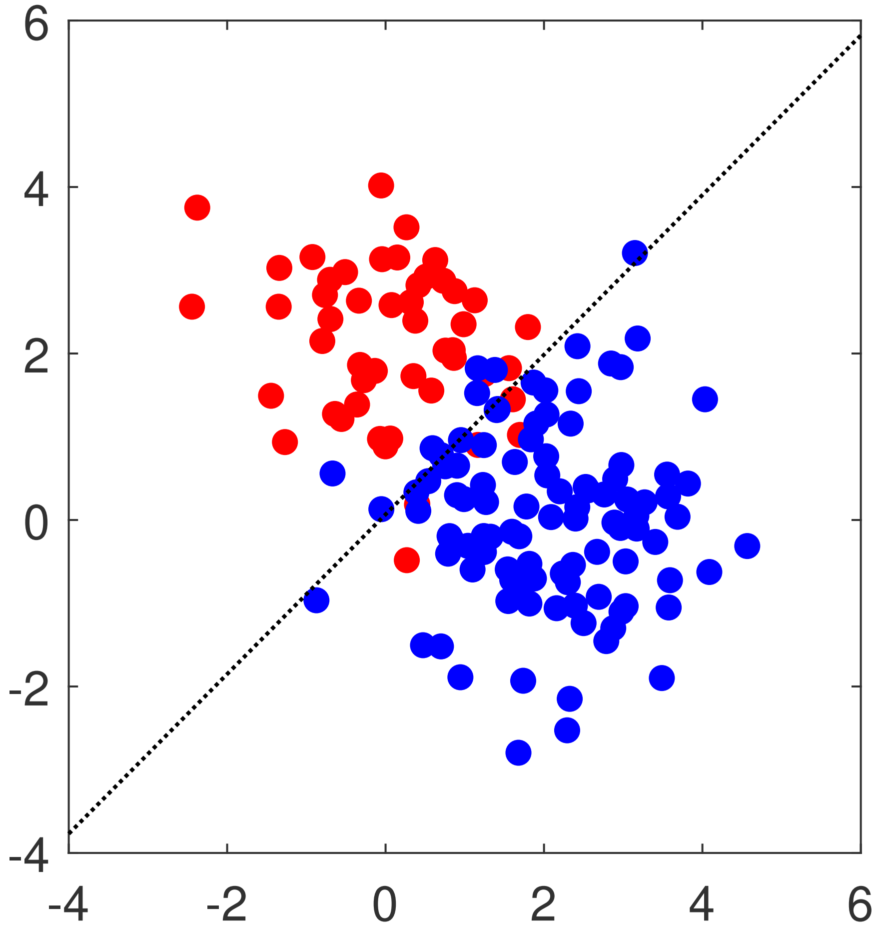
Discriminance by Class Centers

discriminance hyperplane
for 2 classes:

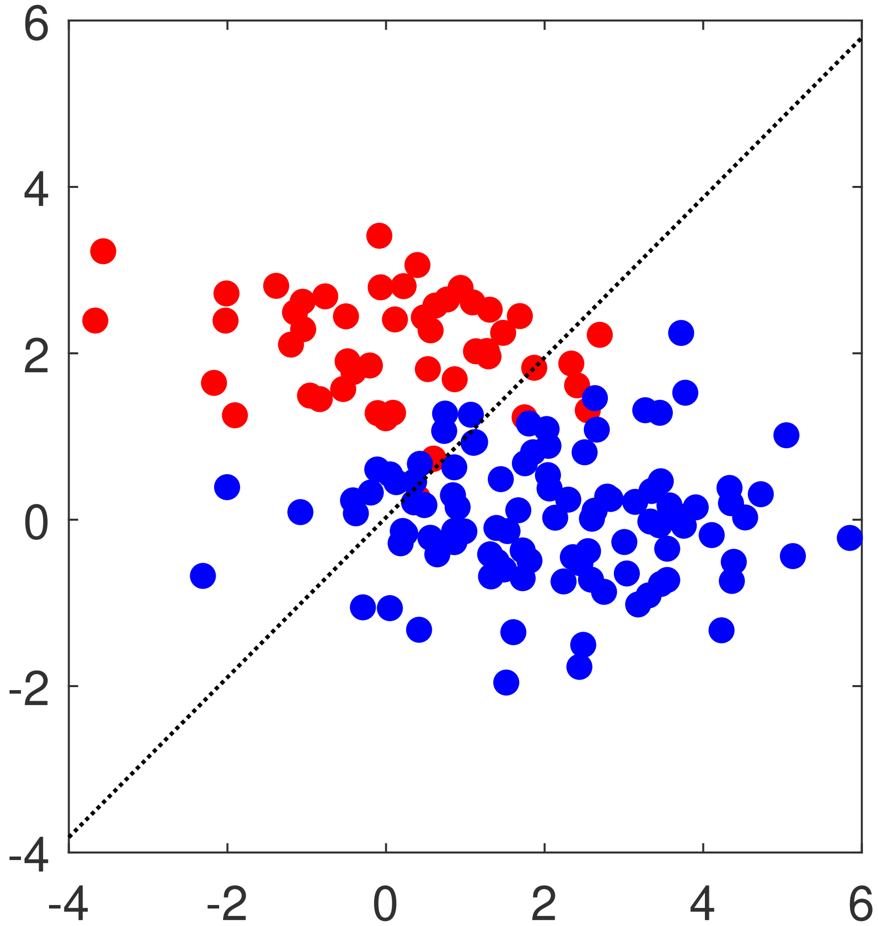
$$
\begin{aligned}
w \cdot x^T + b &= 0 \\
w &= \mu_1 - \mu_2 \\
b &= -w \cdot \frac{\mu_1^T + \mu_2^T}{2}
\end{aligned}
$$



Prof. Dr. Thomas A. Runkler

Discriminance by Class Centers

bad classification when covariances are different



Prof. Dr. Thomas A. Runkler

Linear Discriminant Analysis

- goal: good class separation
- within class covariance should be small

$$v_w = \sum_{i=1}^{c} \sum_{y_k=i} (x_k - \mu_i)^T (x_k - \mu_i)$$

- covariance between classes should be high

$$v_b = \sum_{i=1}^{c} (\mu_i - \bar{x})^T (\mu_i - \bar{x})$$

- special case 2 classes

$$v_b = (\mu_1 - \mu_2)^T (\mu_1 - \mu_2)$$

- maximize Fisher's quotient

$$J = \frac{w^T \cdot v_b \cdot w}{w^T \cdot v_w \cdot w}$$

Prof. Dr. Thomas A. Runkler

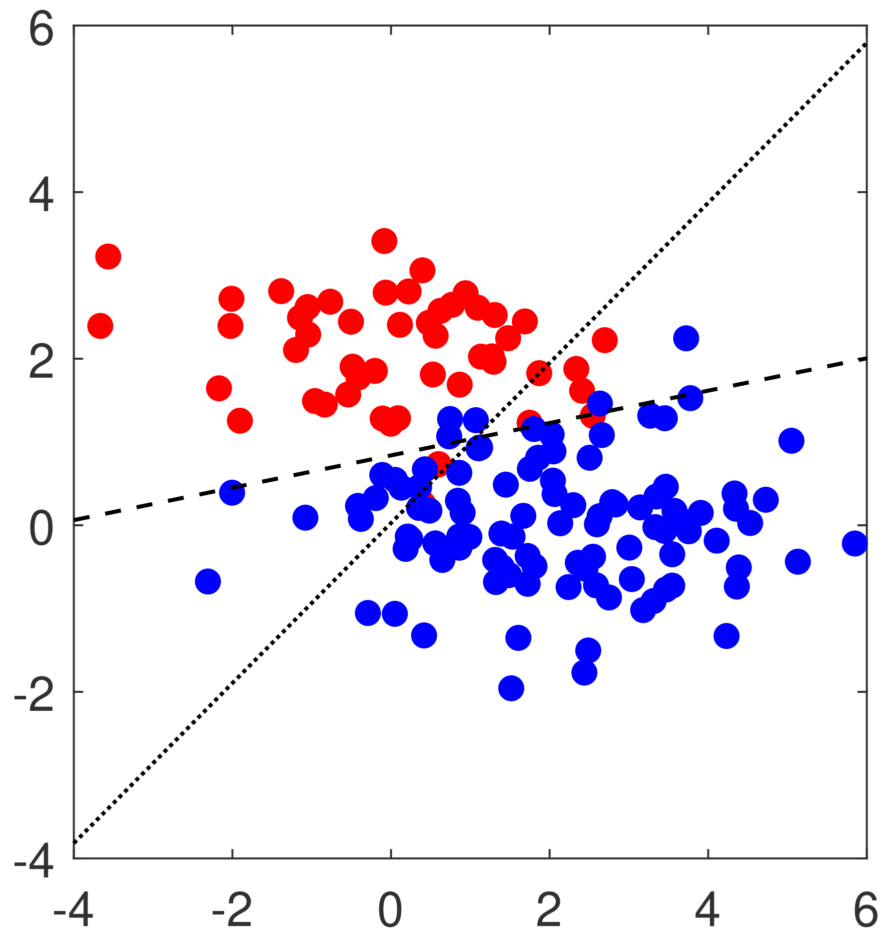
Linear Discriminant Analysis

max $J \Rightarrow$ eigen problem

$$(v_b^{-1} v_w) \cdot w = \lambda \cdot w$$

solution

$$\begin{aligned} w \cdot x^T + b &= 0 \\ w &= {\color{blue}v_w^{-1}}(\mu_1 - \mu_2) \\ b &= -w \cdot \frac{\mu_1^T + \mu_2^T}{2} \end{aligned}$$



Prof. Dr. Thomas A. Runkler

$+/-$ Linear Discriminant Analysis

$+$ training data have to be evaluated only once

$+$ features may be continuous

$-$ only linear class separation

$-$ vectors with missing data can not be used

Support Vector Machine

- constraint: distance b from discriminant hyperplane

$$\begin{aligned} w \cdot x_k^T + b &\geq +1 \quad \text{if } y_k = 1 \\ w \cdot x_k^T + b &\leq -1 \quad \text{if } y_k = 2 \end{aligned}$$

- solution by minimizing $\|w\|^2 \Leftrightarrow$ maximizing b
- if not solvable: slack variable ξ

$$\begin{aligned} w \cdot x_k^T + b &\geq +1 - \xi_k \quad \text{if } y_k = 1 \\ w \cdot x_k^T + b &\leq -1 + \xi_k \quad \text{if } y_k = 2 \end{aligned}$$

- with penalty term in cost function

$$J = \|w\|^2 + \gamma \cdot \sum_{k=1}^{n} \xi_k, \quad \gamma > 0$$

Prof. Dr. Thomas A. Runkler

Support Vector Machine

- classification: determine w and b
- possibility: quadratic programming
- alternative: normal vector as linear combination of the data

$$w = \sum_{y_j=1} \alpha_j x_j - \sum_{y_j=2} \alpha_j x_j$$

- then optimization in (larger!) parameter space α
- classification rule

$$\begin{aligned}
\sum_{y_j=1} \alpha_j x_j x_k^T - \sum_{y_j=2} \alpha_j x_j x_k^T + b &\geq +1 - \xi_k \quad \text{if } y_k = 1 \\
\sum_{y_j=1} \alpha_j x_j x_k^T - \sum_{y_j=2} \alpha_j x_j x_k^T + b &\leq -1 + \xi_k \quad \text{if } y_k = 2
\end{aligned}$$

Kernel Trick

- idea: transform the data $X = \{x_1, \ldots, x_n\} \in \mathrm{IR}^p$ to $X' = \{x'_1, \ldots, x'_n\} \in \mathrm{IR}^q$, $q \gg p$, so that the structure in X' is easier than in X
- example: non linearly separable data $X \rightarrow$ linearly separable data X'
- Mercer's theorem (1909!): $\exists$ a mapping $\varphi : \mathrm{IR}^p \rightarrow \mathrm{IR}^q$ so that

$$k(x_j, x_k) = \varphi(x_j) \cdot \varphi(x_k)^T$$

- kernel trick: scalar product in X' = kernel function in X

Kernel Functions

- linear kernel

$$k(x_j, x_k) = x_j \cdot x_k^T$$

- polynomial kernel

$$k(x_j, x_k) = (x_j \cdot x_k^T)^d, \qquad d \in \{2, 3, \ldots\}$$

- Gaussian kernel

$$k(x_j, x_k) = e^{-\frac{\|x_j - x_k\|^2}{\sigma^2}}, \qquad \sigma > 0$$

- hyperbolic tangent kernel

$$k(x_j, x_k) = 1 - \tanh \frac{\|x_j - x_k\|^2}{\sigma^2}, \qquad \sigma > 0$$

- RBF kernel

$$k(x_j, x_k) = f\left(\|x_j - x_k\|\right)$$

Support Vector Machine

- SVM classification rule with kernel trick (nonlinear!)

$$\begin{aligned}
\sum_{y_j=1} \alpha_j k(x_j, x_k) - \sum_{y_j=2} \alpha_j k(x_j, x_k) + b \ &\geq \ +1 - \xi_k \quad \text{if } y_k = 1 \\
\sum_{y_j=1} \alpha_j k(x_j, x_k) - \sum_{y_j=2} \alpha_j k(x_j, x_k) + b \ &\leq \ -1 + \xi_k \quad \text{if } y_k = 2
\end{aligned}$$

+/− Support Vector Machine

+ nonlinear class borders
+ features may be continuous

– high computational effort for solving optimization problem
– many algorithmic parameters
– vectors with missing data can not be used

Nearest Neighbor Classifier

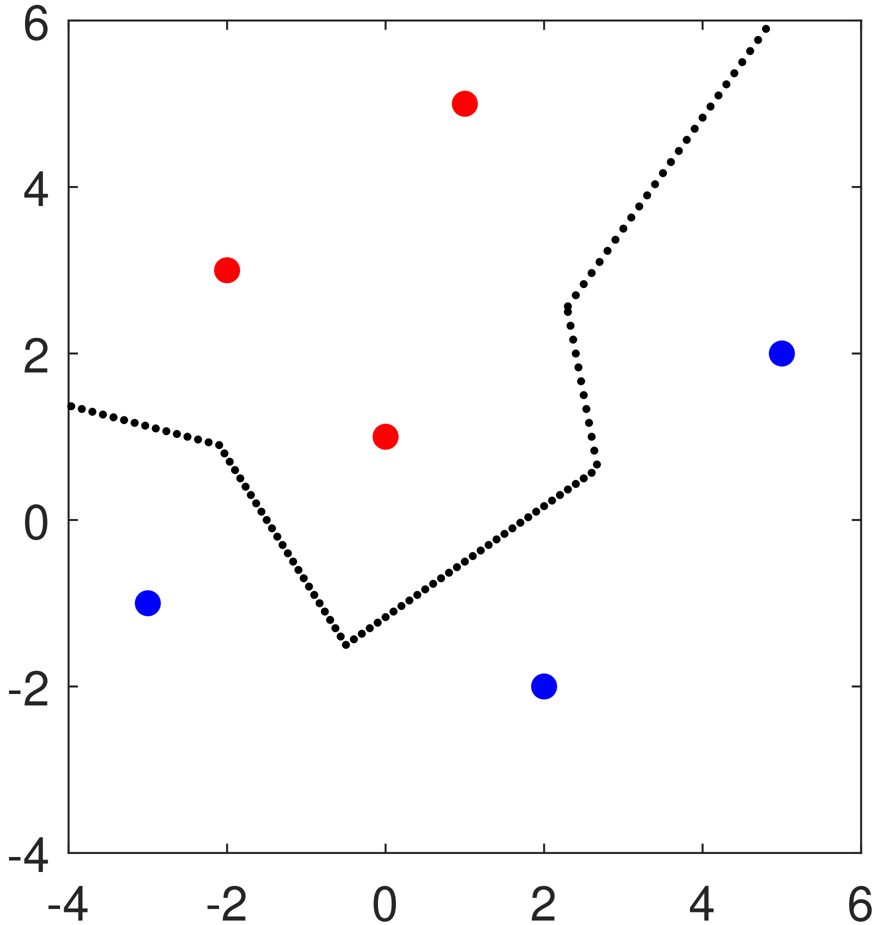
- object is assigned to the class of the object with the most similar features
- nearest neighbor classifier

$$y(x) = \underset{j=1,\ldots,n}{\arg\min}\{|x - x_j|\}$$

- k nearest neighbors classifier

$$y(x) = \text{most frequent class of the } k \text{ nearest neighbors}$$

Nearest Neighbor Classifier



Prof. Dr. Thomas A. Runkler

$+/-$ Nearest Neighbor Classifier

$+$ nonlinear class borders

$+$ features may be continuous

$+$ with suitable metric features may be dependent

$-$ training data have to be evaluated at each test

$-$ vectors with missing data can not be used

Learning Vector Quantization (Kohonen '88)

1. given classified data set
   $X = \{x_1, \ldots, x_n\} \subset \mathrm{IR}^p$, $y = \{y_1, \ldots, y_n\} \subset \{1, \ldots, c\}$,
   class number $c \in \{2, \ldots, n-1\}$, weight function $\alpha(t)$
2. initialize $V = \{v_1, \ldots, v_c\} \subset \mathrm{IR}^p$, $t = 1$
3. for $k = 1, \ldots, n$
   (a) determine winner prototype $v_i \in V$ with

   $$\|x_k - v_i\| \leq \|x_k - v\| \quad \forall v \in V$$

   (b) move winner prototypes

   $$v_i = \begin{cases} v_i + \alpha(t)(x_k - v_i) & \text{if } y_k = i \\ v_i - \alpha(t)(x_k - v_i) & \text{otherwise} \end{cases}$$
4. $t = t + 1$
5. repeat from (3.) until termination criterion
6. output: prototypes $V = \{v_1, \ldots, v_c\} \subset \mathrm{IR}^p$

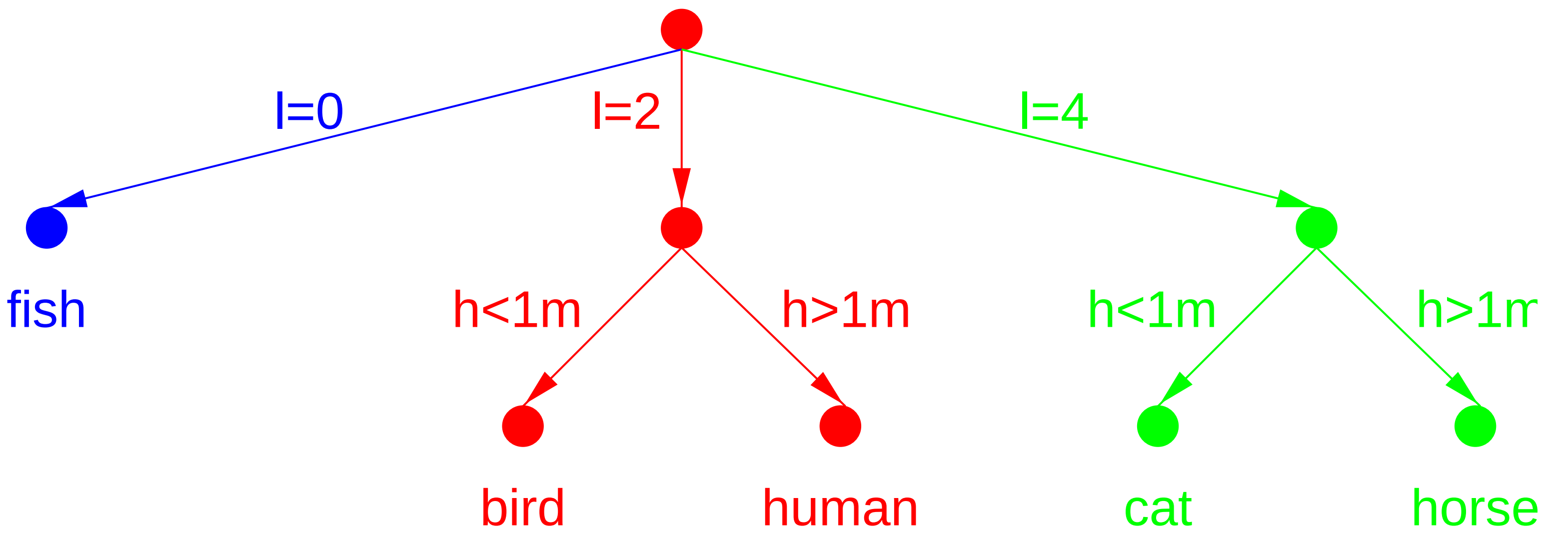
Prof. Dr. Thomas A. Runkler

+/− Learning Vector Quantization

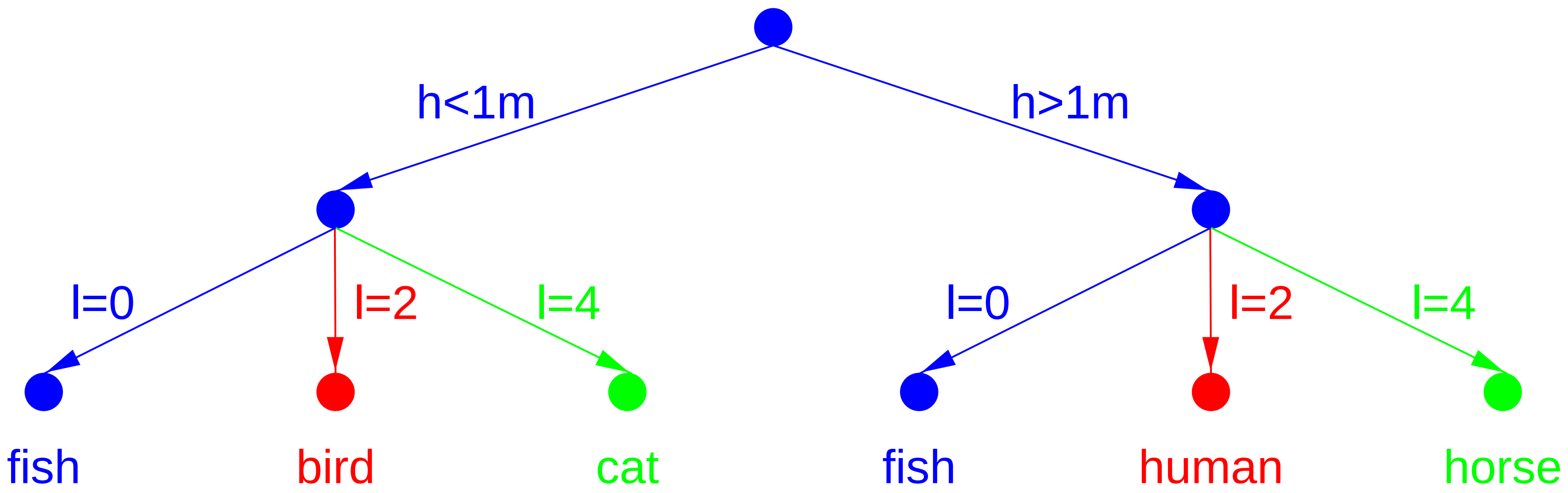
+ less test effort than nearest neighbor classifier
+ features may be continuous
+ with suitable metric features may be dependent

− high effort to determine prototypes
− pure heuristics
− vectors with missing data can not be used

Decision Tree



Decision Tree



Information Gain

- entropy at the root node

$$H(Z) = -\sum_{k=1}^{c} \underbrace{\frac{|\{y \in Y \mid y = k\}|}{|Y|}}_{p(y=k)} \log_2 \underbrace{\frac{|\{y \in Y \mid y = k\}|}{|Y|}}_{p(y=k)}$$

- information gain by additional feature j

$$g_j = H(Z) - \underbrace{\sum_{k=1}^{v_j} \underbrace{\frac{|\{x \in X \mid x^{(j)} = k\}|}{|X|}}_{p(x^{(j)}=k)} H(Z \mid x^{(j)} = k)}_{\substack{\text{expected} \\ \text{value} \\ \text{of entropy} \\ \text{for feature } j}}$$

ID3

- input $X = \{x_1, \ldots, x_n\} \subset \{1, \ldots, v_1\} \times \ldots \times \{1, \ldots, v_p\}$, $Y = \{y_1, \ldots, y_n\} \subset \{1, \ldots, c\}$
  call ID3$(X, Y, \text{root}, \{1, \ldots, p\})$

- procedure ID3(X, Y, N, I)
  1. if $I = \{\}$ or all Y are equal then break
  2. compute information gain $g_i(X, Y)\ \forall i \in I$
  3. determine winner feature $j = \text{argmax}\{g_i(X, Y)\}$
  4. partition X, Y into v_j disjoint subsets

     $$X_i = \{x_k \in X \mid x_k^{(j)} = i\}, \quad Y_i = \{y_k \in Y \mid x_k^{(j)} = i\}, \quad i = 1, \ldots, v_j$$

  5. for i with $X_i \neq \{\}$, $Y_i \neq \{\}$
     - create new node N_i and append it to N
     - call ID3$(X_i, Y_i, N_i, I \backslash \{j\})$

Example ID3

index	1	2	3	4	5	6	7	8
height h [m]	0.1	0.2	1.8	0.2	2.1	1.7	0.1	1.6
legs l	0	2	2	4	4	2	4	2
species	fish	bird	human	cat	horse	human	cat	human

$$H(Z) = -\frac{1}{8}\log_2\frac{1}{8} - \frac{1}{8}\log_2\frac{1}{8} - \frac{3}{8}\log_2\frac{3}{8} - \frac{2}{8}\log_2\frac{2}{8} - \frac{1}{8}\log_2\frac{1}{8} \approx 2.1556 \text{ bit}$$

$$\begin{aligned} H(Z \mid h < 1m) &= -\frac{1}{4}\log_2\frac{1}{4} - \frac{1}{4}\log_2\frac{1}{4} - \frac{2}{4}\log_2\frac{2}{4} \approx 1.5 \text{ bit} \\ H(Z \mid h > 1m) &= -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} \approx 0.8113 \text{ bit} \end{aligned}$$

$$g_h = H(Z) - \frac{4}{8}H(Z \mid h < 1m) - \frac{4}{8}H(Z \mid h > 1m) \approx 1 \text{ bit}$$

Example ID3

index	1	2	3	4	5	6	7	8
height h [m]	0.1	0.2	1.8	0.2	2.1	1.7	0.1	1.6
legs l	0	2	2	4	4	2	4	2
species	fish	bird	human	cat	horse	human	cat	human

$$\begin{aligned} H(Z \mid l=0) &= -1 \log_2 1 = 0 \\ H(Z \mid l=2) &= -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} \approx 0.8113 \text{ bit} \\ H(Z \mid l=4) &= -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} \approx 0.9183 \text{ bit} \end{aligned}$$

$$g_l = H(Z) - \frac{1}{8}H(Z \mid l=0) - \frac{4}{8}H(Z \mid l=2) - \frac{3}{8}H(Z \mid l=4) \approx 1.4056 \text{ bit}$$

Prof. Dr. Thomas A. Runkler

+/− ID3

\+ only relevant features are needed
\+ additional information gain by hierarchical structure

− only axis parallel class borders
− features have to be discrete

