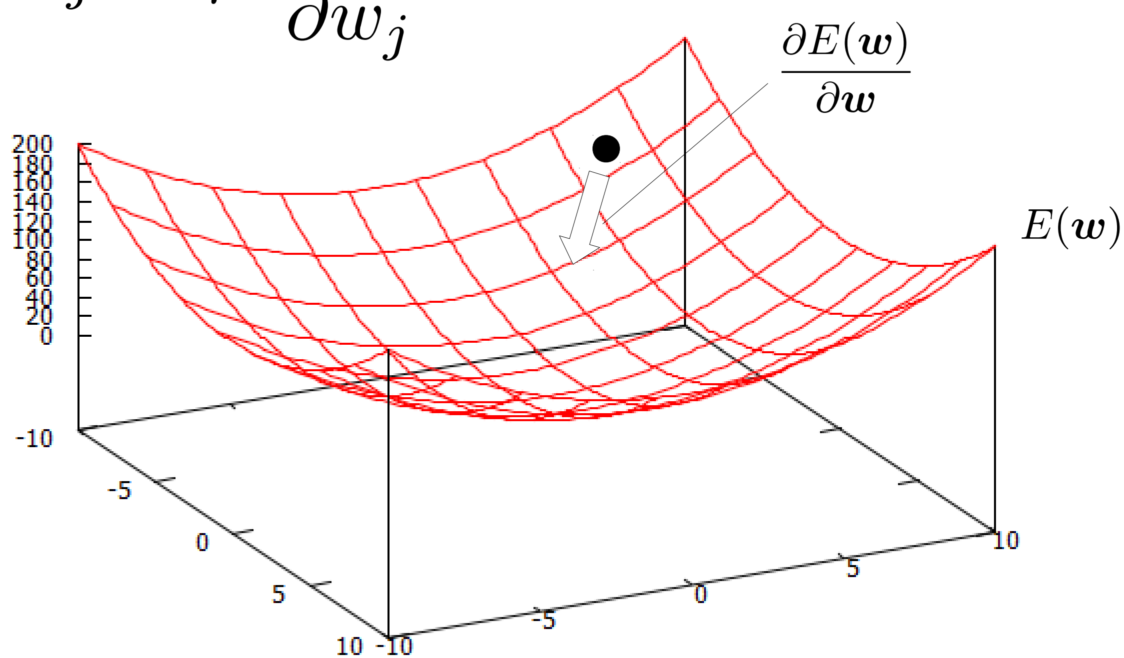
$$E(\boldsymbol{w}) = -\log P(D|\boldsymbol{w}) = -\log \prod_{\boldsymbol{x}_i \in D} o_i^{y_i}(1-o_i)^{(1-y_i)}$$

$$o = P(\oplus|\boldsymbol{x})$$
$$y = 0 \;\; or \;\; 1$$

正解ラベル

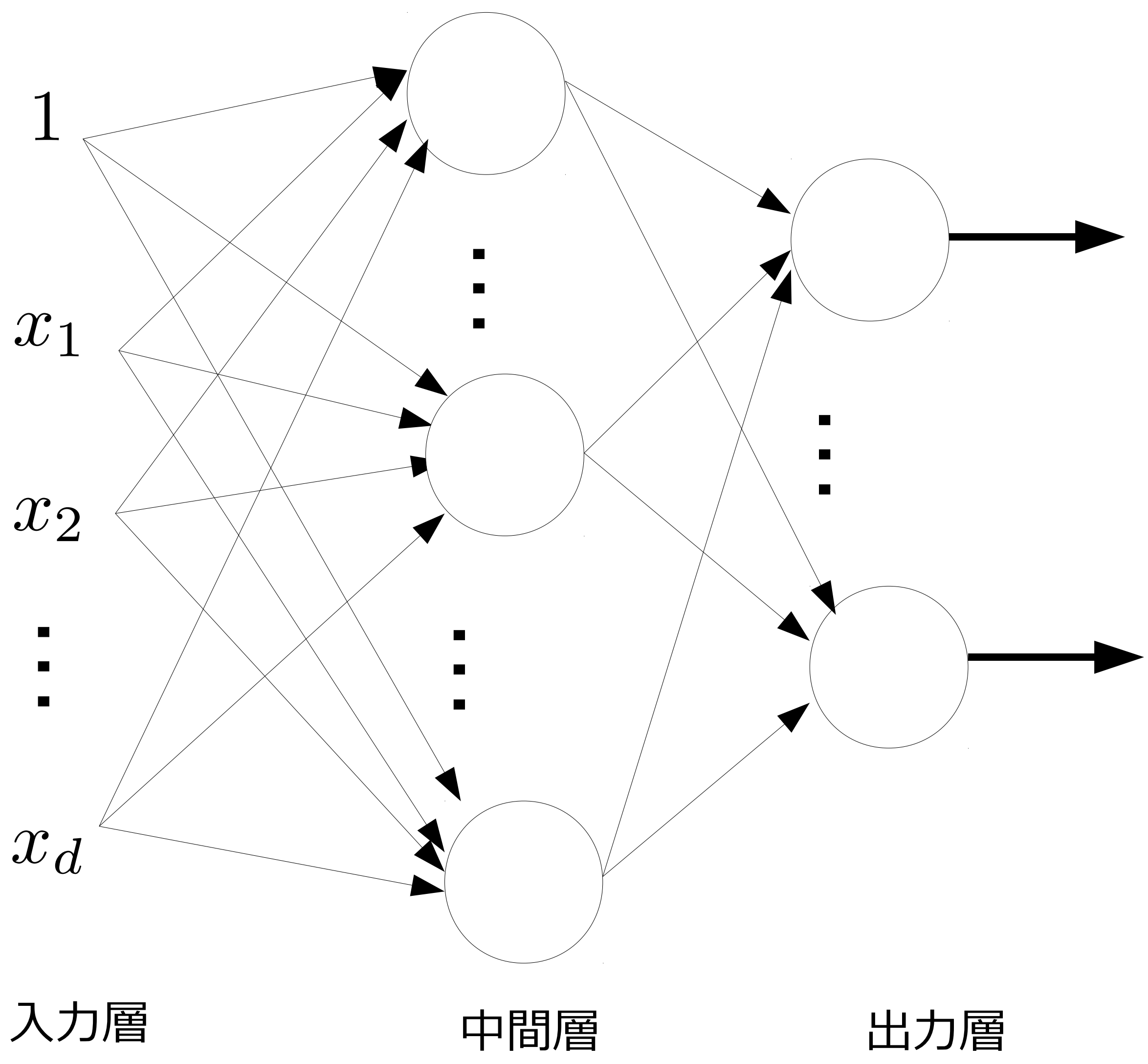
  - $E(\boldsymbol{w})$ を最急勾配法で最小化

$$w_j \leftarrow w_j - \eta \frac{\partial E(\boldsymbol{w})}{\partial w_j}$$



6.4 ニューラルネットワーク

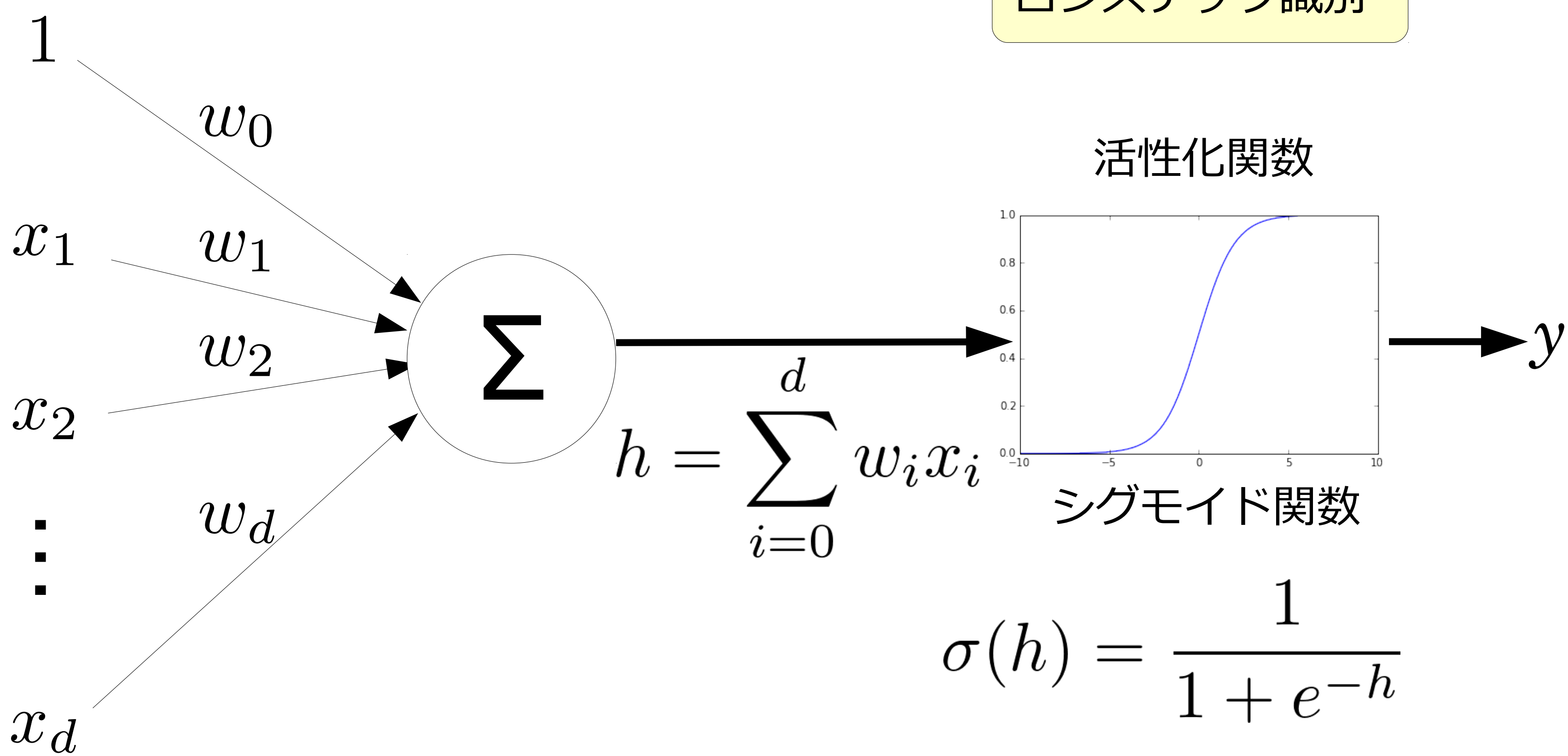
- 3 層のフィードフォワードネットワーク



6.3 最小二乗法による学習

- シグモイド関数の適用
  - 多層の誤差修正に対応するために、勾配計算の際に微分可能な活性化関数を用いる

ロジスティック識別



$$h = \sum_{i=0}^{d} w_i x_i$$

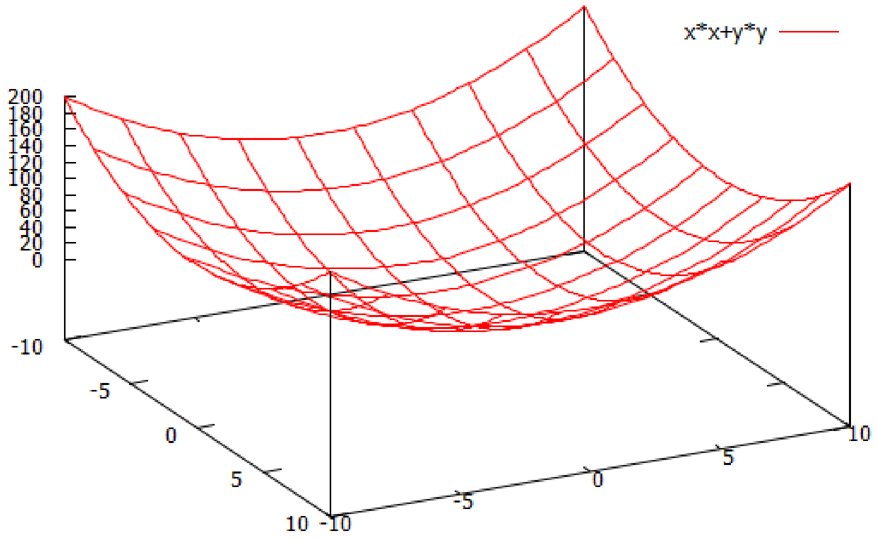
$$\sigma(h) = \frac{1}{1 + e^{-h}}$$

$$\sigma'(h) = \sigma(h) \cdot (1 - \sigma(h))$$

6.3 最小二乗法による学習

- エラーの定義
  - 二乗誤差 $$J(\boldsymbol{w}) \equiv \frac{1}{2}\sum_{\boldsymbol{x}_i \in D}(y_i - \boldsymbol{w}^T\boldsymbol{x}_i)^2$$
  - J は $\boldsymbol{w}$ の関数
    - $\boldsymbol{w}$ を J の勾配方向へ一定量だけ動かすことを繰り返して、最適解へ収束させる

全データに対する正解と関数の出力との差の２乗和

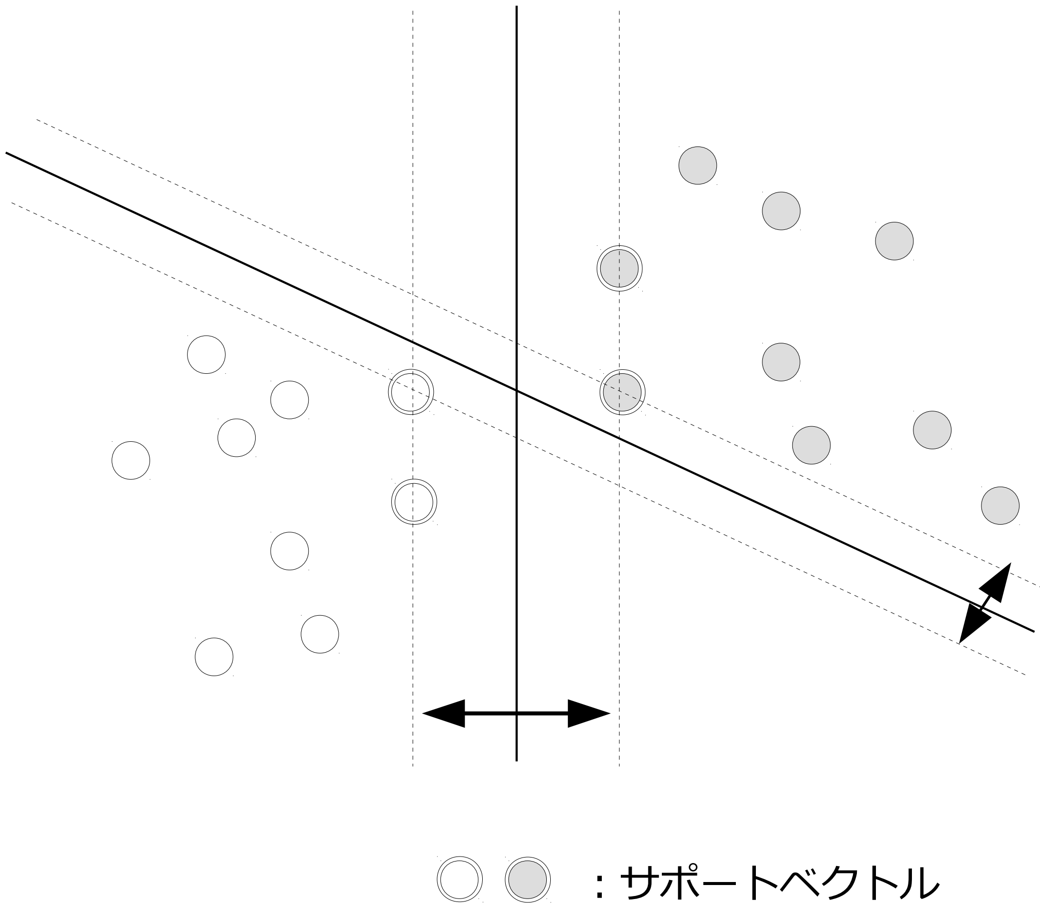


$$\boldsymbol{w}' = \boldsymbol{w} - \rho\frac{\partial J}{\partial \boldsymbol{w}}$$
$$= \boldsymbol{w} - \rho\sum_{p=1}^{n}(\boldsymbol{w}^T\boldsymbol{x}_p - b_p)\boldsymbol{x}_p$$

7. 識別 - サポートベクトルマシン -

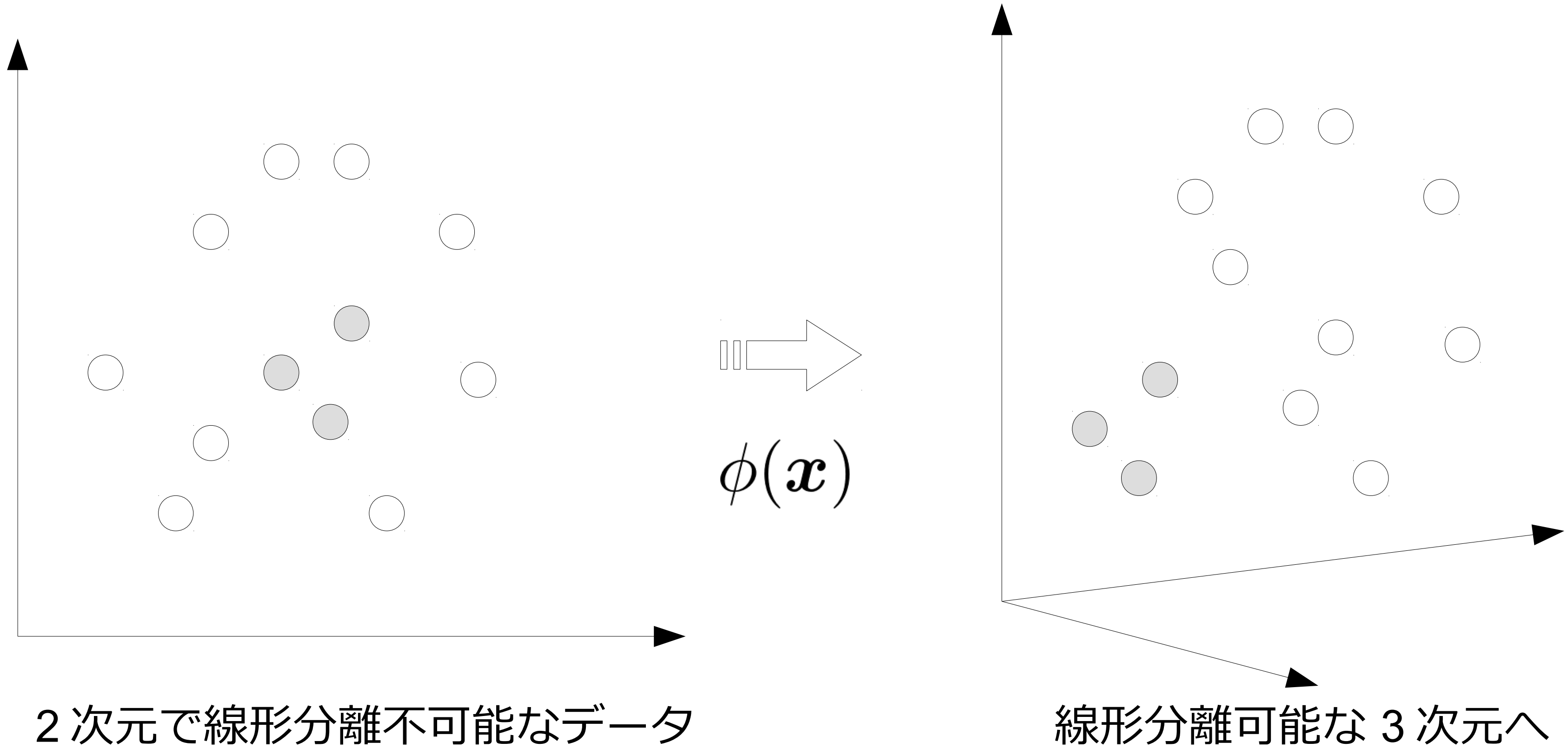
- マージンを最大化する識別面を求める

識別面と、最も近いデータとの距離



7.3 カーネル関数を用いた SVM

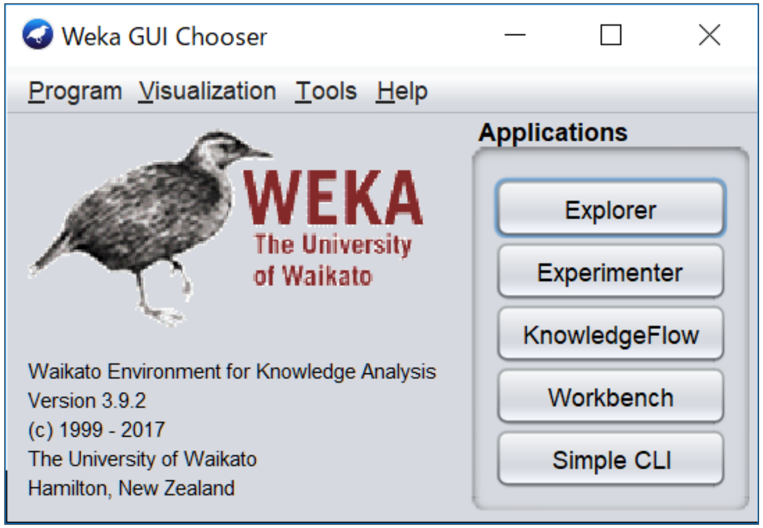
- 特徴ベクトルの次元を増やす



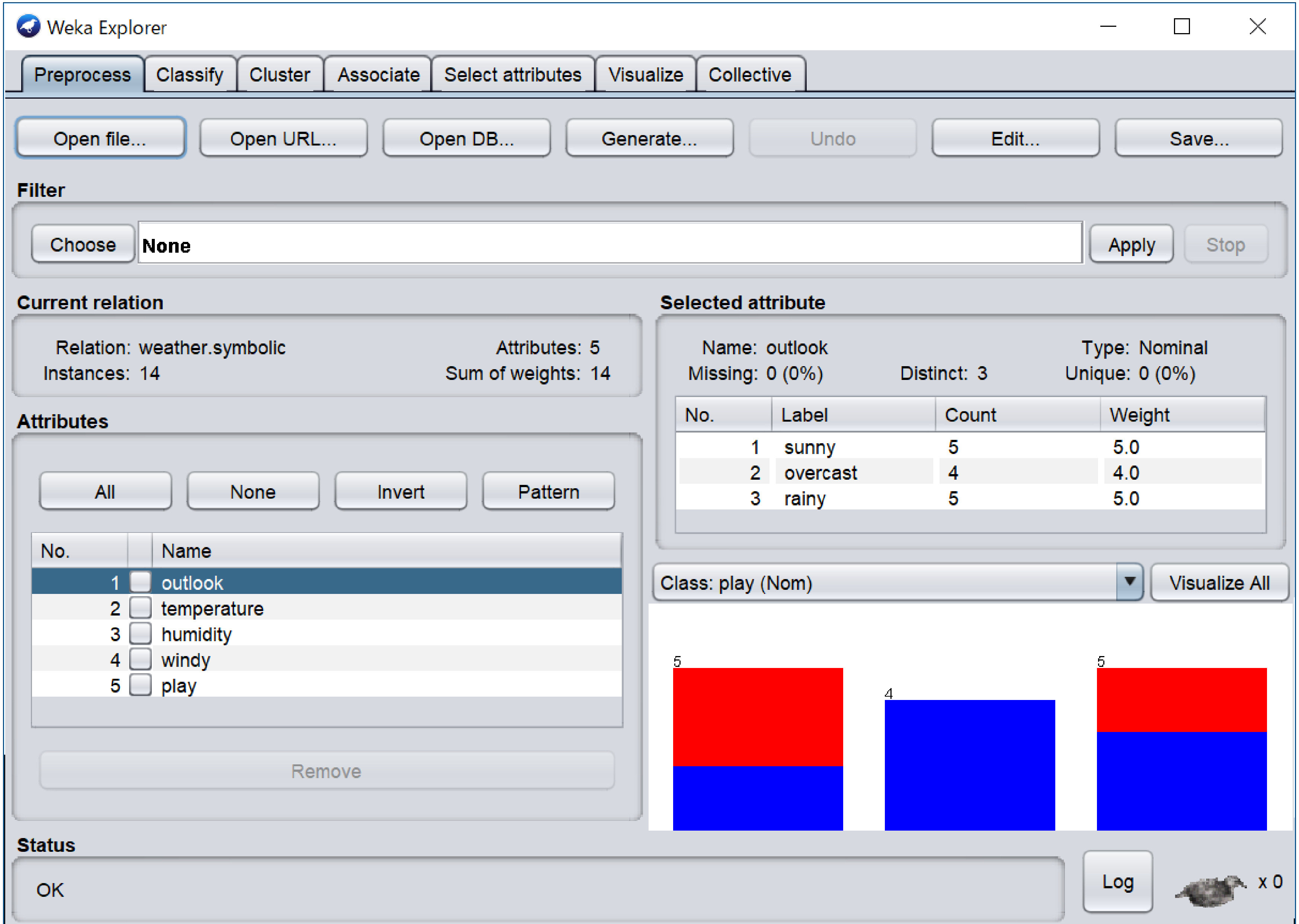
ただし、元の空間でのデータ間の
距離関係は保持するように

機械学習ライブラリの紹介 Weka

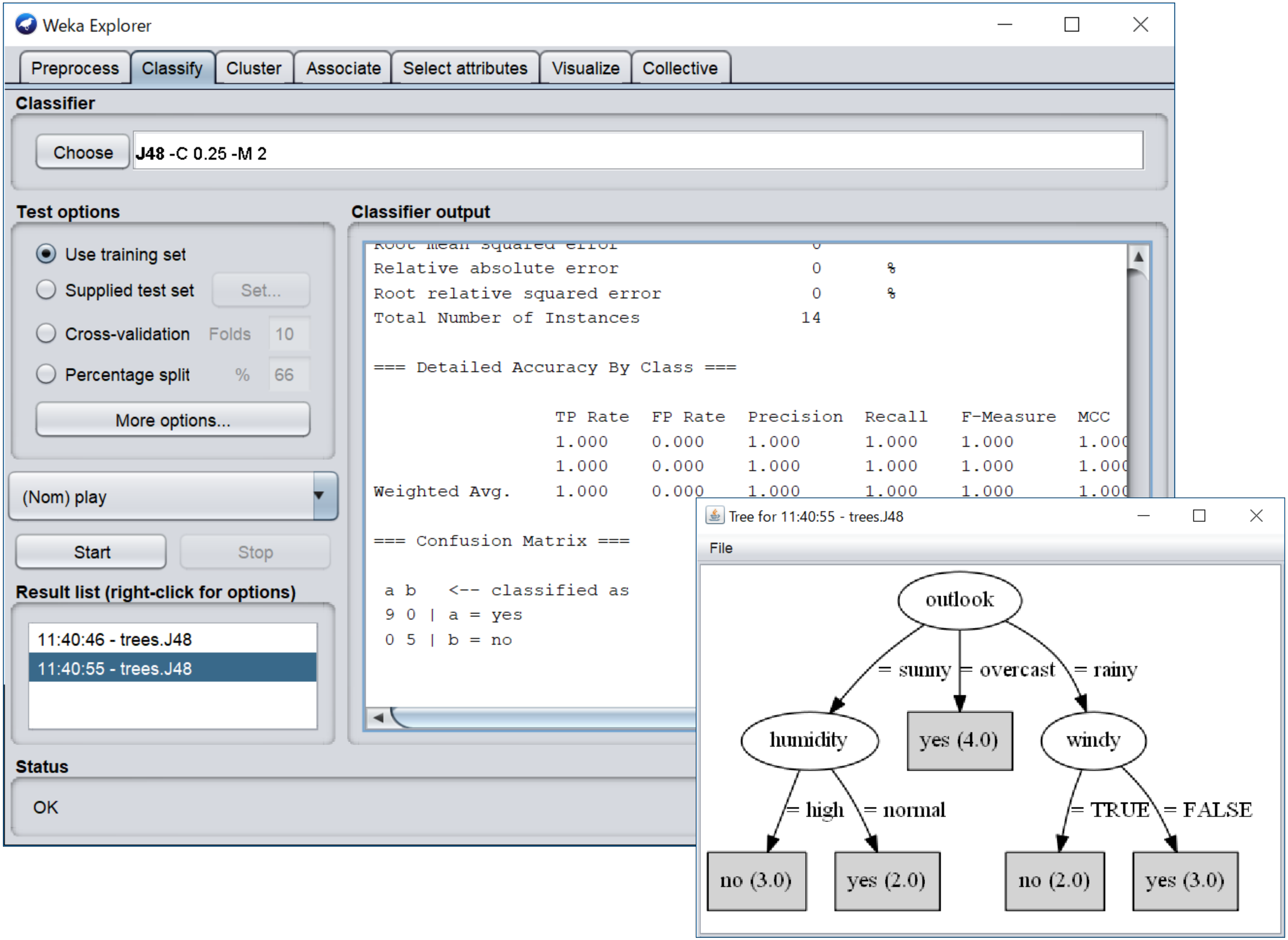
- 機械学習のアルゴリズムを実装した Java ライブラリ
- データファイルを直接操作できる GUI を持つ
- ライセンスは GNU GPL
  - プログラムの実行・改変・再配布が自由
  - ただし二次的著作物に対しても GNU GPL が適用される



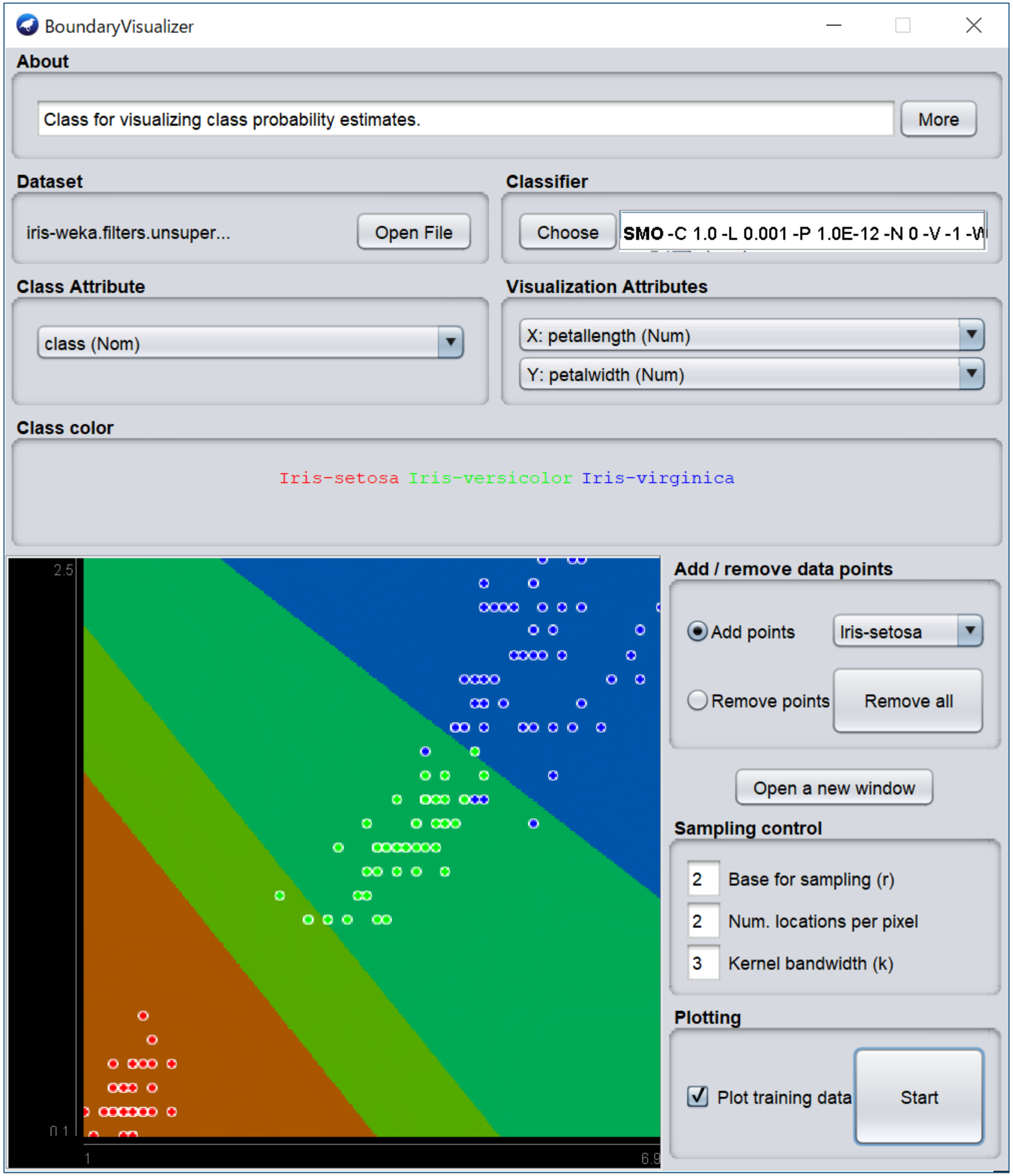
- データの読み込み


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Collective
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Stop
Current relation
Relation: weather.symbolic
Instances: 14
Attributes: 5
Sum of weights: 14
Attributes
All
None
Invert
Pattern
No.
Name
1 outlook
2 temperature
3 humidity
4 windy
5 play
Remove
Selected attribute
Name: outlook
Type: Nominal
Missing: 0 (0%)
Distinct: 3
Unique: 0 (0%)
No. Label Count Weight
1 sunny 5 5.0
2 overcast 4 4.0
3 rainy 5 5.0
Class: play (Nom)
Visualize All
5
4
5
Status
OK
Log
x 0

• 学習、結果の表示

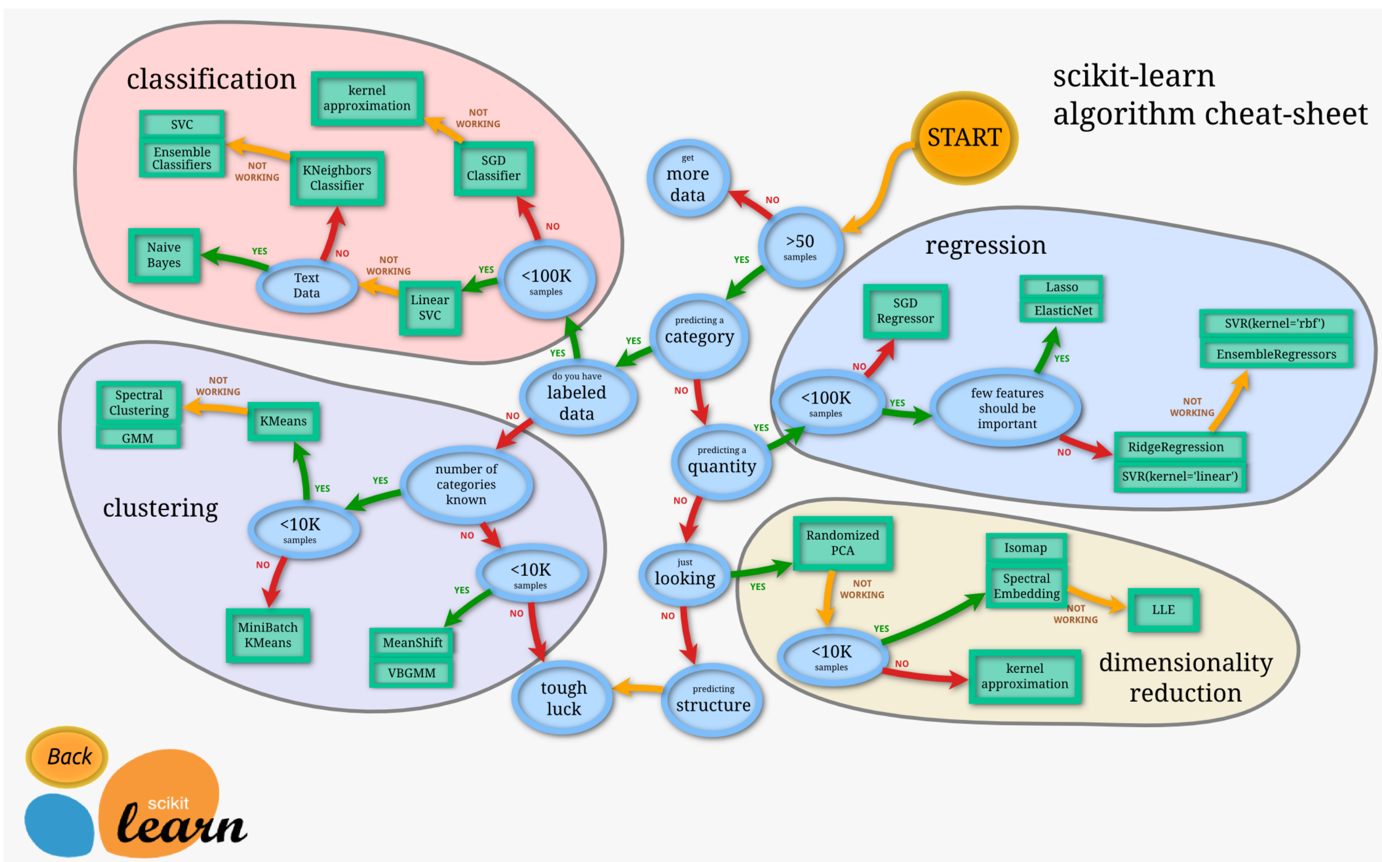

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Collective
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) play
Start
Stop
Result list (right-click for options)
11:40:46 - trees.J48
11:40:55 - trees.J48
Status
OK
Classifier output
Relative absolute error 0 %
Root relative squared error 0 %
Total Number of Instances 14
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC
1.000 0.000 1.000 1.000 1.000
1.000 0.000 1.000 1.000 1.000
Weighted Avg. 1.000 0.000 1.000 1.000 1.000
=== Confusion Matrix ===
a b <-- classified as
9 0 | a = yes
0 5 | b = no
Tree for 11:40:55 - trees.J48
File
outlook
= sunny
= overcast
= rainy
humidity
yes (4.0)
windy
= high
= normal
= TRUE
= FALSE
no (3.0)
yes (2.0)
no (2.0)
yes (3.0)

- 学習、結果の表示

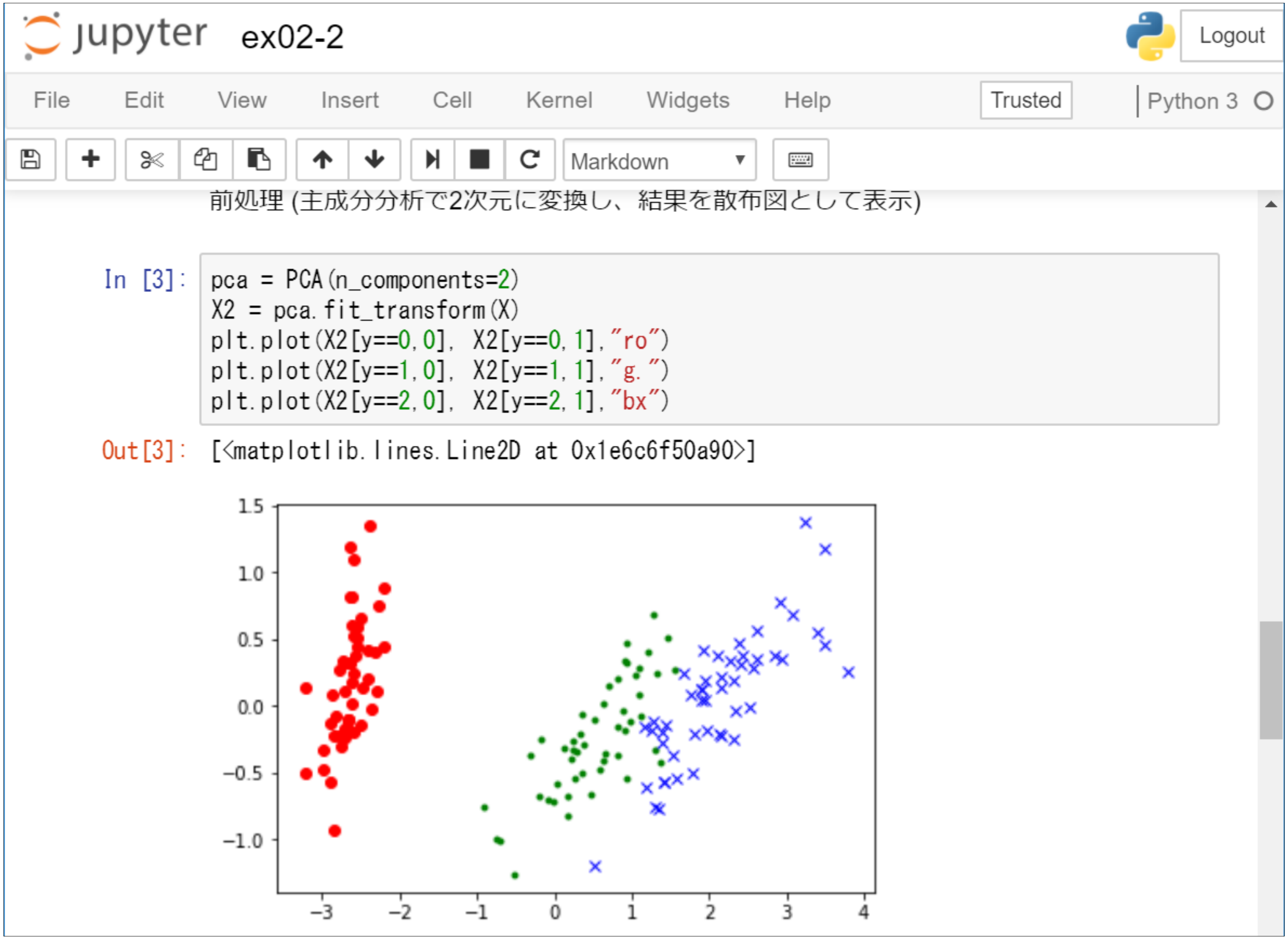

BoundaryVisualizer
About
Class for visualizing class probability estimates.
More
Dataset
iris-weka.filters.unsuper...
Open File
Classifier
Choose
SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W
Class Attribute
class (Nom)
Visualization Attributes
X: petallength (Num)
Y: petalwidth (Num)
Class color
Iris-setosa Iris-versicolor Iris-virginica
2.5
0.1
1
6.9
Add / remove data points
Add points
Iris-setosa
Remove points
Remove all
Open a new window
Sampling control
2 Base for sampling (r)
2 Num. locations per pixel
3 Kernel bandwidth (k)
Plotting
Plot training data
Start

機械学習ライブラリの紹介 Scikit-learn

- Python の機械学習ライブラリ
- 最新のアルゴリズムが実装されている
- Jyputer Notebook を使ったインタラクティブな開発が可能



- Jupyter Notebook
  - ブラウザで実行可能な Python 環境

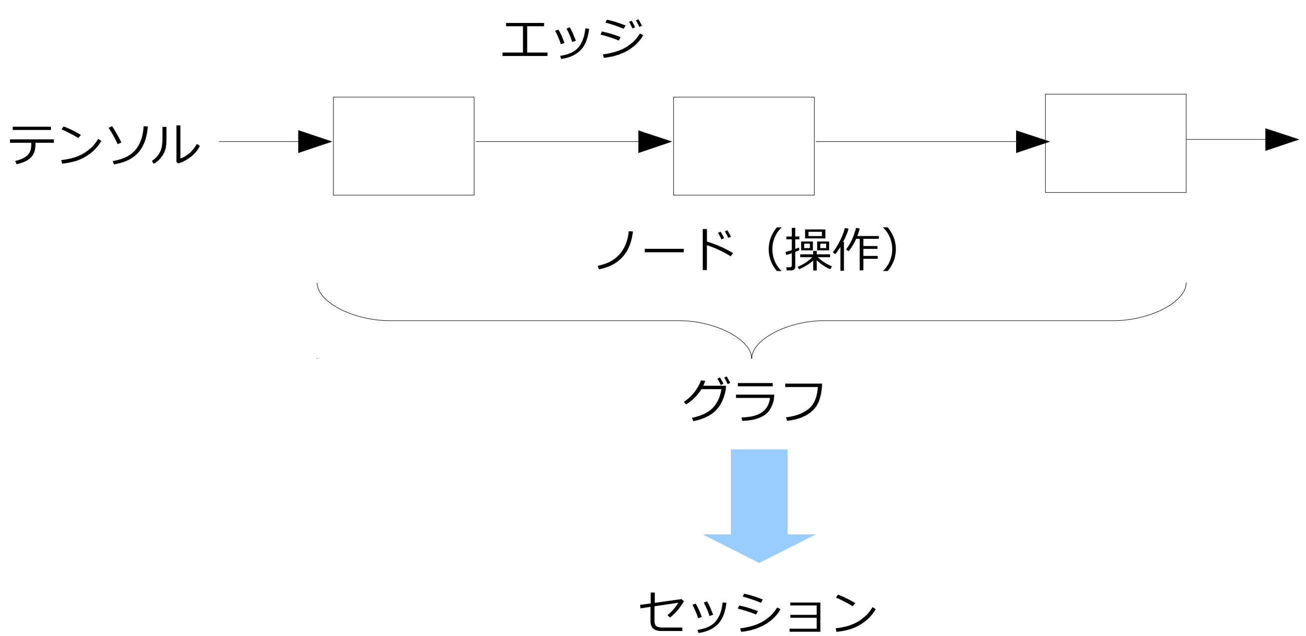


機械学習ライブラリの紹介 Tensorflow

- Google 社が開発した深層学習のライブラリ
- Python がベース
- ネットワークの定義が容易にできるラッパーライブラリ Keras が有用

Tensorflow の基本概念

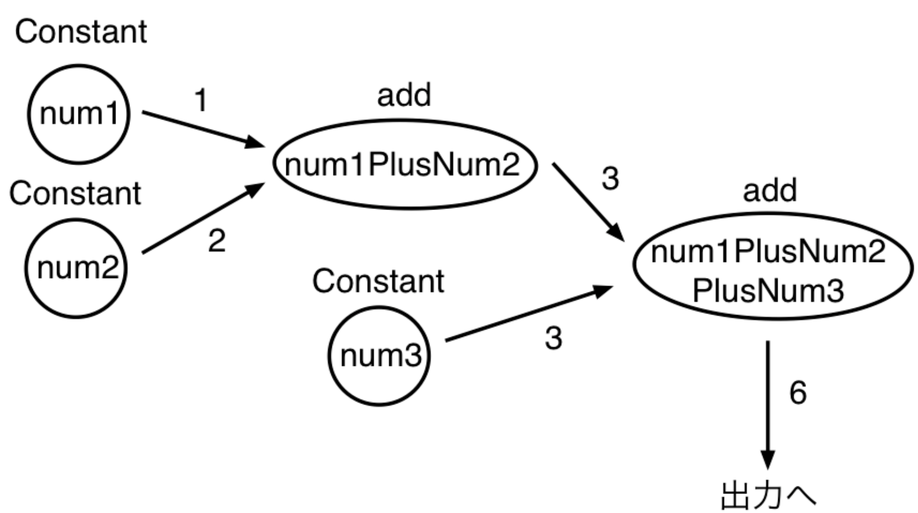
- テンソル
  - n 次元にデータを並べたもの
    - 1 次元：ベクトル
    - 2 次元：行列
- Tensorflow のデータモデル



- グラフによる計算の表現

```
import tensorflow as tf
num1 = tf.constant(1)
num2 = tf.constant(2)
num3 = tf.constant(3)
num1PlusNum2 = tf.add(num1,num2)
num1PlusNum2PlusNum3 = tf.add(num1PlusNum2,num3)
sess = tf.Session()
result = sess.run(num1PlusNum2PlusNum3)
print(result)
```

この出力に必要な計算は
自動的に行われる



Section2 のまとめ

- 統計的識別
  - 識別結果を確率付きで出力することができる
- 生成モデル：データの分布を示す関数を推定
- 識別モデル：データの境界を推定
  - 最急勾配法を用いて誤差最小のパラメータを求める
- 機械学習のライブラリ
  - Weka: GUI で操作できるので初学者の勉強に有用
  - Scikit-learn: 最も広く使われている Python のライブラリ
  - Tensorflow: 深層学習のライブラリ