Im Rahmen der Vorlesung „Neuronale Netze“:

Learning Vector Quantization (and beyond)

Siehe auch im Skript:

„Neuronale Netze“, Kap. 7.2, S.109 ff. – LVQ

10. Juni 2002

Marc Strickert

http://www.inf.uos.de/lnm

Aufgabe:

Hochdimensionale Merkmalsvektoren sollen auf einen charakteristischen Zustand abgebildet werden.

$$\mathbb{R}^N \rightarrow \{c_1, c_2, c_3, \ldots, c_M\}$$

Vektor - Quantisierung

Beispiel: Farbklassifikation bei Mischfarben

$$[0,1]_r \times [0,1]_g \times [0,1]_b \rightarrow \{\text{lila, braun, gelb, sonstige}\}$$

Annahme: Daten mit derselben Klassenzugehörigkeit sind einander ähnlich und bilden mehr oder weniger zusammenhängende Wolken im Datenraum.

Idee: Punkte die zu einer Klasse gehören werden von einem Stellvertreter für diese Punktemenge und ihrer Klassenzugehörigkeit repräsentiert.

-> Daten-Prototypen.

LVQ - Algo

Mit der Euklidischen Norm als Ähnlichkeitsmaß kann folgende Heuristik für das Lernen der Prototypen-Positionen $\mathbf{w}^i$ aus den Trainingsdaten $\mathbf{x}$ formuliert werden (Kohonen, 1989):

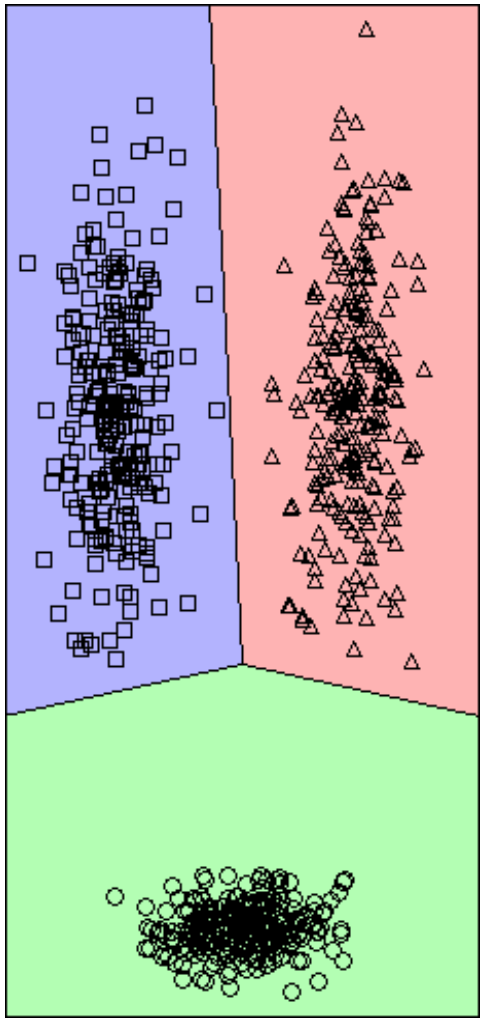
wiederhole

berechne $y^i = |\mathbf{x} - \mathbf{w}^i|$

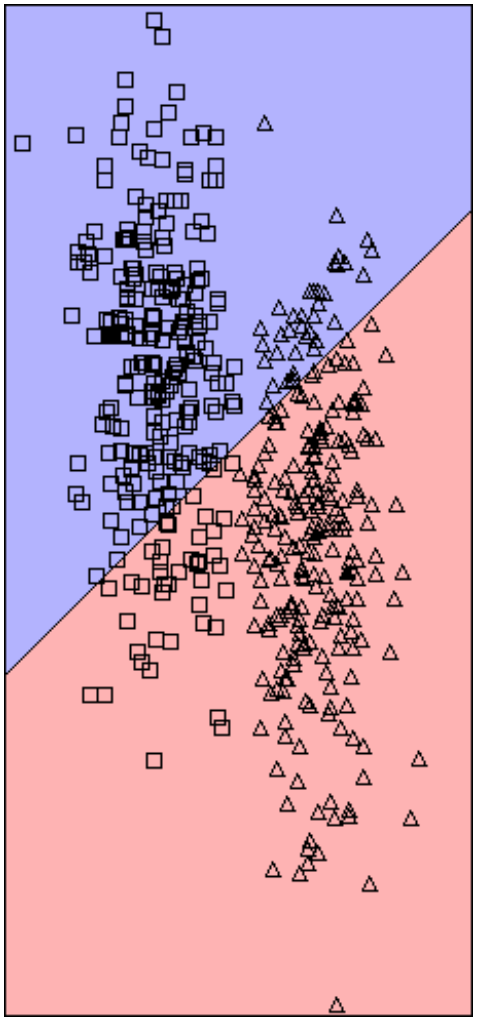
für ein i mit minimalem y^i

$$\mathbf{w}^i := \mathbf{w}^i + \eta(\mathbf{x} - \mathbf{w}^i)$$

LVQ - Demo



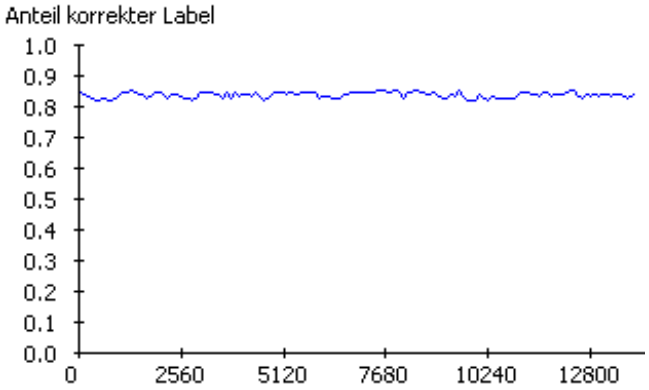
(A) 3 Klassen in 2D, perfekt getrennt.



(B) 2 Klassen in 2D, suboptimal getrennt.

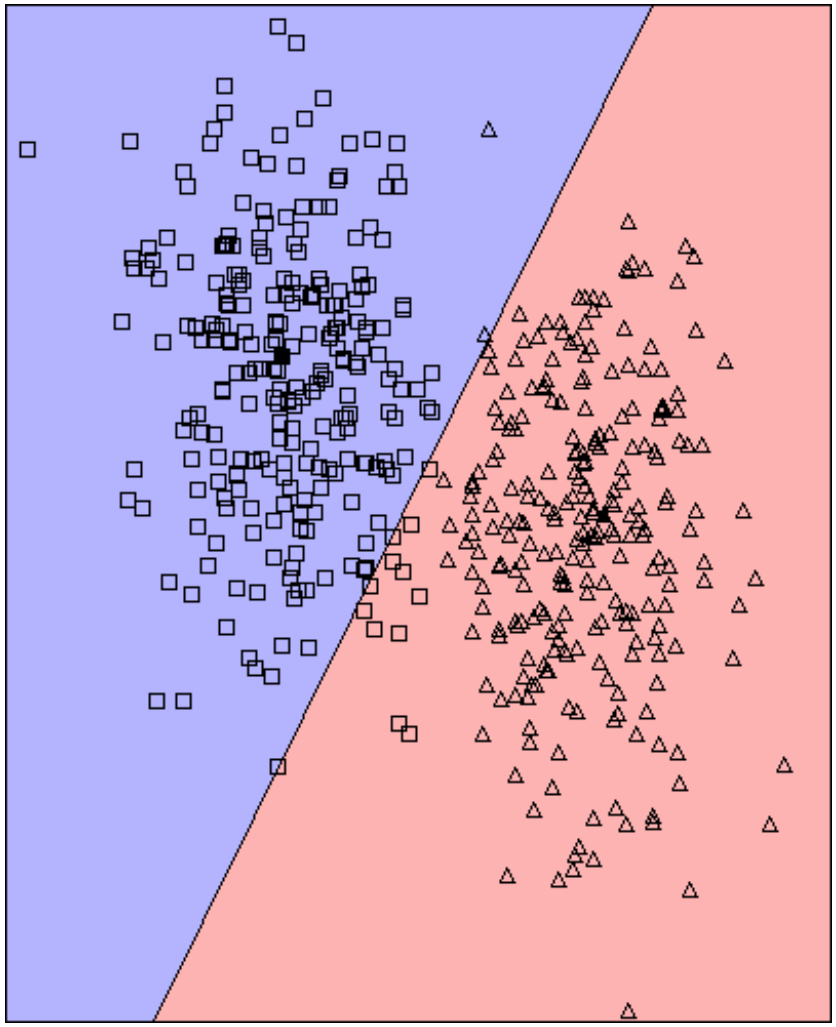
Zu (B)

Die Gaußschen Cluster liegen ungünstig, so dass LVQ nur eine ungünstige Trennebene findet, obwohl eine optimale Trennung horizontal möglich wäre.



Bilder: Screenshot von Softy, Java Code von Andreas Rechtien, LNM.

Daten-Transformation I



(C) 2 Klassen in 2D aus (B), z-transformiert.

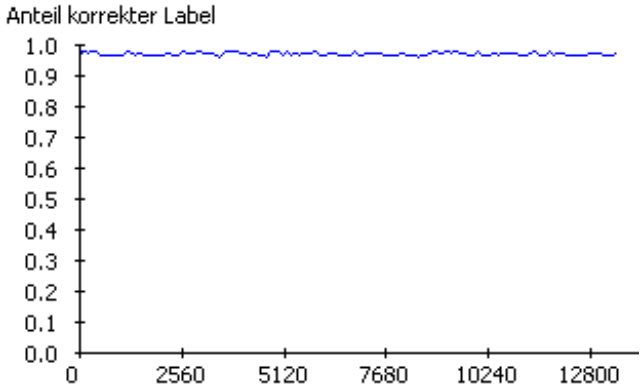
Z-Transformation der Dimensionen von (B)

$$\text{Wert}_Z = \frac{\text{Wert - Mittelwert}}{\text{Standardabweichung}}$$

=> Daten werden ausreißertolerant um 0 verteilt.

=> Datenraum wird in den Dimensionen homogen.

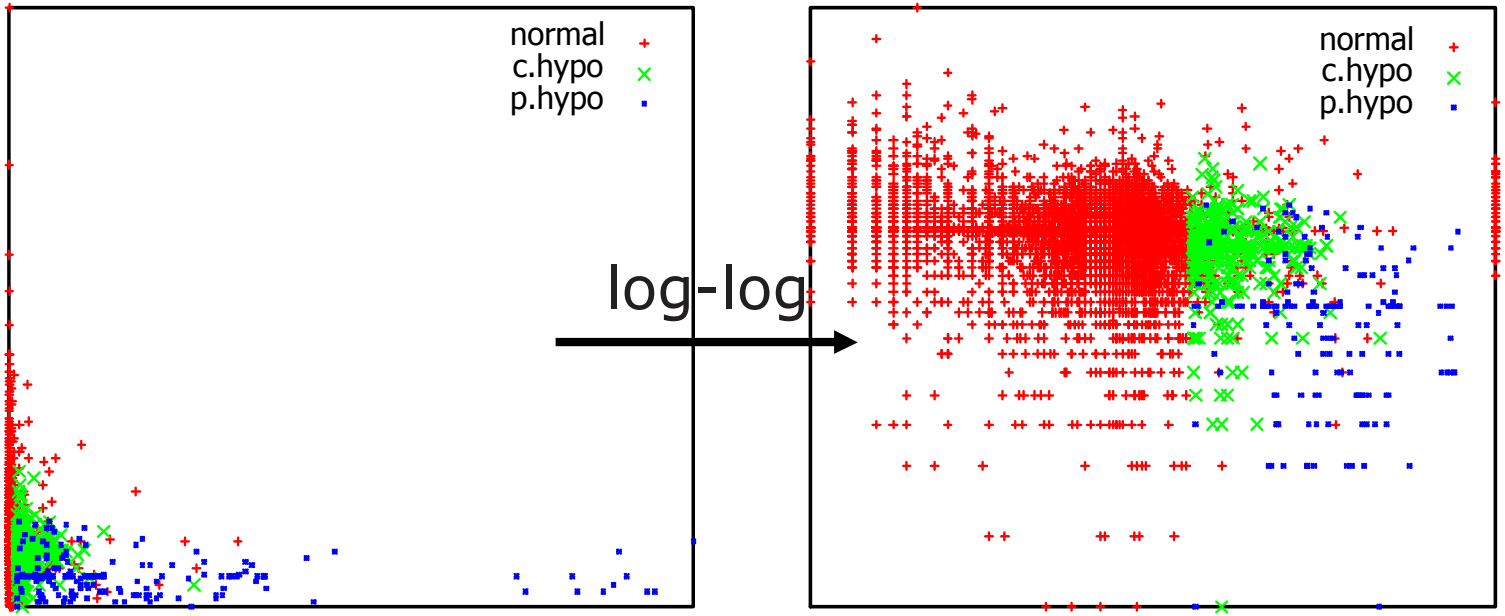
-> Verbesserung von LVQ und anderer Verfahren.



Daten-Transformation II

Neben der Z-Transformation bieten sich weitere Transformationen an, wenn man weiß, dass Daten z.B. eine physiologische Bedeutung besitzen, also Potenz- oder Exponentialgesetze eine Rolle spielen.
Ziel: bessere Repräsentation durch Prototypen möglich.

Beispiel: Doppel-Logarithmische Transformation bei Daten zur Schilddrüsen-Erkrankung.



Terminologie von LVQ

Lernende Vektor-Quantisierer sind

prototypenbasierte, … Prototyp = Datenrepräsentant.

überwachte, … Daten mit Klassenzugehörigkeit (Label).

selbstorganisierende, … eine lokale Adaptations-Dynamik.

neuronale … Neuronen rangeln sich um den Input.

Verfahren.

Dimension: Attribut oder Merkmal (feature) als Komponente in der Vektor-Darstellung.

Netz := Prototypen + Datenabstandsmaß (Metrik).

Im Folgenden: Erweiterungen von LVQ

Ein Vorteil von LVQ ist das intuitive Verständnis bezüglich der Update-Strategie für die Prototypen-Positionen.

Ein weiterer Vorteil: in dieser Originalform spielt der Umfang einzelner Klassen für das Endergebnis keine Rolle (Häufigkeits-Verteilungs-Unabhängigkeit).

Demgegenüber stellen sich einige Nachteile:

- Wie gesehen: ungünstig gelegene Daten führen zu suboptimalen Klassifikations-Ergebnissen.
- Die Zahl der Prototypen für multimodale Daten ist unklar.
- Die initialen Prototypenpositionen haben großen Einfluss.

Die meisten Erweiterungen von LVQ berücksichtigen zur Verbesserung der Ergebnisse neben dem nächsten richtigen Prototypen auch den nächsten falschen Prototypen, um eine aktive Klassenseparation durch Einstellung ihrer Entscheidungsgrenzen durchzuführen.

Dynamic **L**earning **V**ector **Q**uantization

Problem:
Wieviele Prototypen sollen spendiert werden?

DLVQ

1. Starte mit je 1 Prototyp pro Klasse,
   initialisiert im Schwerpunkt.

2. Solange Verbesserung möglich,
   passe ähnlich wie LVQ Prototypen weiter an.

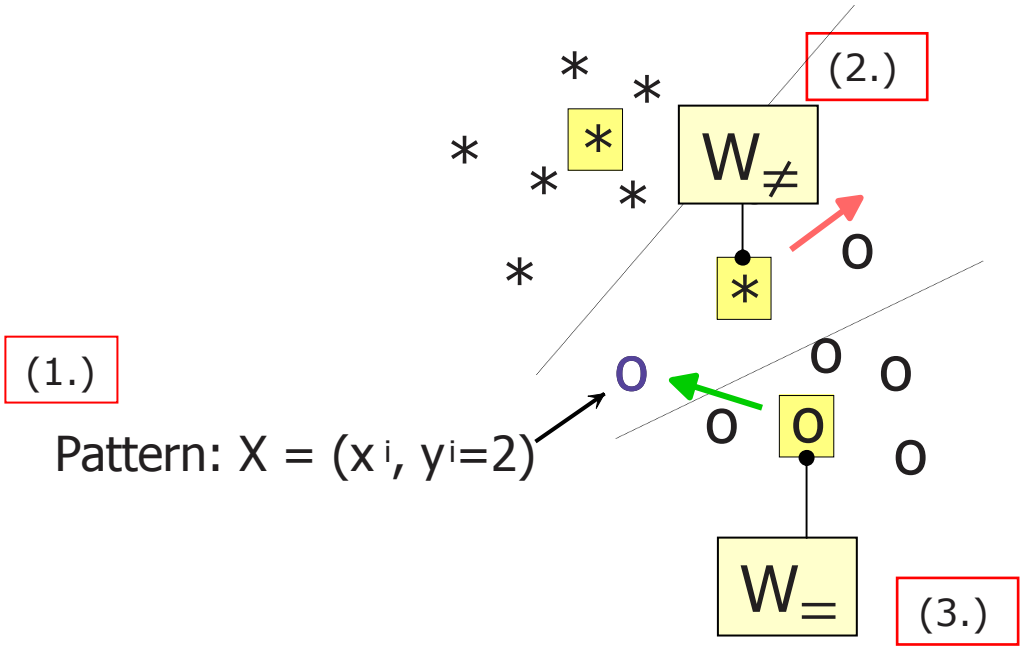
3. Falls weitere keine Verbesserung möglich,
   a. breche ab, falls o.k. !
   b. spendiere neuen Prototyp im Schwerpunkt
      schlecht klassifizierter Daten.

Relevance Learning Vector Quantization

$$d_E(X, W) = \sqrt{\sum_i (x_i - w_i)^2} \quad \rightarrow \quad d_R(X, W) = \sum_i \lambda_i (x_i - w_i)^2$$

bekannte Euklidische Distanz

new: **R**elevance factor for dimension i



$\Delta w_i \propto \pm (x_i - w_i)$ LVQ mit nächstem richtigen / falschen Prototyp.

$\Delta \lambda_i \propto \mp (x_i - w_i)^2$ $\lambda_i \geq 0$ normalized, $\lambda_i := \lambda_i / |\lambda|$.

Intuition 2 Math: GRLVQ I/II

Nachteil des intuitiven Ansatzes bei (R)LVQ:

Konvergiert nicht immer!

Denn: Anpassung der Dimensionsgewichte ist

äquivalent zu Perzeptron-Lernen.

=> Zyklen bei rauschbehafteten Daten.

II
V

Gradientenabstieg: Generalized Relevance LVQ

Fehlerfunktion : $E = \sum_x f((d_1 - d_2)/(d_1 + d_2))$.

mit adapt. Metrik $d^2(x, y, \boldsymbol{\lambda}) := \sum_{i=1}^{n} \lambda_i (x_i - y_i)^2$, $\lambda_i \geq 0$.

Prototype Adaptation

Closest *correct* prototype $\tilde{W}^1$:

$$\tilde{W}^1 := \tilde{W}^1 + \epsilon\, f' \cdot \frac{d_2}{(d_1+d_2)^2}\,(x - \tilde{W}^1)\ .$$

Closest *wrong* prototype $\tilde{W}^2$:

$$\tilde{W}^2 := \tilde{W}^2 - \epsilon\, f' \cdot \frac{d_1}{(d_1+d_2)^2}\,(x - \tilde{W}^2)\ .$$

Weight Adaptation

$$\lambda_l := \lambda_l - \epsilon_1\, f' \cdot \left(\frac{d_2}{(d_1+d_2)^2}\,(x_l^i - \tilde{W}_l^1)^2 - \frac{d_1}{(d_1+d_2)^2}\,(x_l^i - \tilde{W}_l^2)^2 \right)\ .$$

Intuition 2 Math: Verallgemeinert

Grundidee: *Minimierung einer geeigneten Fehlerfunktion führt zur Maximierung der Klassifikationsgenauigkeit*.

$$C_X(w,\lambda) \stackrel{!}{=} \min$$

- Position der Prototypen, lokal
- Dimensionsgewichte, global

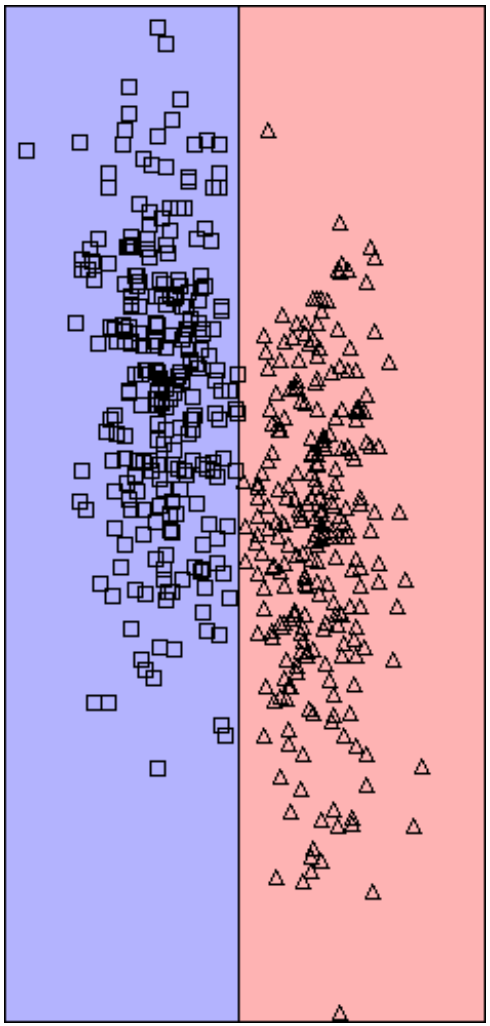
$$C \leftarrow \sum_{\text{Muster } X} \sum_{\text{Prototypen } W} \left(\text{hochdimensionale Metrik}\right)$$

$$\frac{\partial C_X(w,\lambda)}{\partial w} \rightarrow 0 \qquad w \leftarrow w + \alpha_1 \cdot f_1 \cdot (x - w)$$

Minimierung durch stochastischen Gradientenabstieg

$$\frac{\partial C_X(w,\lambda)}{\partial \lambda} \rightarrow 0 \qquad \lambda \leftarrow \lambda + \alpha_2 \cdot f_2 \cdot (x - w)$$

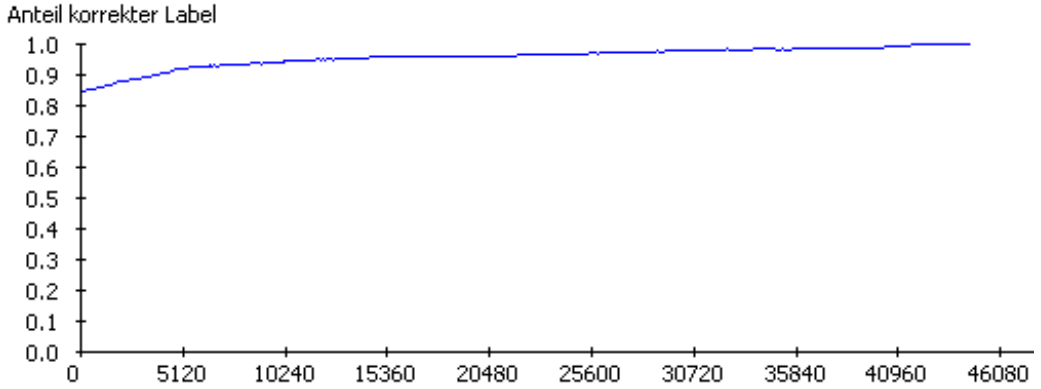
GRLVQ - Demo Time



GRLVQ-Klassifikation des ‚schwierigen' Datensatzes.

GRLVQ erkennt, dass nur die x-Richtung für die Klassifikation relevant ist: $\lambda_1=1$, $\lambda_2=0$.

Die nach rund 50000 Musterpräsentationen ermittelte Trennlinie ist optimal.



Mehr Demos

GRLVQ + Verbesserte Initialisierung

(a) 10-D künstliche Daten mit 3 Klassen.

! Daten-Überlapp mit Rauschen:

2 erzeugende Dimensionen, 1 linear-Komb., 7-dim Rauschen

? Empirische Konvergenz.

? Dimensions-Relevanzen.

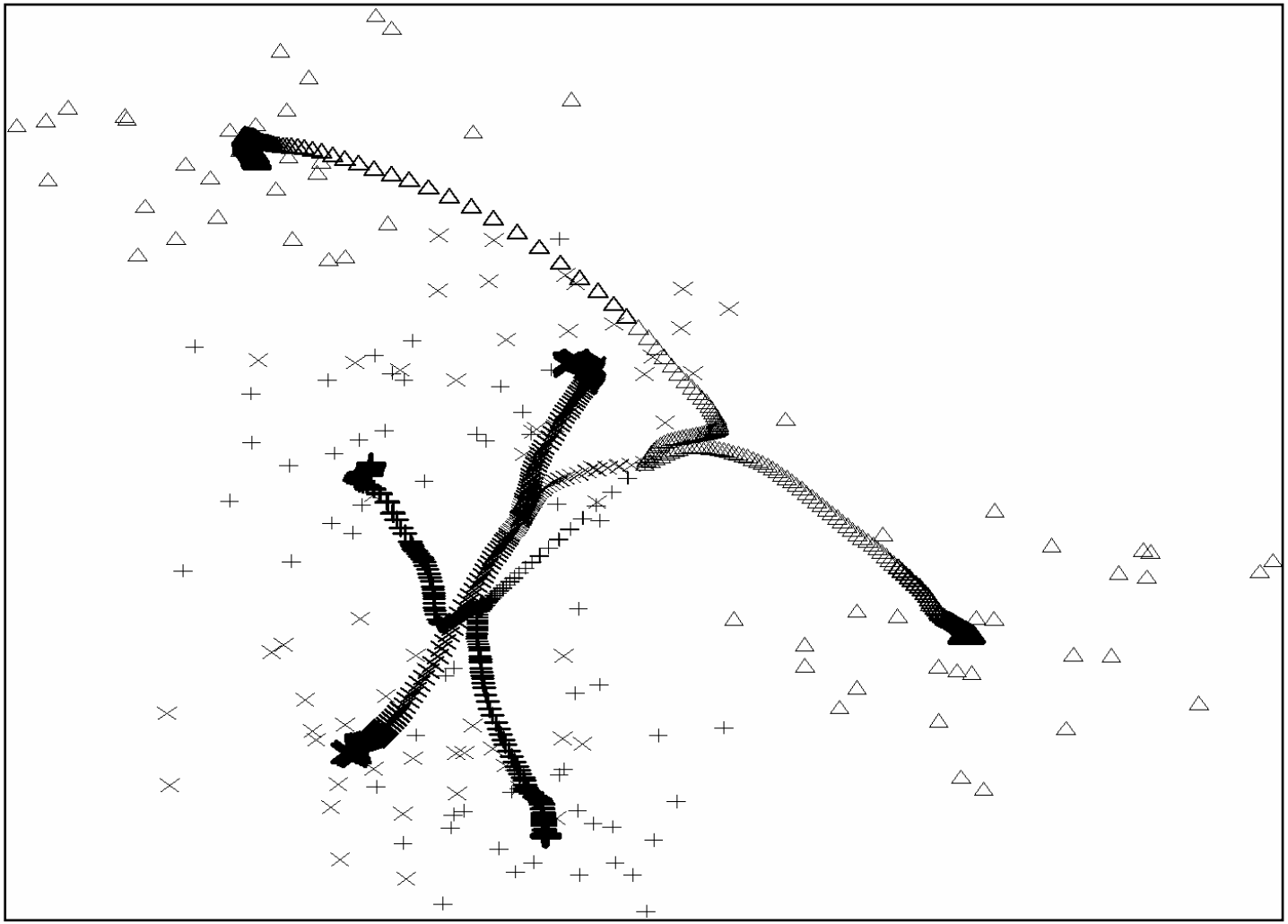
(b) 2-D multi-modale Problem mit 2 Klassen.

! Crispe Cluster.

? Initialisierungs-Toleranz.

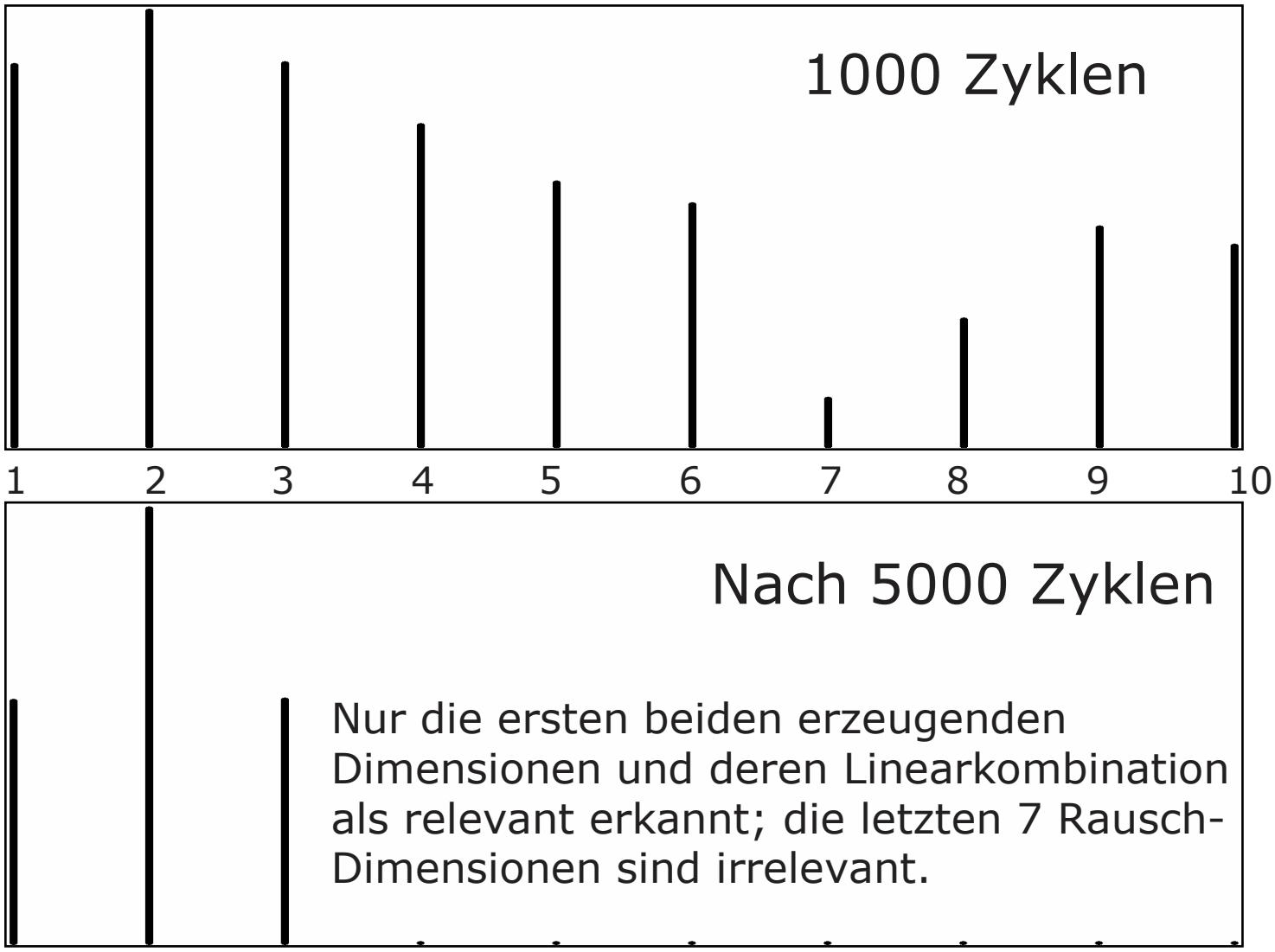
? Konvergenz.

Demo (a): 10-D Daten



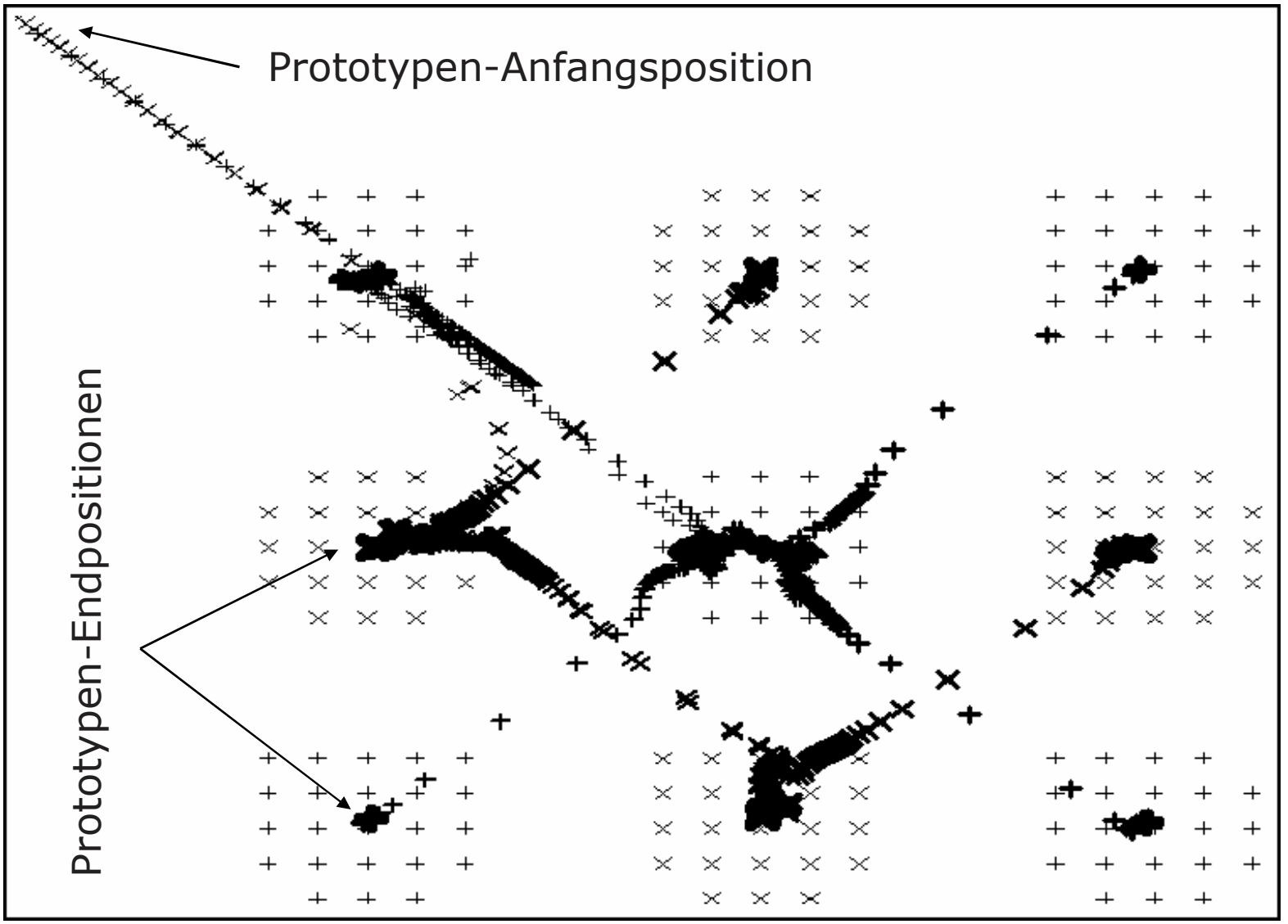
Projektion auf erste beide Dimensionen

Demo (a): 10-D Daten



Relevanzen der Dimensionen

Demo (b): 2-D multimodale Daten



Anwendungen

Dimensions-Reduktion:

...bei hochdimensionalen, multispektralen Satellitendaten.

Diagnose:

Überwachung von Fehlerzuständen bei Kolbenmaschinen.

Nichtlineare Zeitreihen-Analyse:

Attraktor-Rekonstruktion von Wasserabfluss-Messreihen.

Regelextraktion

Prototypen reduzieren bereits die Komplexität der Daten, aber:

Mentaler Zugang zu den konvexen Voronoi-Zellen nicht immer leicht bei Interpretation eines trainierten Netzes.

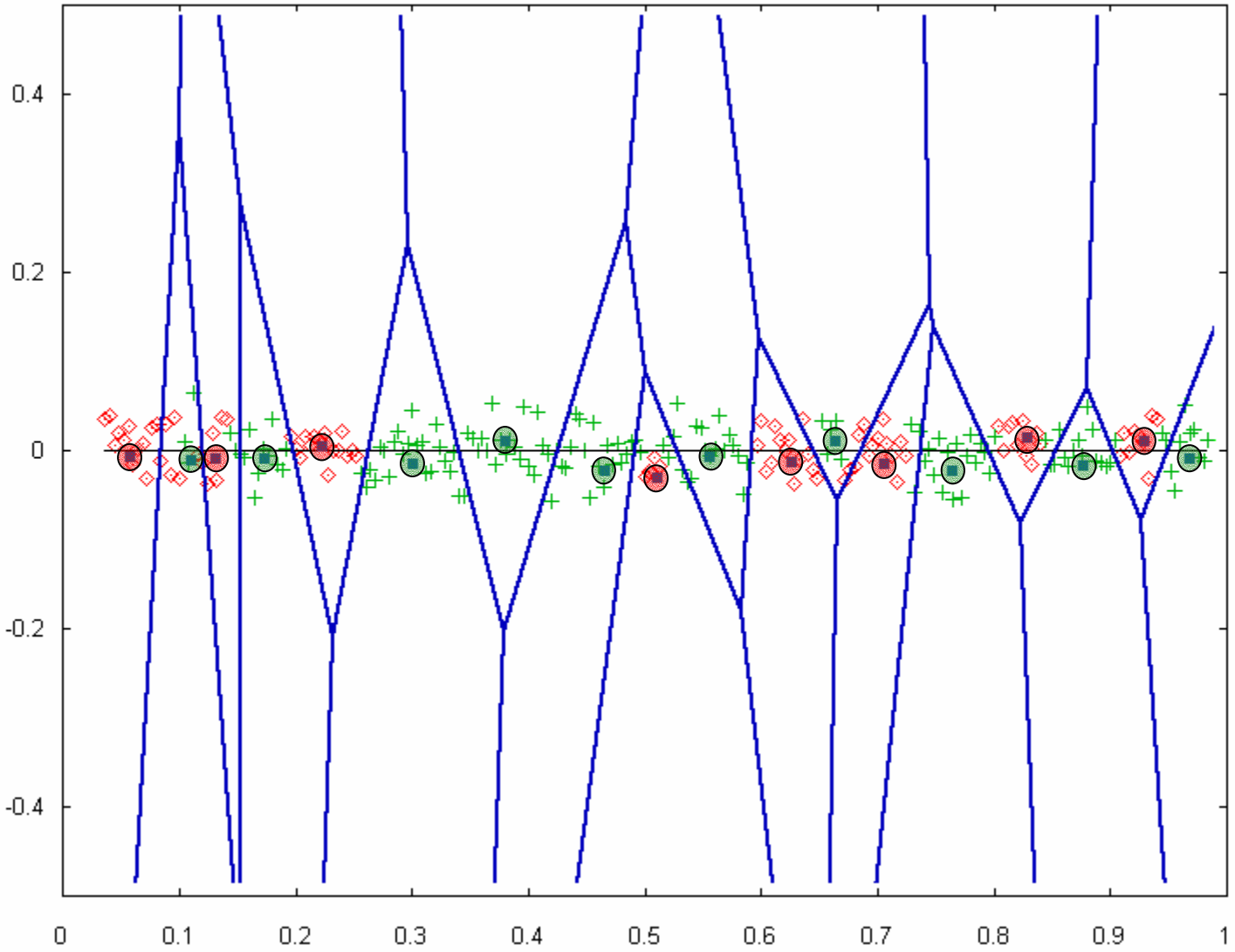
-> Ähnlichkeits- und Analogie-Argumente.

Wunsch:

Verwandle deutliche Merkmale eines trainierten Netzes durch achsenparallele Zerlegung des Datenraumes in einen Entscheidungsbaum.

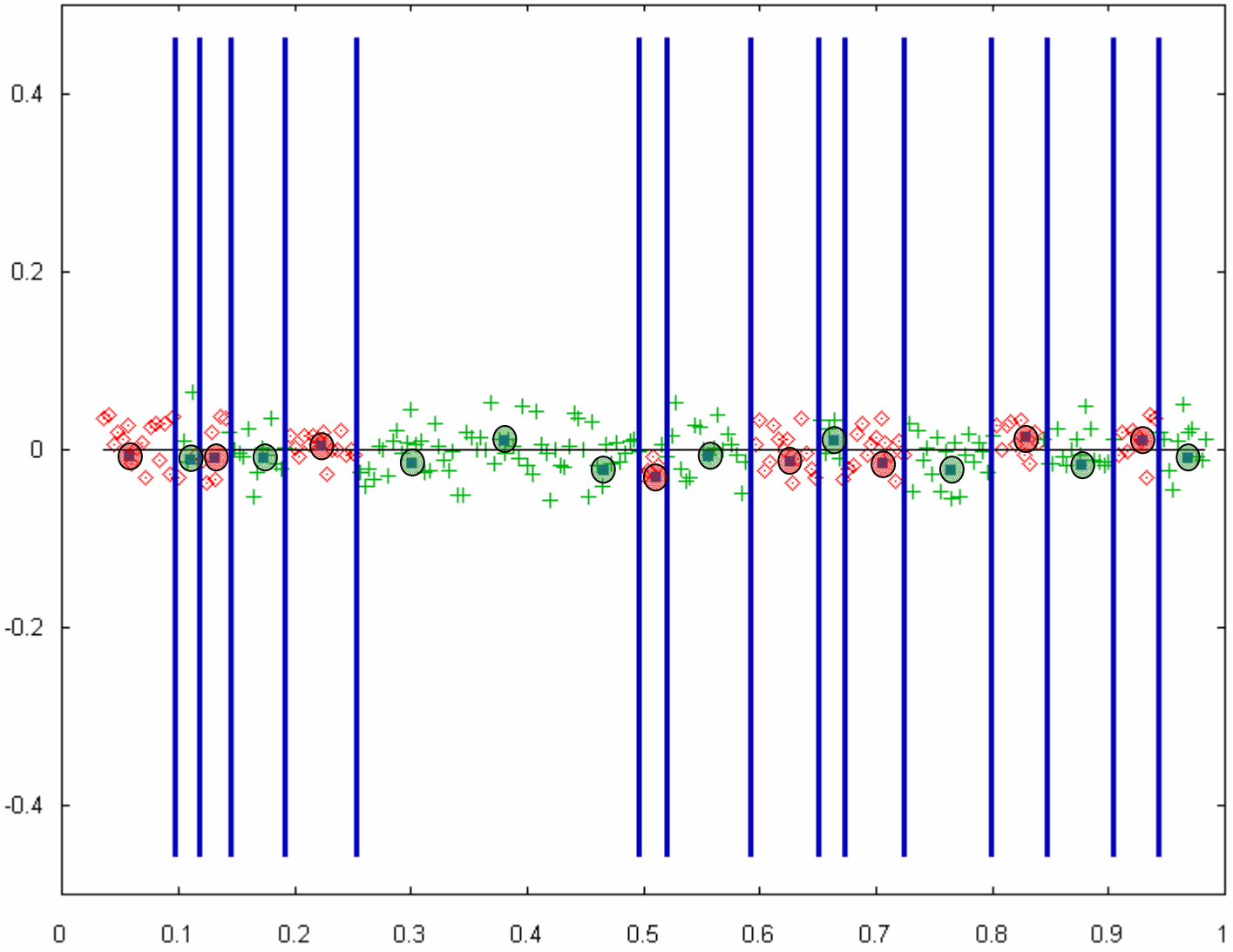
-> klare Regeln.

Entscheidungsbaum: Bsp. mit 2 Klassen



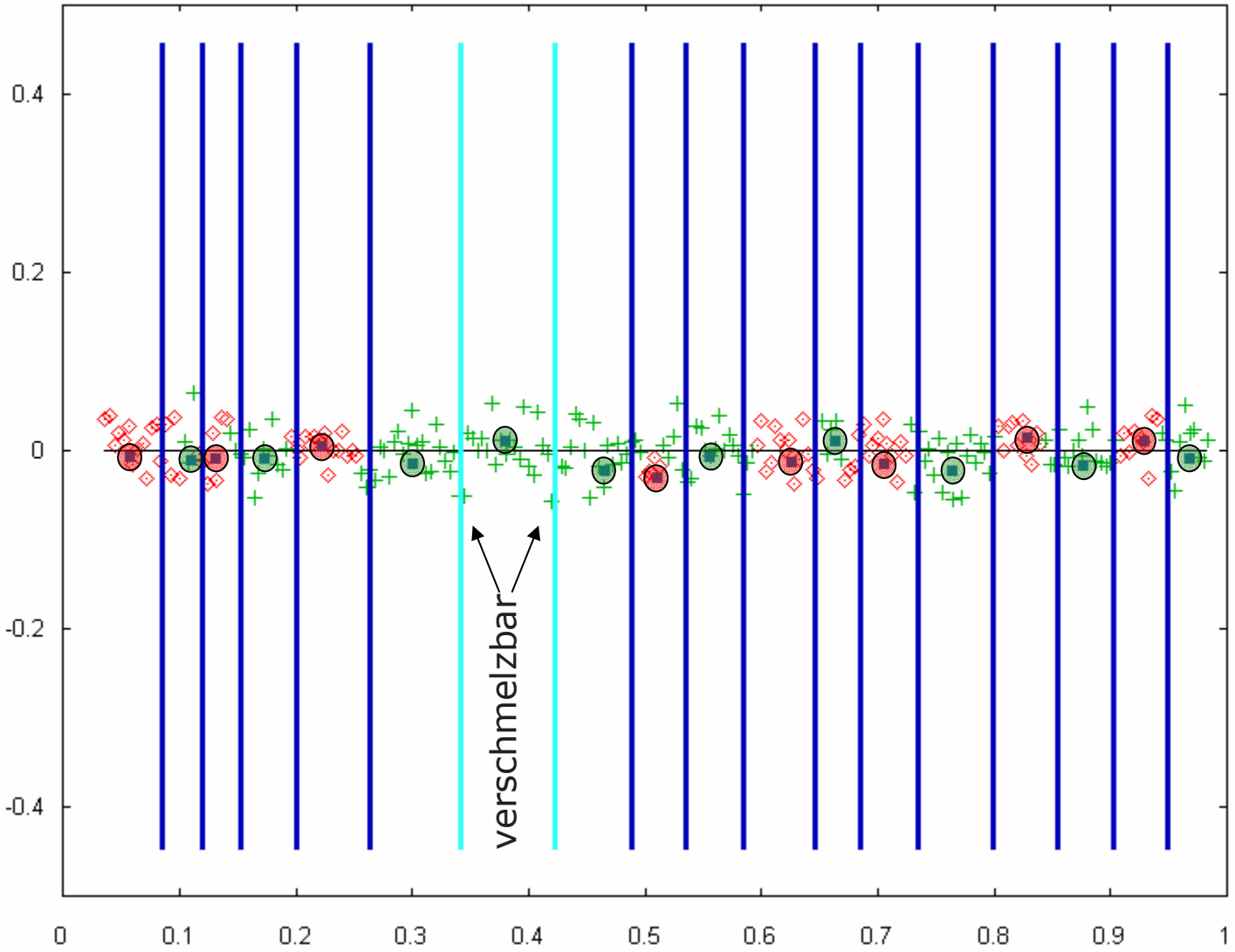
I/III. Trainiertes Netz mit Prototypen-Positionen und Entscheidungsgrenzen.

Entscheidungsbaum: Bsp. mit 2 Klassen



II/III. Optimale achsenparallele Schnitte im Datenraum.

Entscheidungsbaum: Bsp. mit 2 Klassen



III/III. Achsenparallele Schnitte bei halber Entfernung benachb. Prototypen.

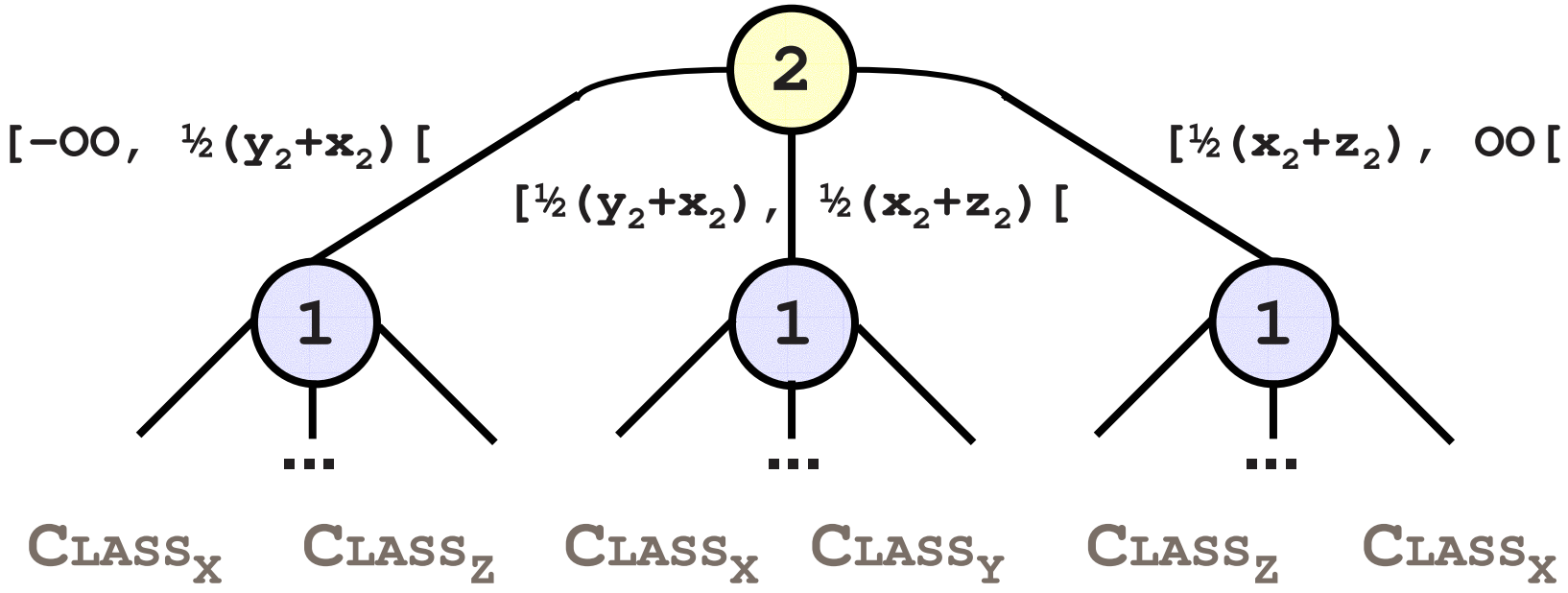
Babsibaum (BB)

Prototypen

$$X = (x_1, x_2, \ldots, x_N, \text{CLASS}_X)$$
$$Y = (y_1, y_2, \ldots, y_N, \text{CLASS}_Y)$$
$$Z = (z_1, z_2, \ldots, z_N, \text{CLASS}_Z)$$

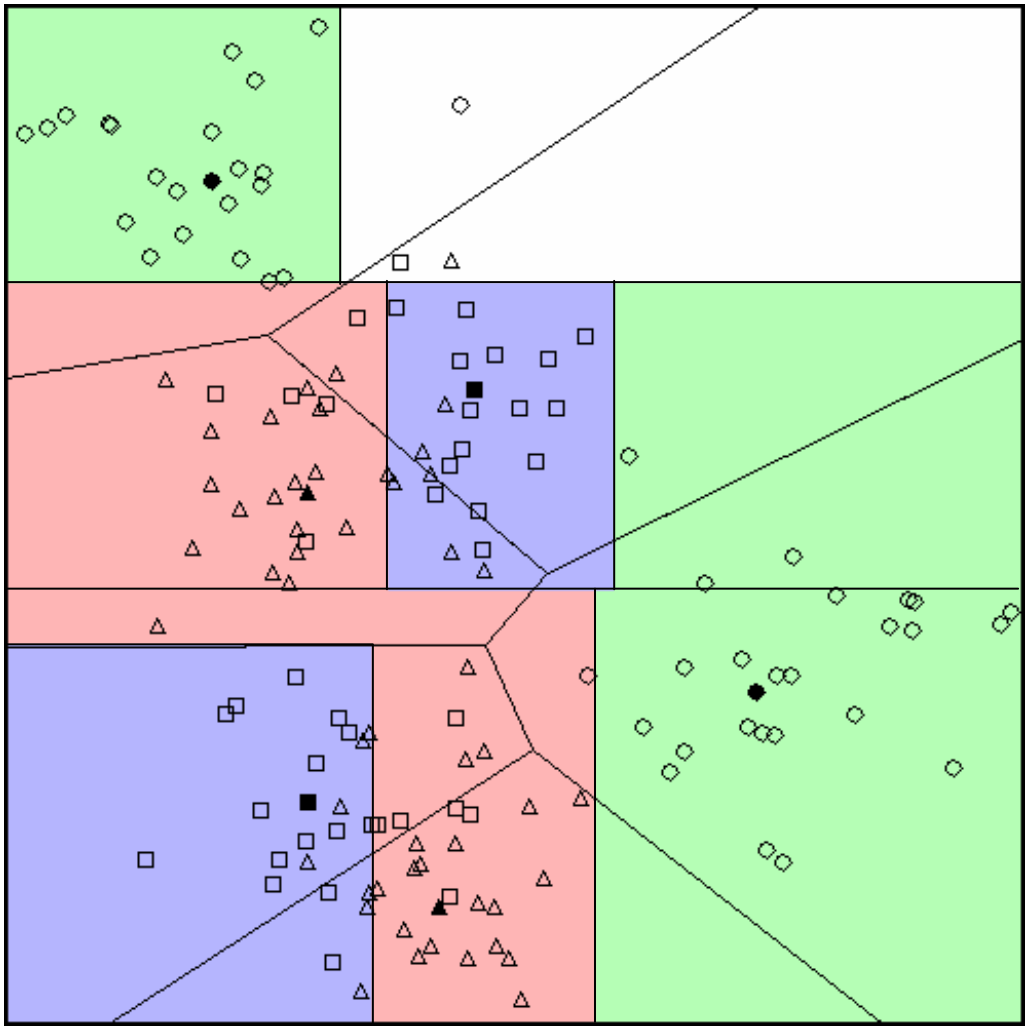
Relevanzen

$$\lambda = (0.3, 0.4, \ldots, 0.1)$$



BB Demo erneut für künstliche 10D-Daten

Nach Folie 15: relevanteste Komponente - y x - 2t wichtigste Kompon.



-Infinity
-Infinity
14.7% Class= 2
0.321
2.1% Class = -1
Infinity
0.276
-Infinity
16.2% Class= 1
0.368
14.8% Class= 0
0.597
2.1% Class= 2
Infinity
0.587
-Infinity
16.2% Class= 0
0.349
19.0% Class= 1
0.576
14.8% Class= 2
Infinity
Infinity

BB-Schnitte und rezeptive Felder

Richtige Labels für $BB_{hight=2}$: 79.5%

BB – Algorithmus formal

BB-Tree (X, W, Λ):

if STOP: output a leaf with class $argmax_c |\{x^i \mid y^i = c, (x^i, y^i) \in X\}|$

else: output an interior node N *with* $|W|$ *children,*

choose $I^N := first(\Lambda)$,

compile a sorted list $[a_1, \ldots, a_{|W|}]$ *from* $\{w^i_{I^N} \mid w^i \in W\}$

choose $W^N_i := (a_i + a_{i+1})/2$, $i = 1, \ldots, |W| - 1$

choose the i *th child of* N, $i = 1, \ldots, |W|$, *as the output of*

BB-Tree $(\{(x, y) \in X \mid x_{I^N} \in (W^N_{i-1}, W^N_i]\}, W, rest(\Lambda) \bullet [first(\Lambda)])$

Pilzdaten: Wichtige Merkmale I/II

Demonstration: Konvergenz der Relevanzen.

Daten

Features: 117 unär kodierte Dimensionen.

Classes: **e**dible, **p**oisonous.

Patterns: 8124.

Klassifikations-Genauigkeit (GLRVQ): 97.5% Test-Set.

Pilzdaten: Kodierung der Nominal-Daten

x,s,n,t,p,f,c,n,k,e,e,s,s,w,w,p,w,o,p,k,s,u -> p (giftig)

x,s,y,t,a,f,c,b,k,e,c,s,s,w,w,p,w,o,p,n,n,g -> e (essbar)

22-dimensionale Nominaldaten

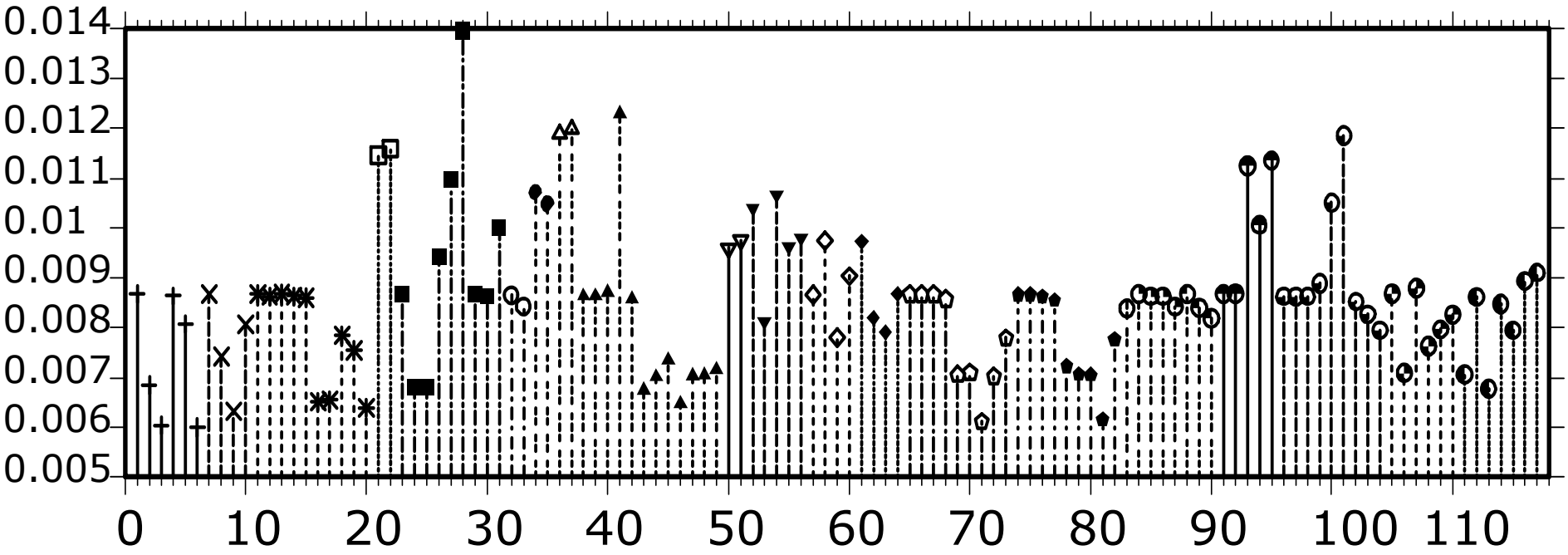
				unäre Kodierung	
1	1	6	c		1 0 0 0 0 0
2	1	5	k		0 1 0 0 0 0
3	1	4	f		0 0 1 0 0 0
4	1	3	s		0 0 0 1 0 0
5	1	2	b		0 0 0 0 1 0
6	1	1	x		0 0 0 0 0 1
7	2	4	g		1 0 0 0
8	2	3	f		0 1 0 0
9	2	2	y		0 0 1 0
10	2	1	s		0 0 0 1
	...				
117	22	1	u		0 0 0 0 0 0 1

117-dimensionale unärkod. Daten

000001, 0001, 0000000001, 01, 000000001, 01, 01, 01,
000000000001, 01, 00001, 0001, 0001, 000000001,
000000001, 1, 0001, 001, 00001, 000000001, 000001,
0000001 -> 0

000001, 0001, 0000000010, 01, 000000010, 01, 01, 10,
000000000001, 01, 00010, 0001, 0001, 000000001,
000000001, 1, 0001, 001, 00001, 000000010, 000010,
0000010 -> 1

Pilzdaten: Wichtige Merkmale II/II



Top 6 der relevanten Merkmale nach GRLVQ:

1. odor = none,
2. gill-color = buff,
3. gill-size = narrow
4. gill-size = broad,
5. spore-color = chocolate,
6. bruises = no

Pilzdaten: Entscheidungstabelle

bruises 22:'no'	odor 28:'none'	gill-size 36:'broad'	gill-size 37:'narrow'	gill-color 41:'buff'	spore-print-color 101:'chocolate'	Class	Freq.
-	0	-	-	1	-	p	21%
-	0	1	0	0	1	p	19%
0	0	-	-	0	0	p	3%
1	0	0	1	0	0	p	4%
1	1	0	1	0	-	p	0.1%
0	1	-	0	0	-	e	16%
1	0	1	0	0	0	e	8%
1	1	-	0	0	-	e	25%
0	1	0	1	0	-	e	3%

$BB_{h=6}$: 97.2 %

odor = none => minimaler Konflikt

Ausblick für BB

Hybrides Modell

INNERE KNOTEN- > characteristische Regeln,

logische UND Ketten in BB: symbolisch;

BLÄTTER- > differenzierende Analogien,

Mehrheitsentscheid der Prototypen: subsymbolisch.

Was gelernt?

- Learning Vector Quantization (**LVQ**)
  - -> DLVQ : dynamische Prototypenzahl,
  - -> RLVQ : Relevanzen, empirisch,
  - -> GRLVQ : Relevanzen, Gradientenabstieg.

- Eigenschaften der LVQ. Sie…
  … sind prototypenbasiert,
  … adaptieren intuitiv oder Kostenfunktion-Minimierung,
  … benötigen oder realisieren eine (adaptive) Metrik.

- Eignen sich für hochdimensionale Daten.

- Datentransformation:

reel	-> reel	(log-log, Z-transform),
nominal	-> reel	(unäre Kodierung).

- Regeln können extrahiert werden.