

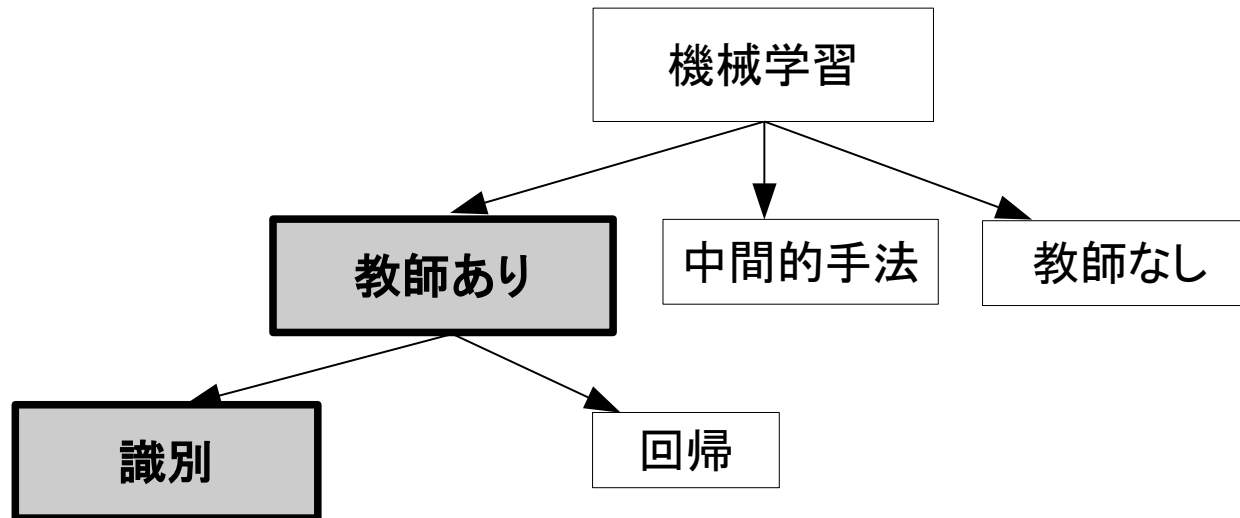
基本的な教師あり学習

- 決定木
- ナイーブベイズ
- ロジスティック識別

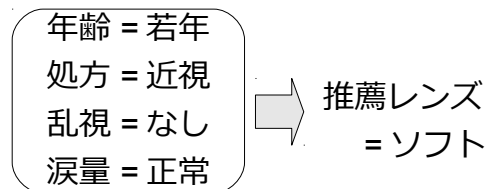
3. 識別 ―概念学習と決定木―

- 問題設定

- 教師あり学習
- ラベル入力 → ラベル出力



- ラベル特徴



- 数値特徴


3.2 概念学習とは

- 概念学習とは

- 正解の概念を説明する特徴ベクトルの性質（論理式）を求めること
- 論理式の例

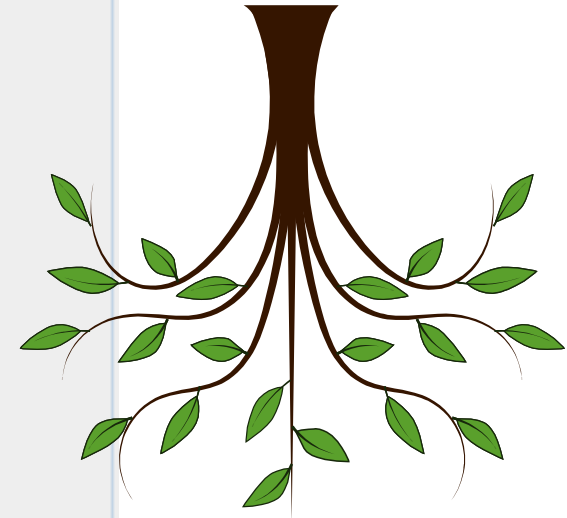
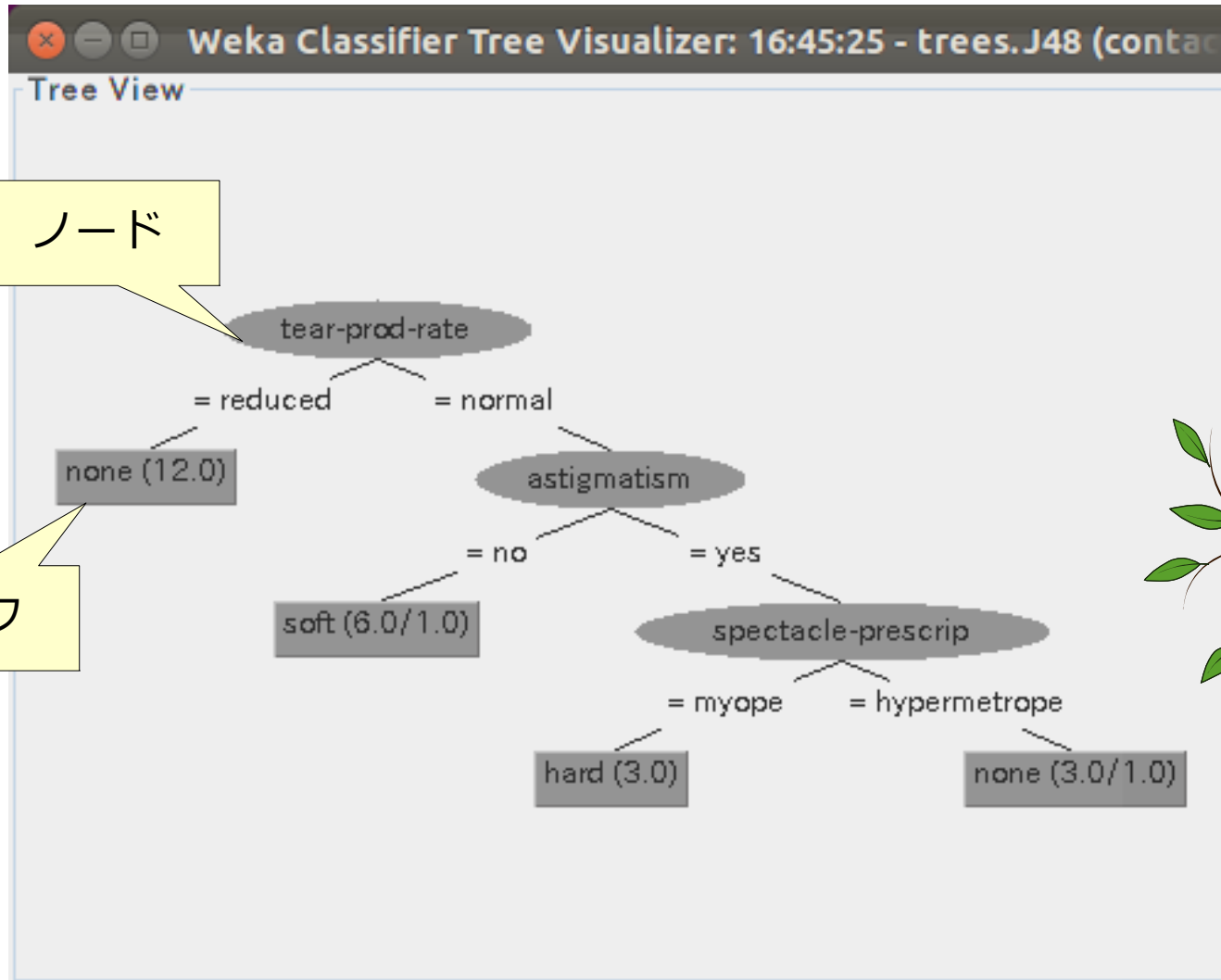
（乱視 = あり） \wedge （ドライアイ = なし） \Rightarrow soft

- 学習の方法

- 可能な論理式が少数
 - 正解概念の候補を絞り込んでゆく（候補削除アルゴリズム）
- 可能な論理式が多数
 - バイアス（偏見）をかけて探索する  決定木

3.4 決定木の学習

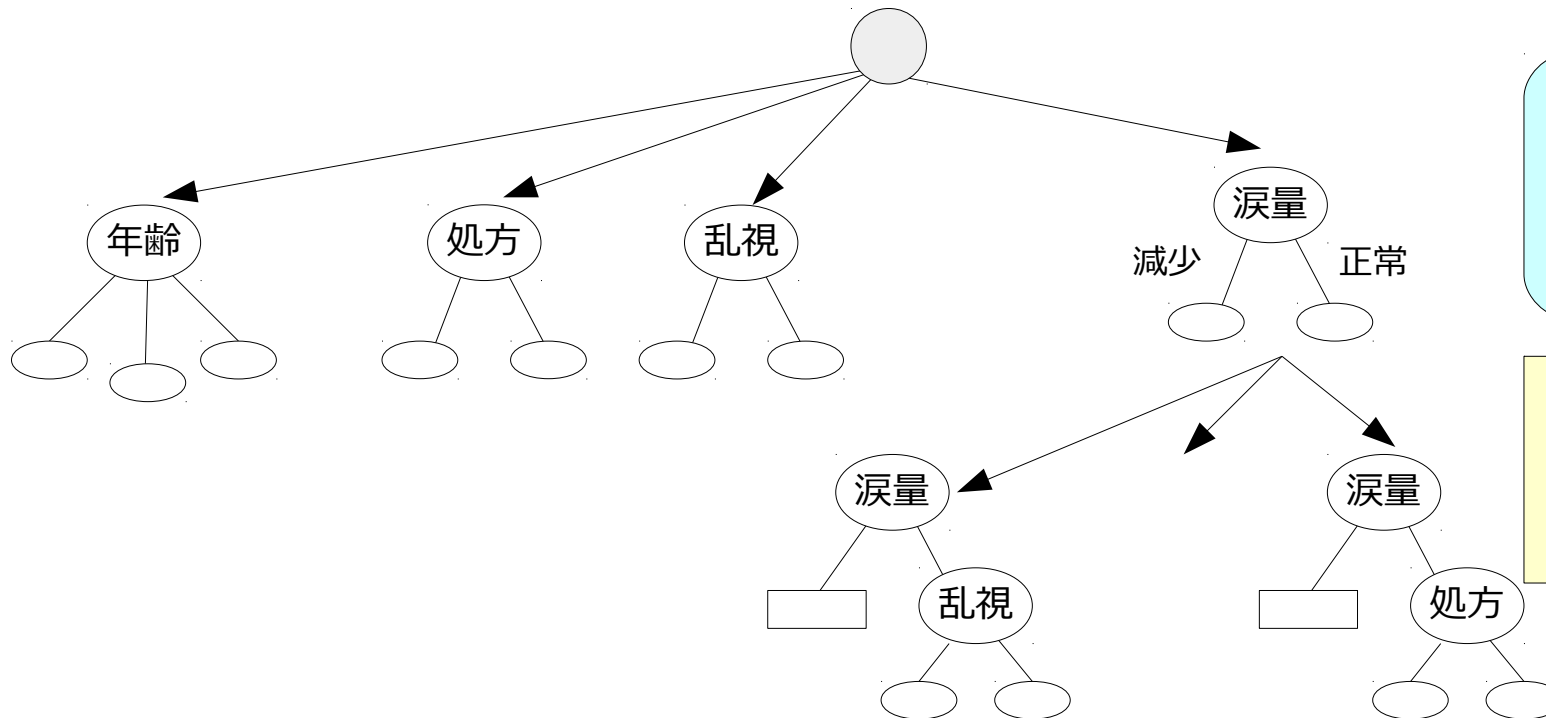
- 学習した決定木の例



3.4 決定木の学習

- 決定木学習の考え方

- ノードは、データを分割する条件を持つ
 - できるだけ同一クラスのデータがリーフに偏るように
- 分割後のデータ集合に対して、同様の操作を行う
- 全てのリーフが単一クラスの集合になれば終了



この手順に従うと、
一般には小さな木
ができる

バイアス

複雑な説明よりも
単純な説明の方が
汎用性が高い

属性の分類能力 (1/2)

- 分類能力の高い属性を決定する方法
 - その属性を使った分類を行うことによって、なるべくきれいにクラスが分かれるように
- エントロピー
 - データ集合 S の乱雑さを表現
 - 正例の割合 : p^+ , 負例の割合 : p^-
 - エントロピーの定義

$$Entropy(S) = -p^+ \log p^+ - p^- \log p^-$$

属性の分類能力 (2/2)

- 情報獲得量

- 属性 A を用いた分類後のエントロピーの減少量
- 値 v を取る訓練例の集合 : S_v
- S_v の要素数 : $|S_v|$
- 情報獲得量の定義

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Weka での決定木学習

決定木は J48

学習データを
評価に使う

右クリック→
Visualize tree
で木を表示

The screenshot shows the Weka Explorer window with the 'Classify' tab selected. The 'Classifier' dropdown is set to 'J48 -C 0.25 -M 2'. Under 'Test options', 'Use training set' is selected. The 'Result list' shows '12:44:23 - trees.J48' selected. The 'Classifier output' pane displays the following information:

```
=== Run information ===

Scheme:      weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:    contact-lenses
Instances:   24
Attributes:  5
              age
              spectacle-prescrip
              astigmatism
              tear-prod-rate
              contact-lenses

Test mode:   evaluate on training data

=== Classifier model (full training set) ===

J48 pruned tree
-----

tear-prod-rate = reduced: none (12.0)
tear-prod-rate = normal
|   astigmatism = no: soft (6.0/1.0)
|   astigmatism = yes
|       spectacle-prescrip = myope: hard (3.0)
|       spectacle-prescrip = hypermetrope: none (3.0/1.0)

Number of Leaves :    4
Size of the tree :    7

Time taken to build model: 0 seconds
```

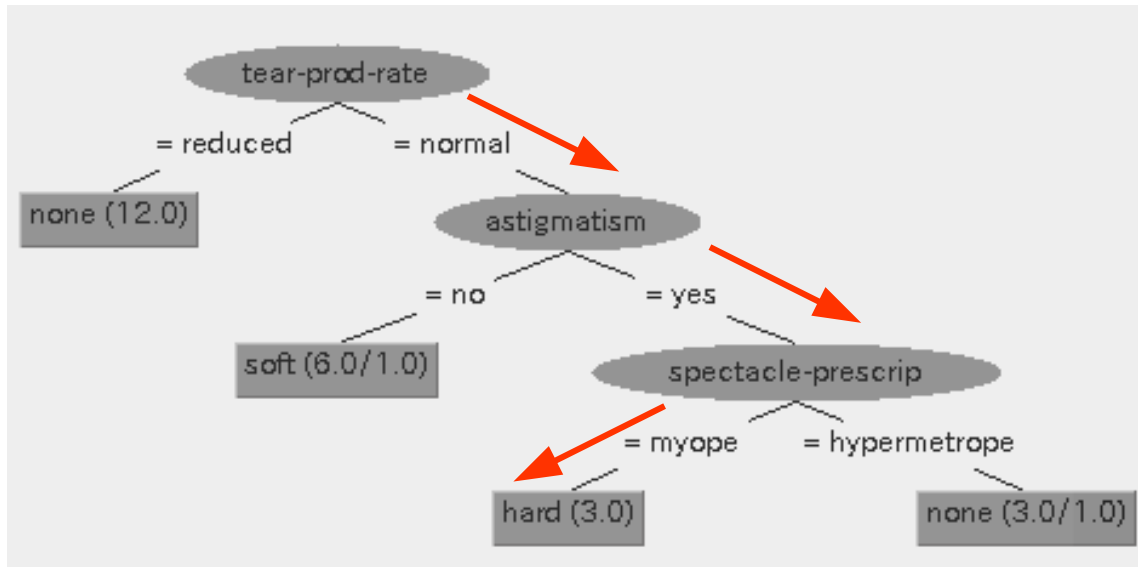
At the bottom, the status is 'OK' and there is a 'Log' button.

木のテキスト
表示

実行例

- 入力

年齢 処方 乱視 涙量
(young, myope, yes, normal)



- 出力

hard

バイアスの検討

なぜ単純な木の方がよいか

- オッカムの剃刀

「データに適合する最も単純な仮説を選べ」

- 複雑な仮説

- 表現能力が高い

- 偶然にデータを説明できるかもしれない

- 単純な仮説

- 表現能力が低い

- 偶然にデータを説明できる確率は低い

- でも説明できた！

- **必然**

連続値属性の扱い

- 連続値 A を持つ属性から真偽値 ($A < c?$) を値とするノードを作成

→ c をどうやって決めるか

気温	40	48	60	72	80	90
playTennis	No	No	Yes	Yes	Yes	No

$$c = (48 + 60) / 2 \\ = 54$$

$$c = (80 + 90) / 2 \\ = 85$$

情報獲得量の多い方

連続値属性の扱い

Weka Explorer

Preprocess | **Classify** | Cluster | Associate | Select attributes | Visualize

Classifier
Choose J48 -G 0.25 -M 2

Test options
☐ Use training set
☐ Supplied test set Set...
☒ Cross-validation Folds 10
☐ Percentage split % 66
More options...

(Nom) class

Start Stop

Result list (right-click for options)
16:44:11 - lazy.IBk
16:45:25 - trees.J48

Classifier output

```
Test mode: 10-fold cross-validation  
=== Classifier model (full training set) ===  
J48 pruned tree  
-----  
petalwidth <= 0.6: Iris-setosa (50.0)  
petalwidth > 0.6  
| petalwidth <= 1.7  
| | petallength <= 4.9: Iris-versicolor (48.0/1.0)  
| | petallength > 4.9  
| | | petalwidth <= 1.5: Iris-virginica (3.0)  
| | | petalwidth > 1.5: Iris-versicolor (3.0/1.0)  
| petalwidth > 1.7: Iris-virginica (46.0/1.0)  
leaves : 5  
tree : 9  
to build model: 0.01 seconds  
ried cross-validation ===  
===  
classified Instances 144 96 %  
Classified Instances 6 4 %  
stic 0.94  
te error 0.035  
quared error 0.1586  
olute error 7.8705 %
```

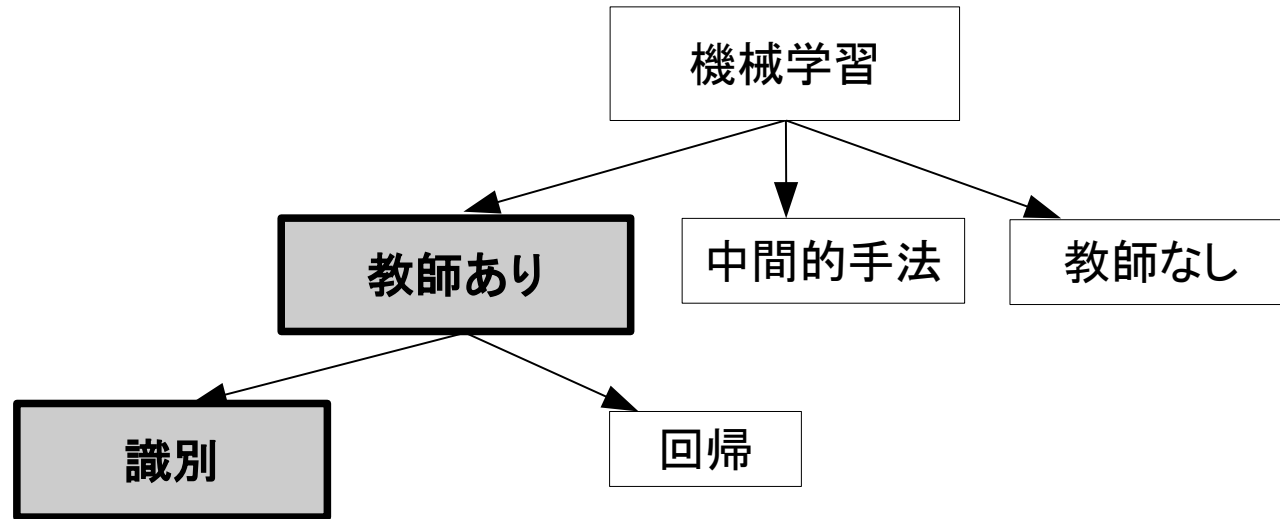
Weka Classifier Tree Visualizer: 16:46:52 - trees.J48 (iris)

Tree View

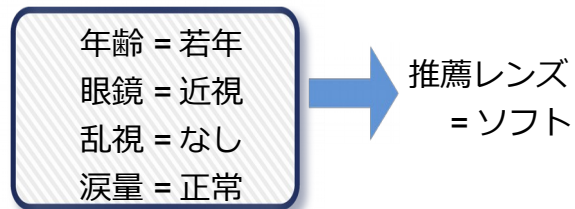
```
graph TD
    A(petalwidth) -- "<= 0.6" --> B[Iris-setosa (50.0)]
    A -- "> 0.6" --> C(petalwidth)
    C -- "<= 1.7" --> D(petallength)
    C -- "> 1.7" --> E[Iris-virginica (46.0/1.0)]
    D -- "<= 4.9" --> F[Iris-versicolor (48.0/1.0)]
    D -- "> 4.9" --> G(petalwidth)
    G -- "<= 1.5" --> H[Iris-virginica (3.0)]
    G -- "> 1.5" --> I[Iris-versicolor (3.0/1.0)]
```

Log x 0

4. 識別 —統計的手法—



- ラベル特徴



- 数値特徴

4.1 統計的識別とは

- 最大事後確率則による識別

\mathbf{x} : 特徴ベクトル

$$C_{MAP} = \arg \max_i P(\omega_i | \mathbf{x}) \quad \omega_i \ (1 \leq i \leq c) : \text{クラス}$$

- 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 $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

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

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 i のデータ数

4.1 統計的識別とは

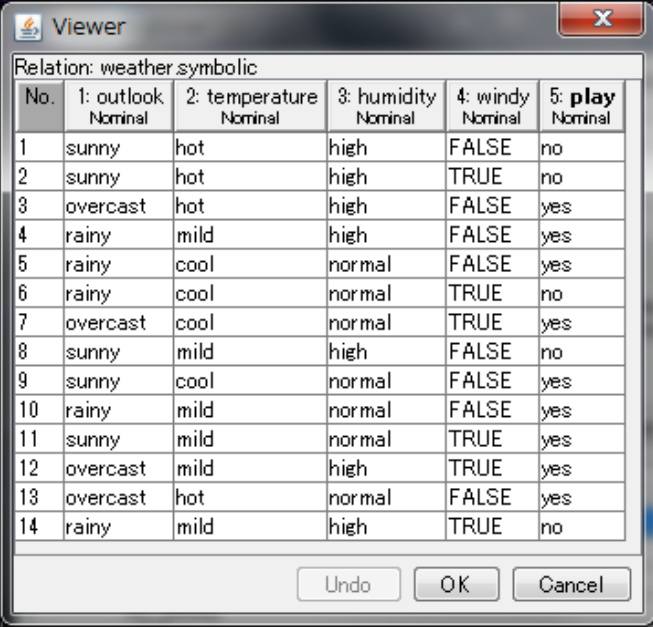
- 尤度

$$P(\boldsymbol{x}|\omega_i)$$

- 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の多数



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

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

$$P(\boldsymbol{x}|\omega_i) = P(x_1, \dots, x_d|\omega_i)$$

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

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

4.2.2 ナイーブベイス識別

- 尤度の最尤推定

$$P(x_j|\omega_i) = \frac{n_{ij}}{n_i}$$

n_{ij} : クラス i のデータのうち、
 j 次元目の値が x_j の個数

ゼロ頻度問題

- 確率の m 推定

$$P(x_j|\omega_i) = \frac{n_{ij} + mp}{n_i + m}$$

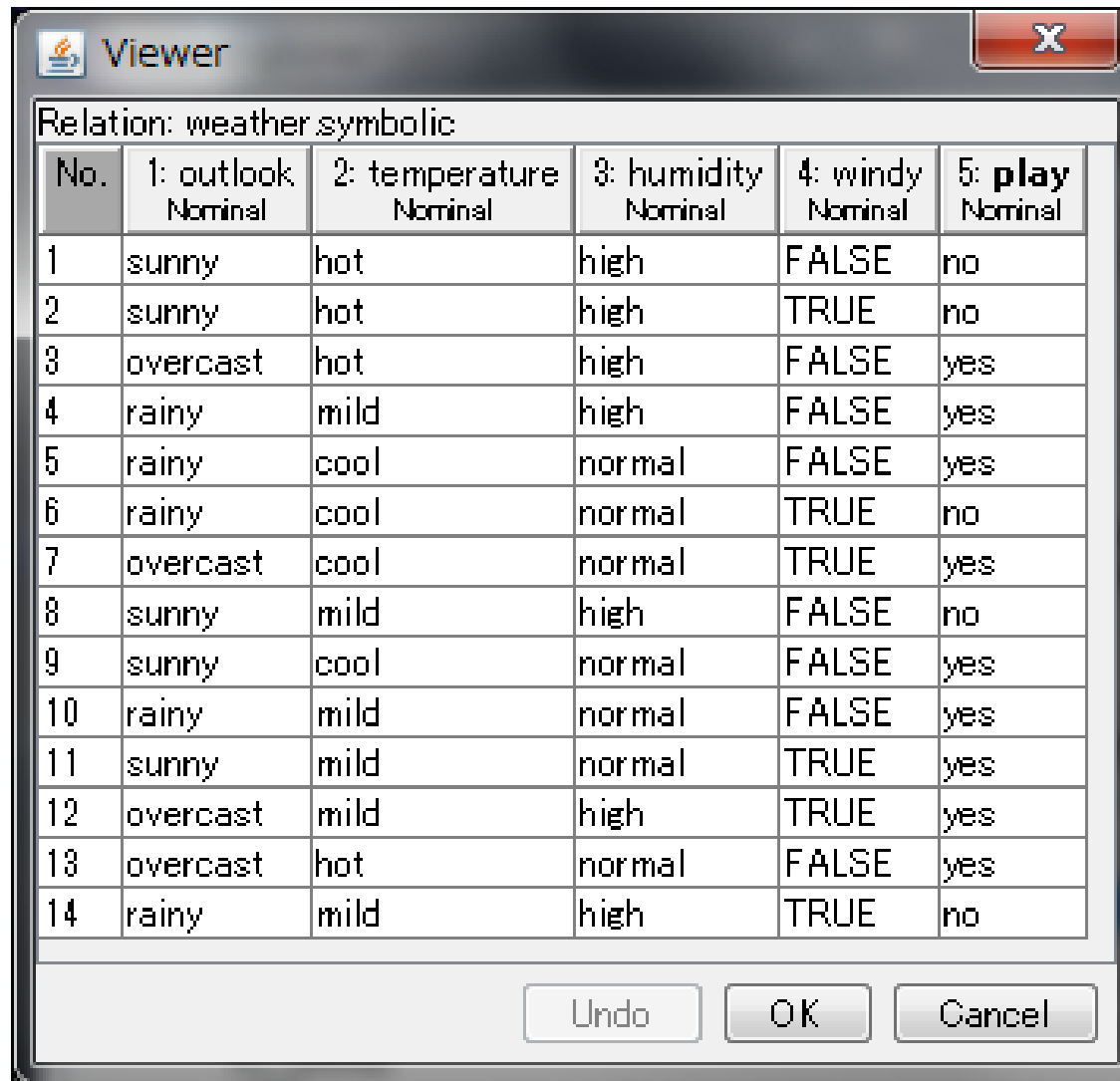
p : 事前に見積もった各特徴値の割合
 m : 事前に用意する標本数

- ラプラス推定

– m : 特徴値の種類数、 p : 等確率 とすると、 $mp=1$

4.2.2 ナイーブベイス識別

weather.nominal データ



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

4.2.2 ナイーブベイス識別

Weka Explorer

Preprocess | **Classify** | Cluster | Associate | Select attributes | Visualize

Classifier
Choose NaiveBayes

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)
12:44:23 - trees.J48
13:10:03 - bayes.NaiveBayes

Classifier output

Attribute	Class	
	yes (0.63)	no (0.38)
outlook		
sunny	3.0	4.0
overcast	5.0	1.0
rainy	4.0	3.0
[total]	12.0	8.0
temperature		
hot	3.0	3.0
mild	5.0	3.0
cool	4.0	2.0
[total]	12.0	8.0
humidity		
high	4.0	5.0
normal	7.0	2.0
[total]	11.0	7.0
windy		
TRUE	4.0	4.0
FALSE	7.0	3.0
[total]	11.0	7.0

Time taken to build model: 0 seconds
=== Evaluation on training set ===

Class

Attribute	yes (0.63)	no (0.38)
outlook		
sunny	3.0	4.0
overcast	5.0	1.0
rainy	4.0	3.0
[total]	12.0	8.0

ラプラス推定
各カウントの初期値が 1
例) no は 5 事例で、
overcast の事例は
データ中がない

Status
OK

Log x 0

実行例

- 入力

天気 気温 湿度 風
 $\mathbf{x}=(\text{rainy}, \text{hot}, \text{high}, \text{false})$

$$P(\text{yes}) = 0.63$$

$$P(\text{no}) = 0.38$$

$$P(\mathbf{x}|\text{yes}) = 4/12 \times 3/12 \times 4/11 \times 7/11 = 0.019$$

$$P(\mathbf{x}|\text{no}) = 3/8 \times 3/8 \times 5/7 \times 3/7 = 0.043$$

$$P(\mathbf{x}|\text{yes}) \cdot P(\text{yes}) = 0.012 < P(\mathbf{x}|\text{no}) \cdot P(\text{no}) = 0.016$$

- 出力

no

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

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

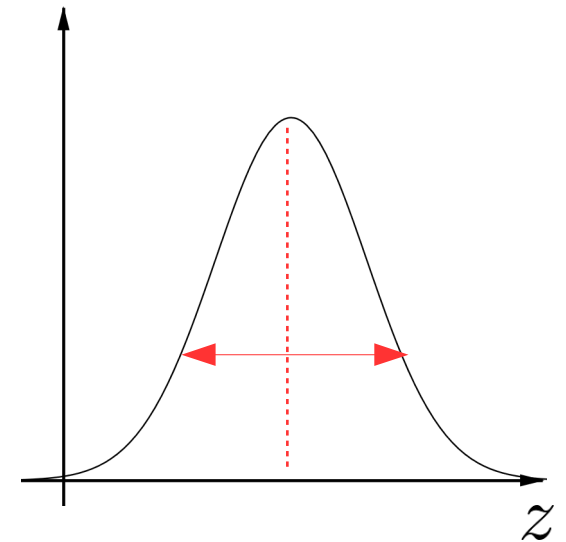
- 確率密度関数 $p(x_j | \omega_i)$ の推定

- 正規分布を仮定

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

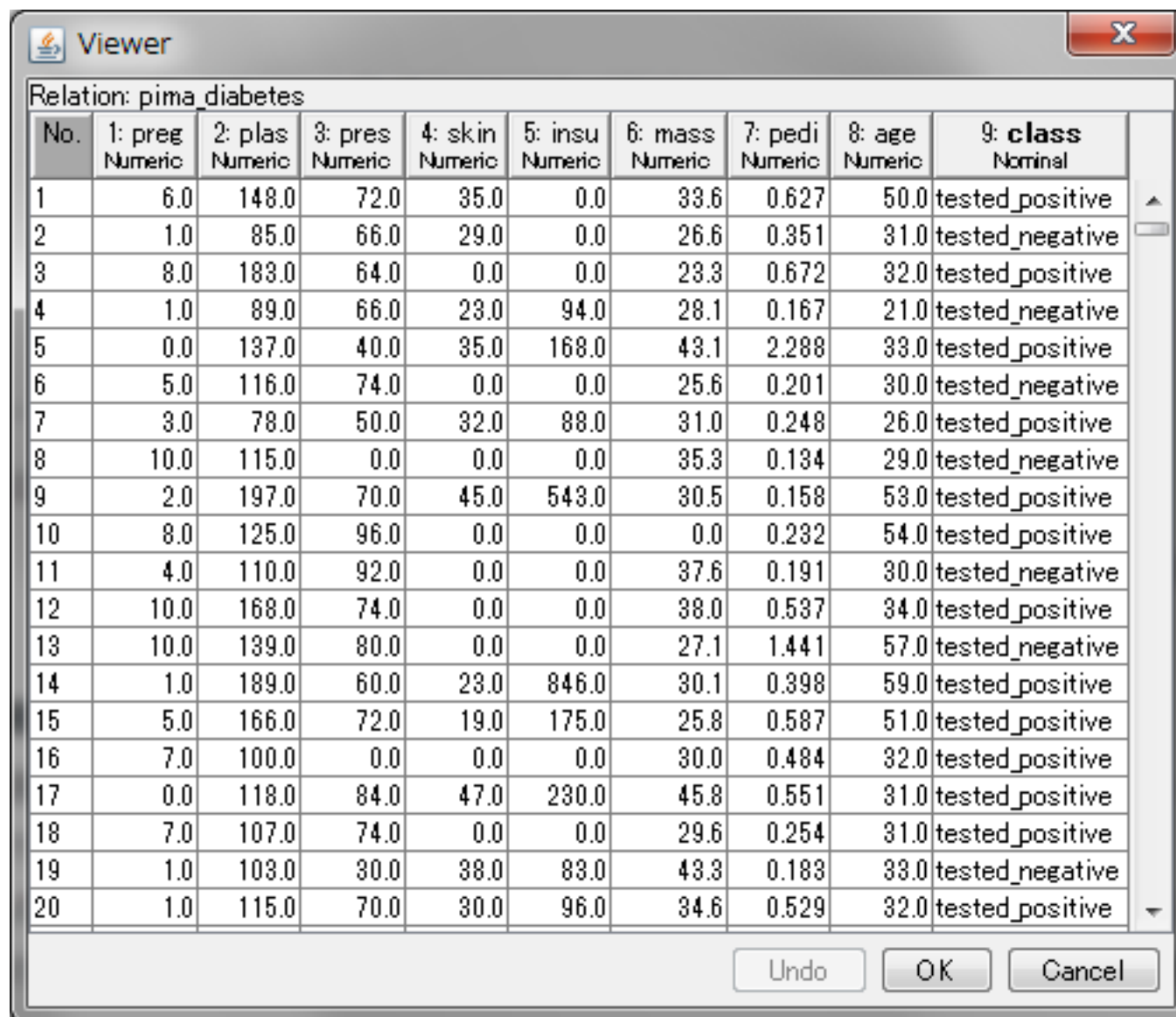
- 平均 μ と分散 σ を最尤推定

- それぞれ、学習データの平均と分散になる



5.2.1 数値特徴に対するナイーブベイズ識別

diabetes データ



Relation: pima diabetes

No.	1: preg Numeric	2: plas Numeric	3: pres Numeric	4: skin Numeric	5: insu Numeric	6: mass Numeric	7: pedi Numeric	8: age Numeric	9: class Nominal
1	6.0	148.0	72.0	35.0	0.0	33.6	0.627	50.0	tested_positive
2	1.0	85.0	66.0	29.0	0.0	26.6	0.351	31.0	tested_negative
3	8.0	183.0	64.0	0.0	0.0	23.3	0.672	32.0	tested_positive
4	1.0	89.0	66.0	23.0	94.0	28.1	0.167	21.0	tested_negative
5	0.0	137.0	40.0	35.0	168.0	43.1	2.288	33.0	tested_positive
6	5.0	116.0	74.0	0.0	0.0	25.6	0.201	30.0	tested_negative
7	3.0	78.0	50.0	32.0	88.0	31.0	0.248	26.0	tested_positive
8	10.0	115.0	0.0	0.0	0.0	35.3	0.134	29.0	tested_negative
9	2.0	197.0	70.0	45.0	543.0	30.5	0.158	53.0	tested_positive
10	8.0	125.0	96.0	0.0	0.0	0.0	0.232	54.0	tested_positive
11	4.0	110.0	92.0	0.0	0.0	37.6	0.191	30.0	tested_negative
12	10.0	168.0	74.0	0.0	0.0	38.0	0.537	34.0	tested_positive
13	10.0	139.0	80.0	0.0	0.0	27.1	1.441	57.0	tested_negative
14	1.0	189.0	60.0	23.0	846.0	30.1	0.398	59.0	tested_positive
15	5.0	166.0	72.0	19.0	175.0	25.8	0.587	51.0	tested_positive
16	7.0	100.0	0.0	0.0	0.0	30.0	0.484	32.0	tested_positive
17	0.0	118.0	84.0	47.0	230.0	45.8	0.551	31.0	tested_positive
18	7.0	107.0	74.0	0.0	0.0	29.6	0.254	31.0	tested_positive
19	1.0	103.0	30.0	38.0	83.0	43.3	0.183	33.0	tested_negative
20	1.0	115.0	70.0	30.0	96.0	34.6	0.529	32.0	tested_positive

Undo OK Cancel

5.2.1 数値特徴に対するナイーブベイズ識別

Weka Explorer

Preprocess | **Classify** | Cluster | Associate | Select attributes | Visualize

Classifier: Choose NaiveBayes

Test options:
☒ Use training set
☐ Supplied test set (Set...)
☐ Cross-validation (Folds: 10)
☐ Percentage split (%: 66)
More options...

(Nom) class: [v]
Start Stop

Result list (right-click for options):
12:44:23 - trees.J48
13:10:03 - bayes.NaiveBayes
13:17:28 - bayes.NaiveBayes

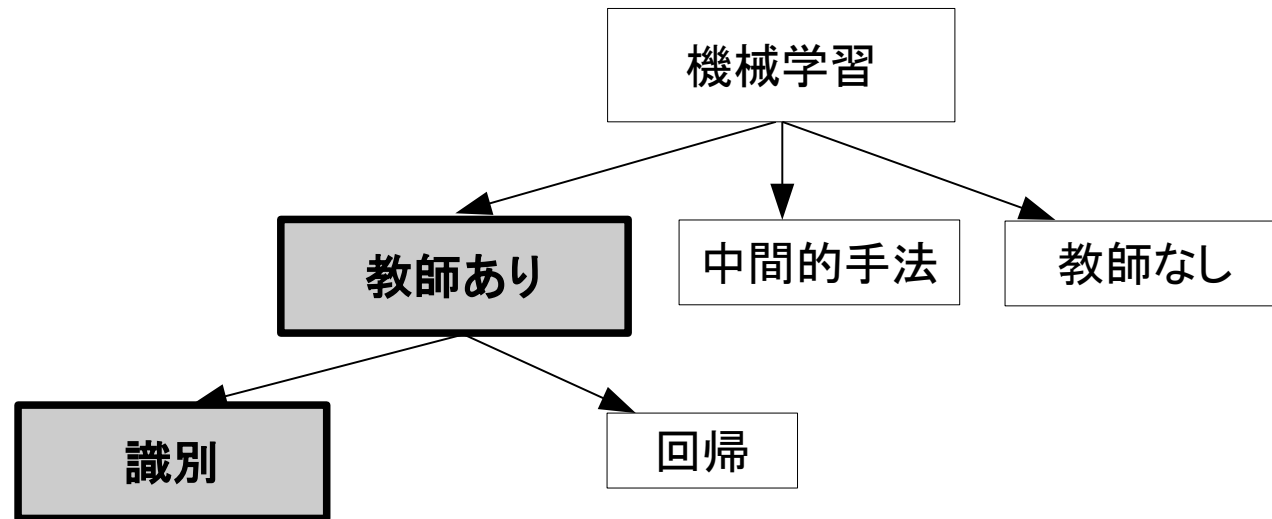
Status: OK

Classifier output:
=== Classifier model (full training set) ===
Naive Bayes Classifier

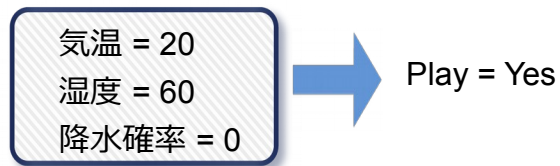
Attribute	Class	tested negative (0.65)	tested positive (0.35)
preg			
mean		3.4234	4.9795
std. dev.		3.0166	3.6827
weight sum		500	268
precision		1.0625	1.0625

=====
Attribute negative positive
=====
preg
mean 3.4234 4.9795
std. dev. 3.0166 3.6827
weight sum 500 268
precision 1.0625 1.0625

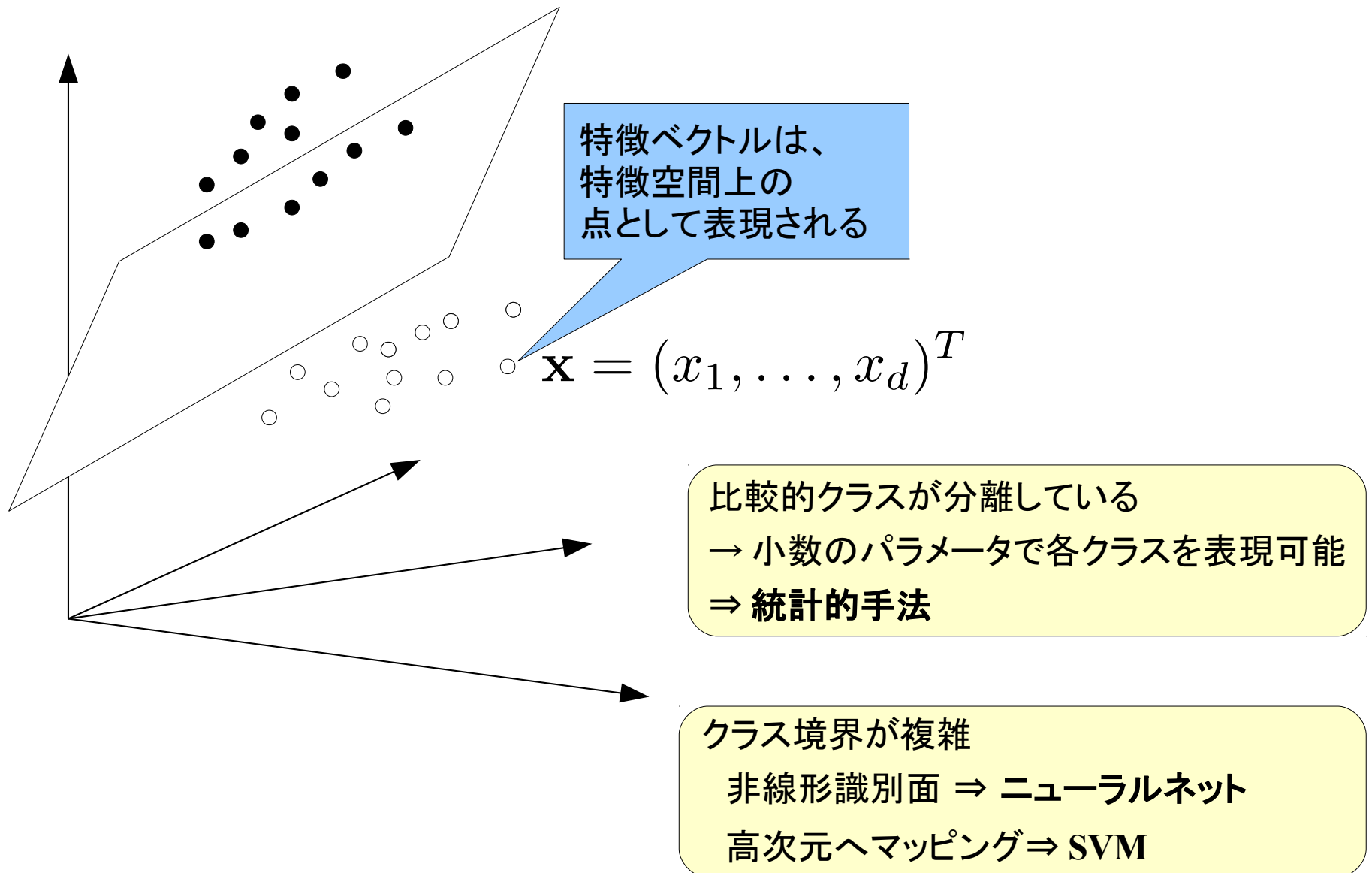
5. 識別 — 生成モデルと識別モデル—



- ラベル特徴
- 数値特徴



5.1 数値特徴に対する「教師あり・識別」問題の定義



5.2.2 生成モデルの考え方

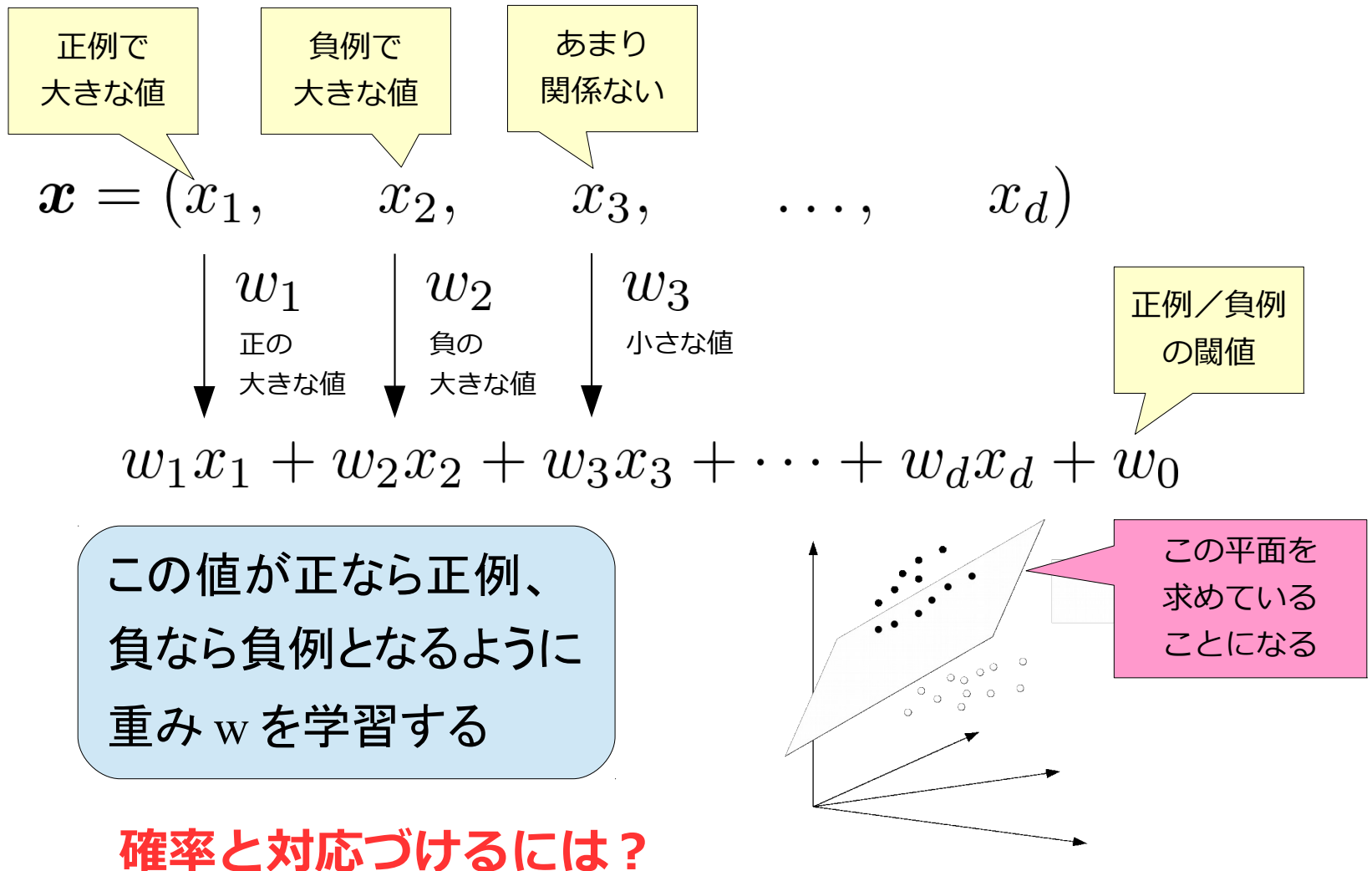
- 事後確率を求めるにあたって、同時確率を求めている
 - データが生成される様子をモデル化しているとも見ることが出来る
 - 事前確率に基づいてクラスを選ぶ
 - そのもとで、特徴ベクトルを出力する

$$\begin{aligned} P(\omega_i | \mathbf{x}) &= \frac{p(\mathbf{x} | \omega_i) P(\omega_i)}{p(\mathbf{x})} \\ &= \frac{p(\omega_i, \mathbf{x})}{p(\mathbf{x})} \end{aligned}$$

事後確率を求めるより、
難しい問題を解いている
のではないかな？

5.3.1 識別モデルの考え方

- 事後確率を直接求める

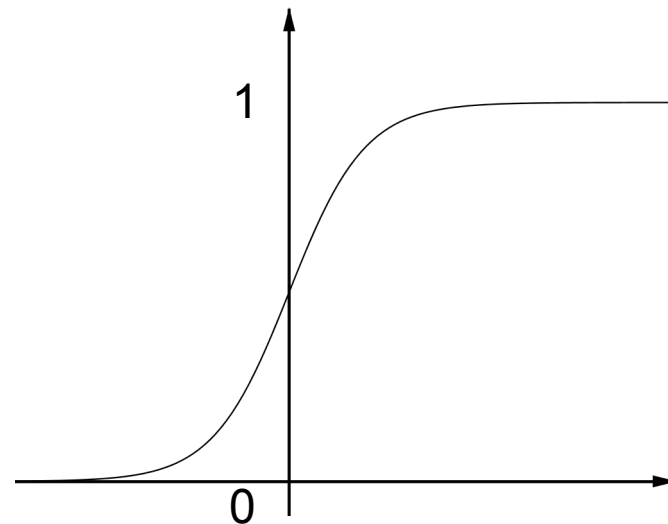


5.3.1 識別モデルの考え方

- ロジスティック識別
 - 入力が正例である確率

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

$-\infty \sim +\infty$ の値域を持つ
ものを、順序を変えずに
 $0 \sim 1$ にマッピング



シグモイド関数

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

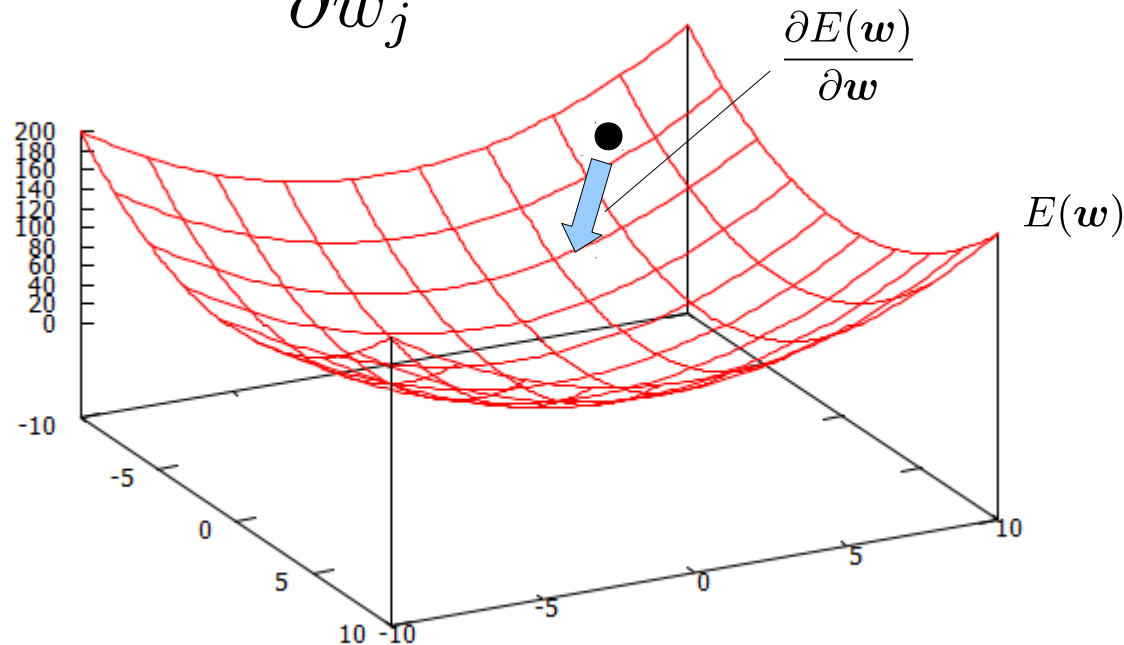
$$E(\mathbf{w}) = -\log P(D|\mathbf{w}) = -\log \prod_{\mathbf{x}_i \in D} o_i^{y_i} (1 - o_i)^{(1-y_i)}$$

- $E(\mathbf{w})$ を最急勾配法で最小化

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

$$o = P(\oplus | \mathbf{x})$$
$$y = o \text{ or } 1$$

正解ラベル



5.3.2 ロジスティック識別器の学習

- 重み更新量の計算

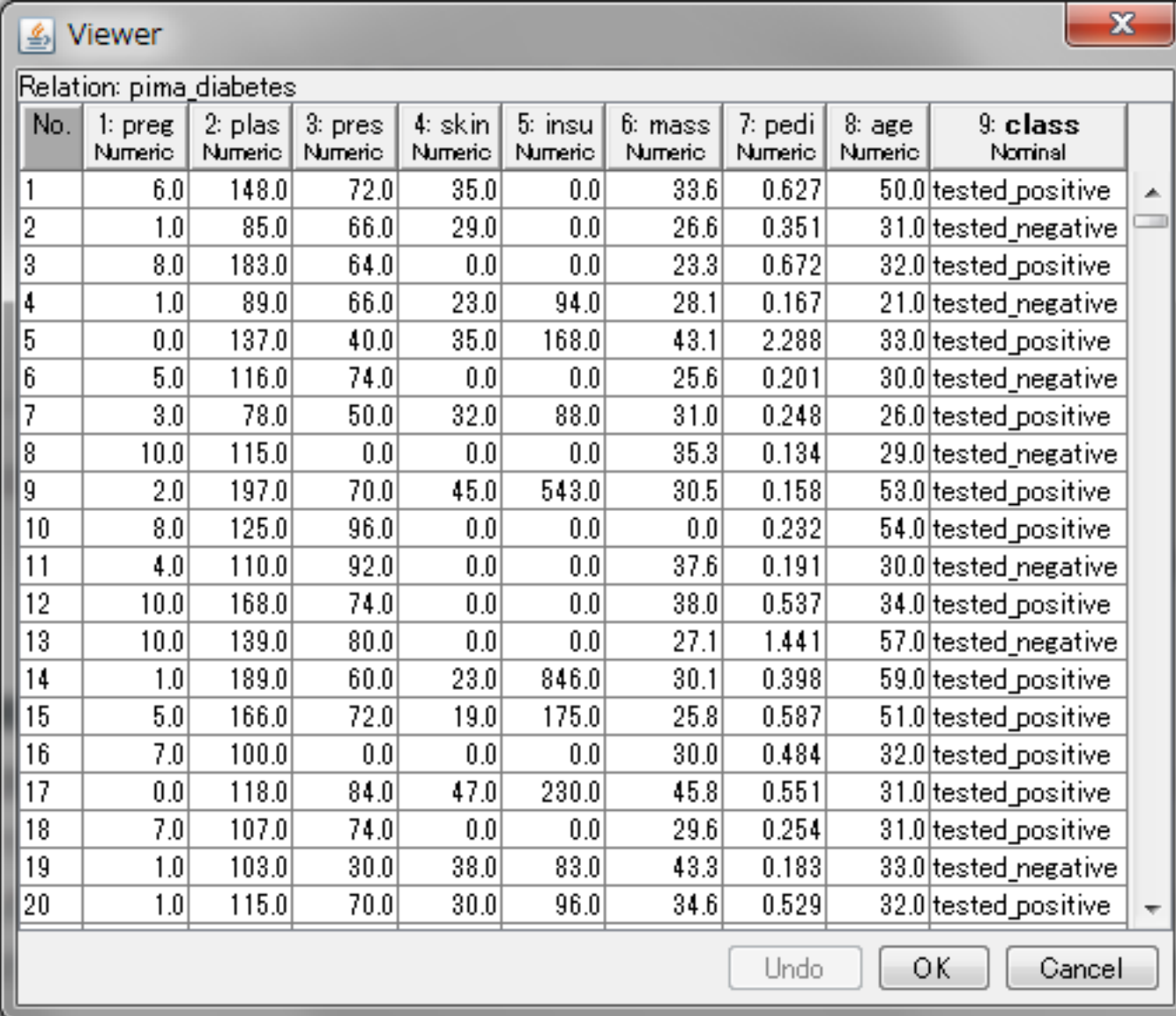
$$\begin{aligned}\frac{\partial E(\boldsymbol{w})}{\partial w_j} &= \sum_{\boldsymbol{x}_i \in D} \left(\frac{y_i}{o_i} - \frac{1 - y_i}{1 - o_i} \right) o_i (1 - o_i) x_{ij} \\ &= \sum_{\boldsymbol{x}_i \in D} (y_i - o_i) x_{ij}\end{aligned}$$

- 重みの更新式

$$w_j \leftarrow w_j - \eta \sum_{\boldsymbol{x}_i \in D} (y_i - o_i) x_{ij}$$

5.3.2 ロジスティック識別器の学習

diabetes データ



Relation: pima_diabetes

No.	1: preg Numeric	2: plas Numeric	3: pres Numeric	4: skin Numeric	5: insu Numeric	6: mass Numeric	7: pedi Numeric	8: age Numeric	9: class Nominal
1	6.0	148.0	72.0	35.0	0.0	33.6	0.627	50.0	tested_positive
2	1.0	85.0	66.0	29.0	0.0	26.6	0.351	31.0	tested_negative
3	8.0	183.0	64.0	0.0	0.0	23.3	0.672	32.0	tested_positive
4	1.0	89.0	66.0	23.0	94.0	28.1	0.167	21.0	tested_negative
5	0.0	137.0	40.0	35.0	168.0	43.1	2.288	33.0	tested_positive
6	5.0	116.0	74.0	0.0	0.0	25.6	0.201	30.0	tested_negative
7	3.0	78.0	50.0	32.0	88.0	31.0	0.248	26.0	tested_positive
8	10.0	115.0	0.0	0.0	0.0	35.3	0.134	29.0	tested_negative
9	2.0	197.0	70.0	45.0	543.0	30.5	0.158	53.0	tested_positive
10	8.0	125.0	96.0	0.0	0.0	0.0	0.232	54.0	tested_positive
11	4.0	110.0	92.0	0.0	0.0	37.6	0.191	30.0	tested_negative
12	10.0	168.0	74.0	0.0	0.0	38.0	0.537	34.0	tested_positive
13	10.0	139.0	80.0	0.0	0.0	27.1	1.441	57.0	tested_negative
14	1.0	189.0	60.0	23.0	846.0	30.1	0.398	59.0	tested_positive
15	5.0	166.0	72.0	19.0	175.0	25.8	0.587	51.0	tested_positive
16	7.0	100.0	0.0	0.0	0.0	30.0	0.484	32.0	tested_positive
17	0.0	118.0	84.0	47.0	230.0	45.8	0.551	31.0	tested_positive
18	7.0	107.0	74.0	0.0	0.0	29.6	0.254	31.0	tested_positive
19	1.0	103.0	30.0	38.0	83.0	43.3	0.183	33.0	tested_negative
20	1.0	115.0	70.0	30.0	96.0	34.6	0.529	32.0	tested_positive

Undo OK Cancel

5.3.2 ロジスティック識別器の学習

The screenshot shows the Weka Explorer application window. The 'Classify' tab is selected. The classifier chosen is 'SimpleLogistic -I 0 -M 500 -H 50 -W 0.0'. The 'Test options' section has 'Cross-validation' selected with 'Folds' set to 10. The 'Classifier output' pane shows the results of a 10-fold cross-validation. A red box highlights the coefficients for Class 0, which are then shown in a larger light blue box on the right. The status bar at the bottom indicates 'OK'.

Classifier output

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

SimpleLogistic:

Class 0 :

4.18 +

[preg] * -0.06 +

[plas] * -0.02 +

[pres] * 0.01 +

[insu] * 0 +

[mass] * -0.04 +

[pedi] * -0.47 +

[age] * -0.01

Class 1 :

-4.18 +

[preg] * 0.06 +

[plas] * 0.02 +

[pres] * -0.01 +

[insu] * 0 +

[mass] * 0.04 +

[pedi] * 0.47 +

[age] * 0.01

Time taken to build model: 0.11 seconds

=== Stratified cross-validation ===

=== Summary ===

Result list (right-click for options)

13:22:41 - functions.SimpleLogistic

Status: OK

Log x 0

実行例

- 入力

妊娠 血糖値 血圧 ...
 $\mathbf{x}=(6, 148, 72, 35, 0, 33.6, 0.627, 50)$

$$\begin{aligned} g(\mathbf{x}) &= 6 \times 0.06 + 148 \times 0.02 - 72 \times 0.01 \\ &\quad + 33.6 \times 0.04 + 0.627 \times 0.47 + 50 \times 0.01 \\ &\quad - 4.18 \\ &= 0.559 \end{aligned}$$

$$P(\text{tested_positive}) = 1/(1 + \exp(-g(\mathbf{x}))) = 0.636$$

- 出力

tested_positive