



Arbeitsgemeinschaft Neuronale Netze und Kognitive Systeme

May, 15 2002

Self-organizing neural networks for structured data

Marc Strickert

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



1. Some introductory statements.
2. Supervised prototype based learning.
3. Relevance determination.
4. Rule extraction.
5. Ongoing work.
6. Outlook and (hopefully) time for questions.



Currently most relevant motivation

2

How loud am I supposed to talk, in order to be (acoustically) understood by the audience ?

- Size of the room,
- type and function of the room,
- are there elderly people,
- is anybody having a nap,
- are there in- or external noise sources,
- what's the number of listeners ?



Many factors, some of which are mutually dependent.

Is there an especially *important* factor ?

Are there *rules*, e.g.:

„If in gymnastics hall without micro. then shout ! “



Dimension: attribute or feature as a component in vector representation.

Prototype: a feature vector with a typical set of features that characterizes a set of data points belonging to a certain class.

Net: a trainable set of free parameters that can be interpreted as neurons.

Here: Prototypes competing for processing an input stimulus (Winner-Take-All), and metric.

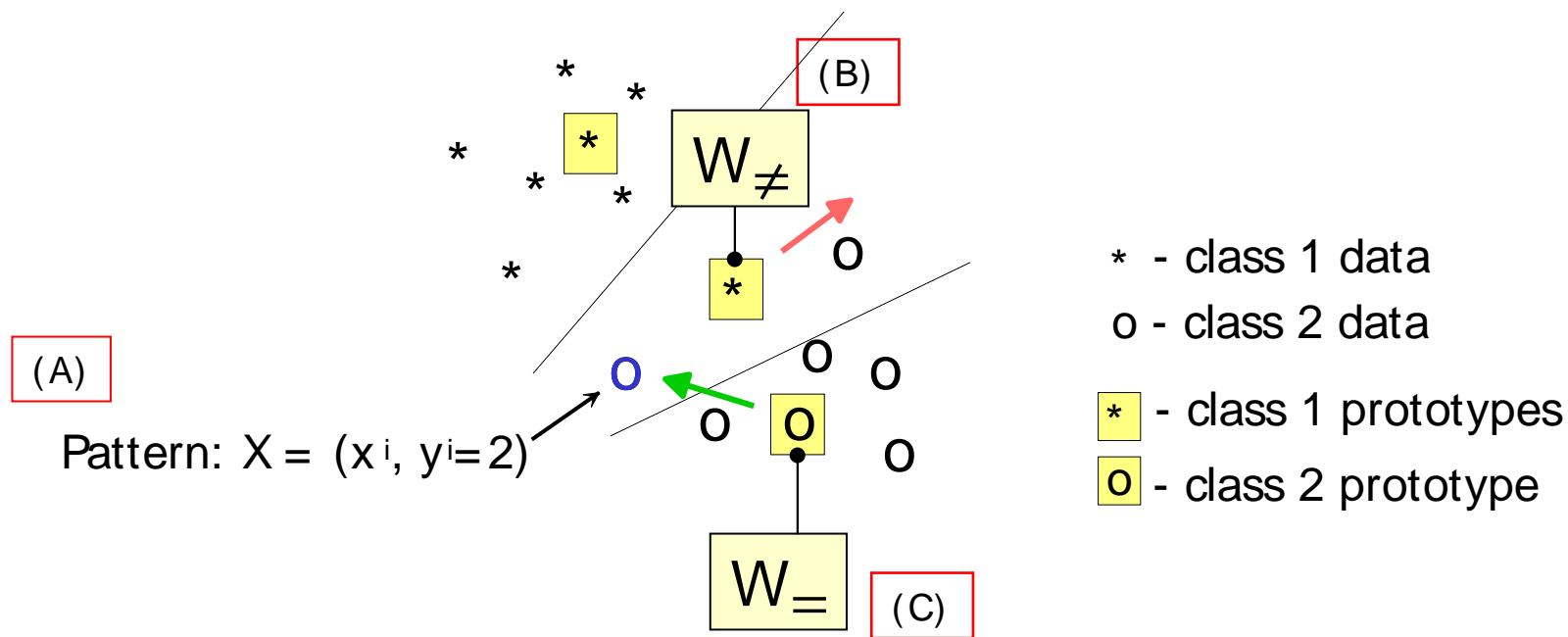


Relevance Learning Vector Quantization

$$d_1(X, W) = \sqrt{\sum_i (x_i - w_i)^2}$$

$$d_2(X, W) = \sum_i \lambda_i (x_i - w_i)^2$$

↑
new: **Relevance factor for dimension i**



$$\Delta w_i \propto \pm (x_i - w_i) \quad \text{standard LVQ.}$$

$$\Delta \lambda_i \propto \mp |x_i - w_i| \quad \lambda_i \geq 0 \text{ normalized, } \lambda_i := \lambda_i / |\lambda| .$$



Drawback of previously presented update:

Does not necessarily converge: equivalent to perceptron learning -> cycles in presence of noise.

||
v

Gradient Descent

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

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



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

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



- LVQ, LVQ2.1, LVQ3 (Kohonen, 1995)
- GLVQ (Sato, Yamada, 1996/1998)
- DSLVQ (Pregenzer, 1997)
- RLVQ (Bojer, Hammer, Schunk, Tluk v. Toschanowitz, 2001)
- GRLVQ (Hammer, Villmann, 2001)
- SRNG (Hammer, Strickert, Villmann, 2002)

Related Work:

- Metrics that learn relevance (Kaski, Sinkkonen, 2000)



SRNG

(a) 10-D artificial data with 3 classes.

! Data overlap with noise.

? Empirical convergence.

? Dimension relevances.

(b) 2-D multi-modal problem with 2 classes.

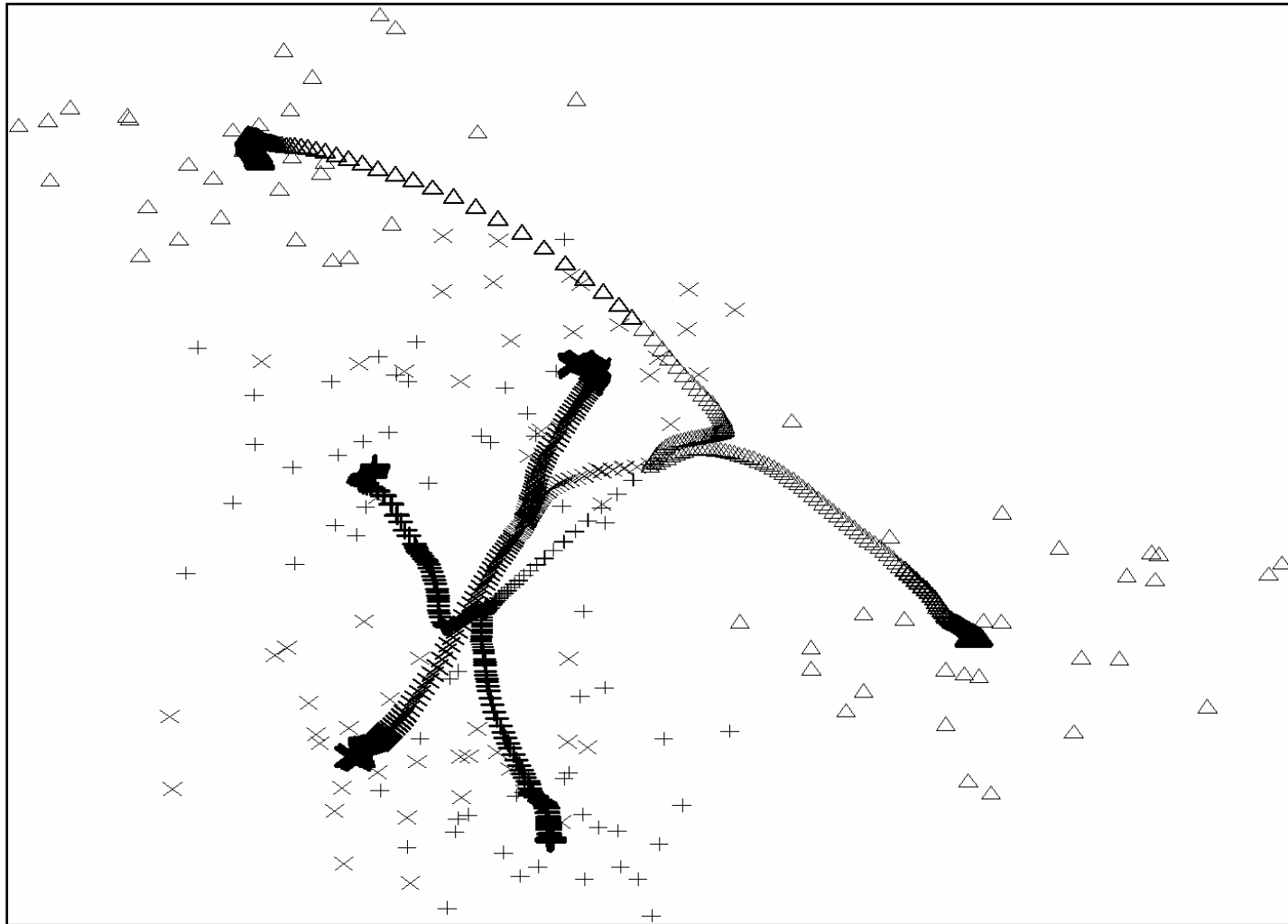
! Crisp clusters.

? Initialization tolerance.

? Convergence.



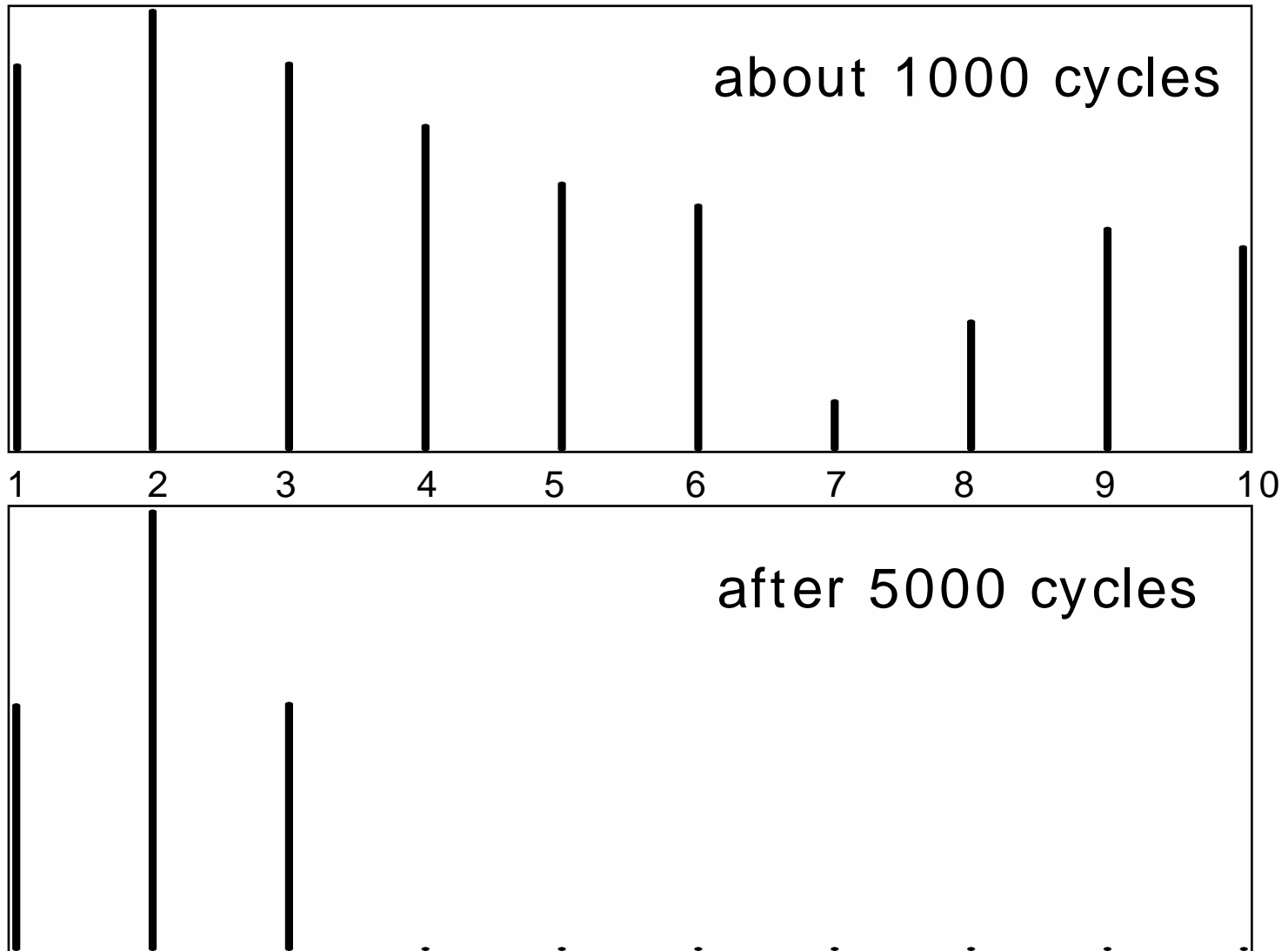
Demo (A): 10-D data



Projection to first 2 dimensions



Demo (A): 10-D data

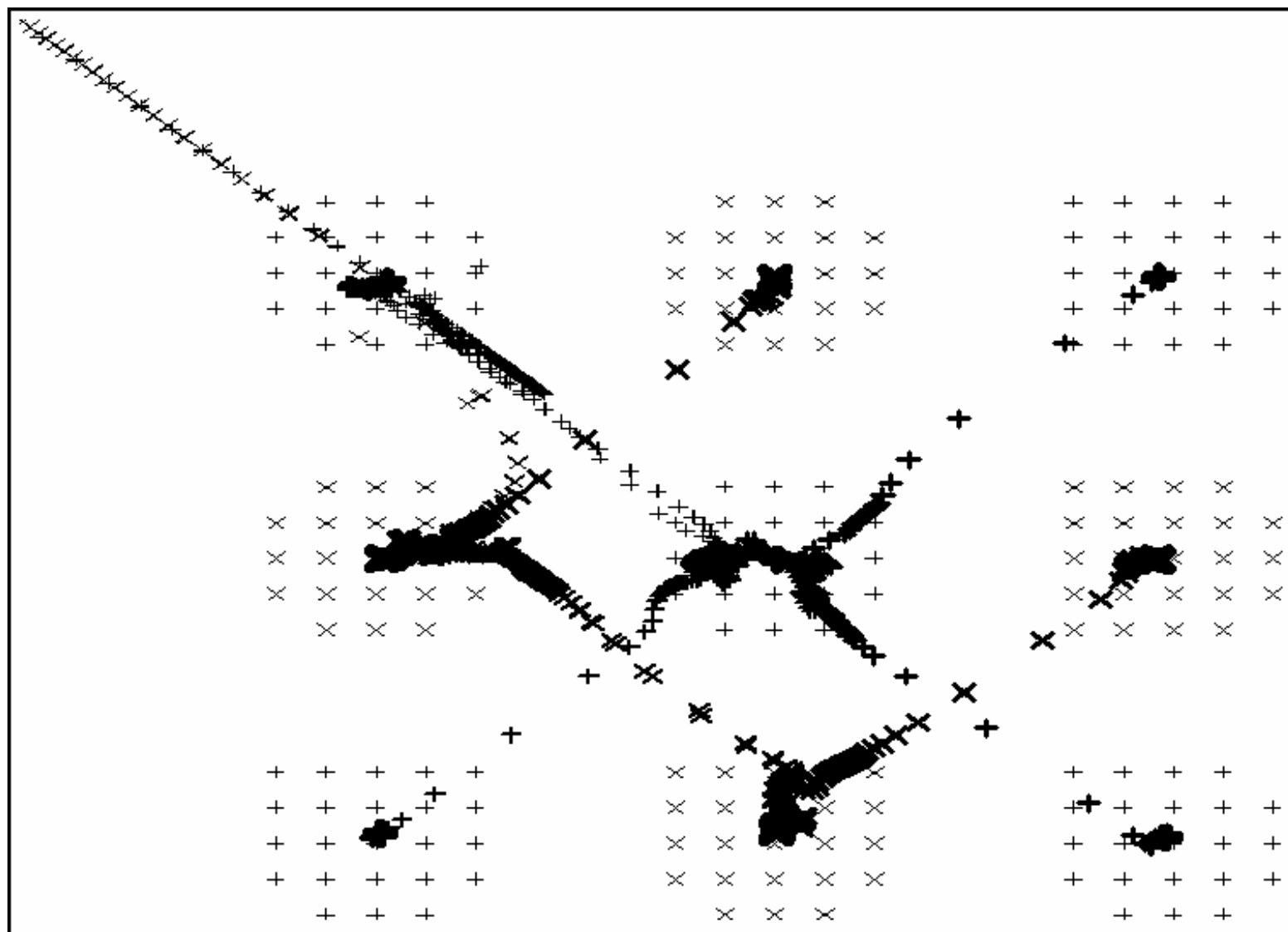


Dimension relevances



Demo (B): 2-D multimodal Data

11





Dimensionality reduction:

...of high dimensional, multispectral satellite data.

Diagnosis:

Improved vector quantization of piston engine states.

Nonlinear time series analysis:

Attractor reconstruction of water runoff observations.



Prototypes reduce complexity of the data, but:

Mental approach to interpretation of convex receptive fields of a trained net is not easy.

-> Similarity arguments.

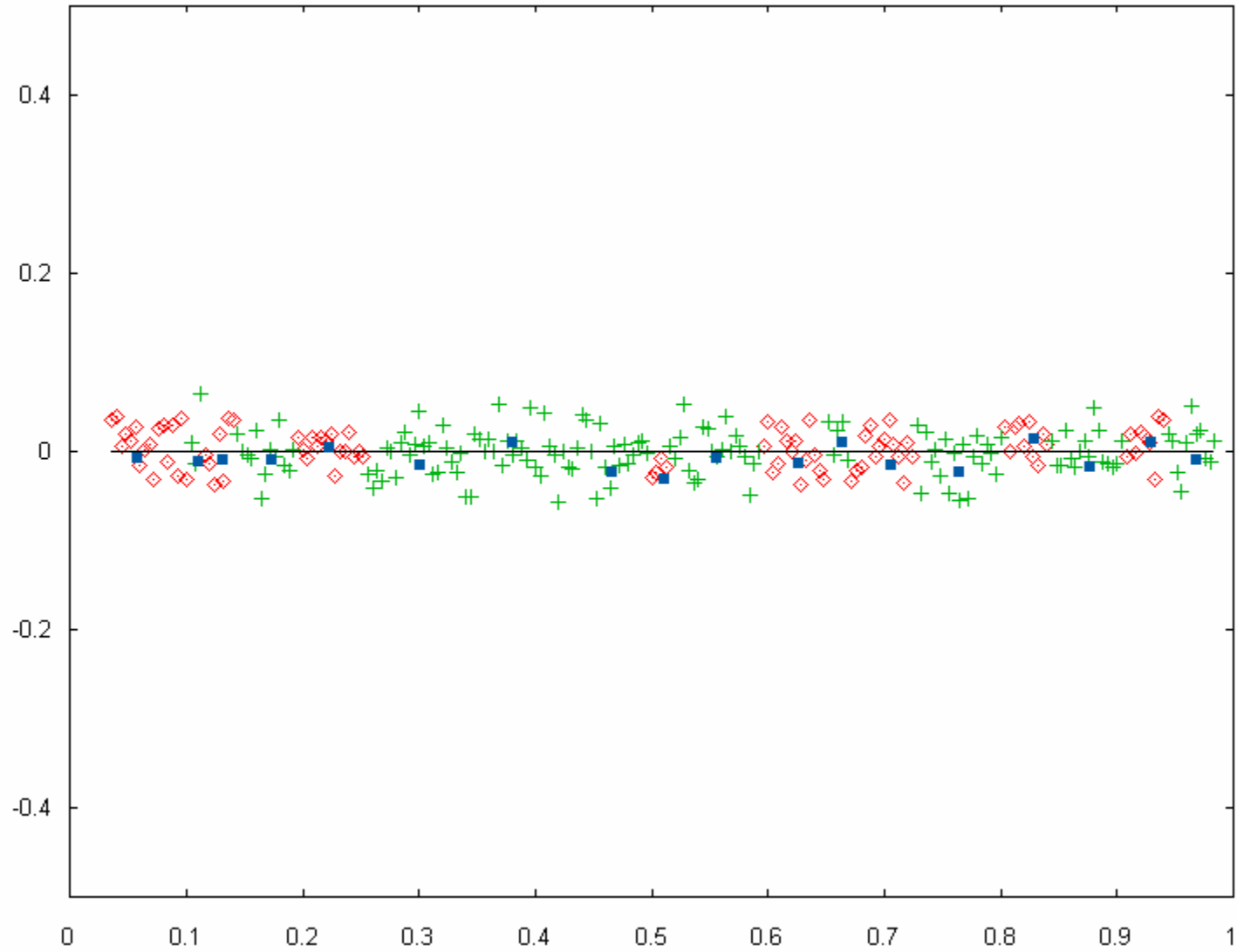
What we also want:

Convert obvious features of a trained net into a decision tree.

-> Crisp rules.

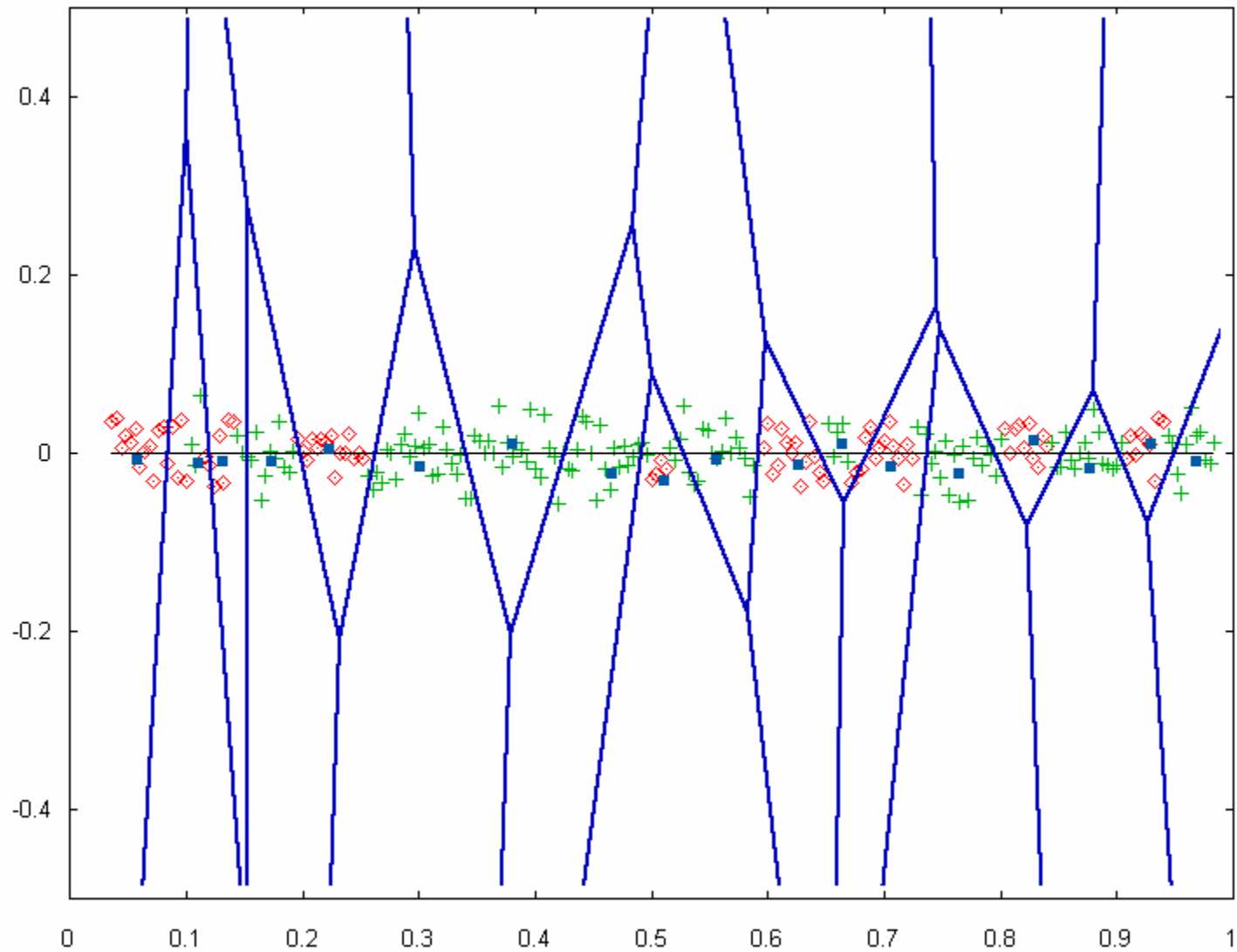


Decision tree: Toy example - 2 classes in 1.23D



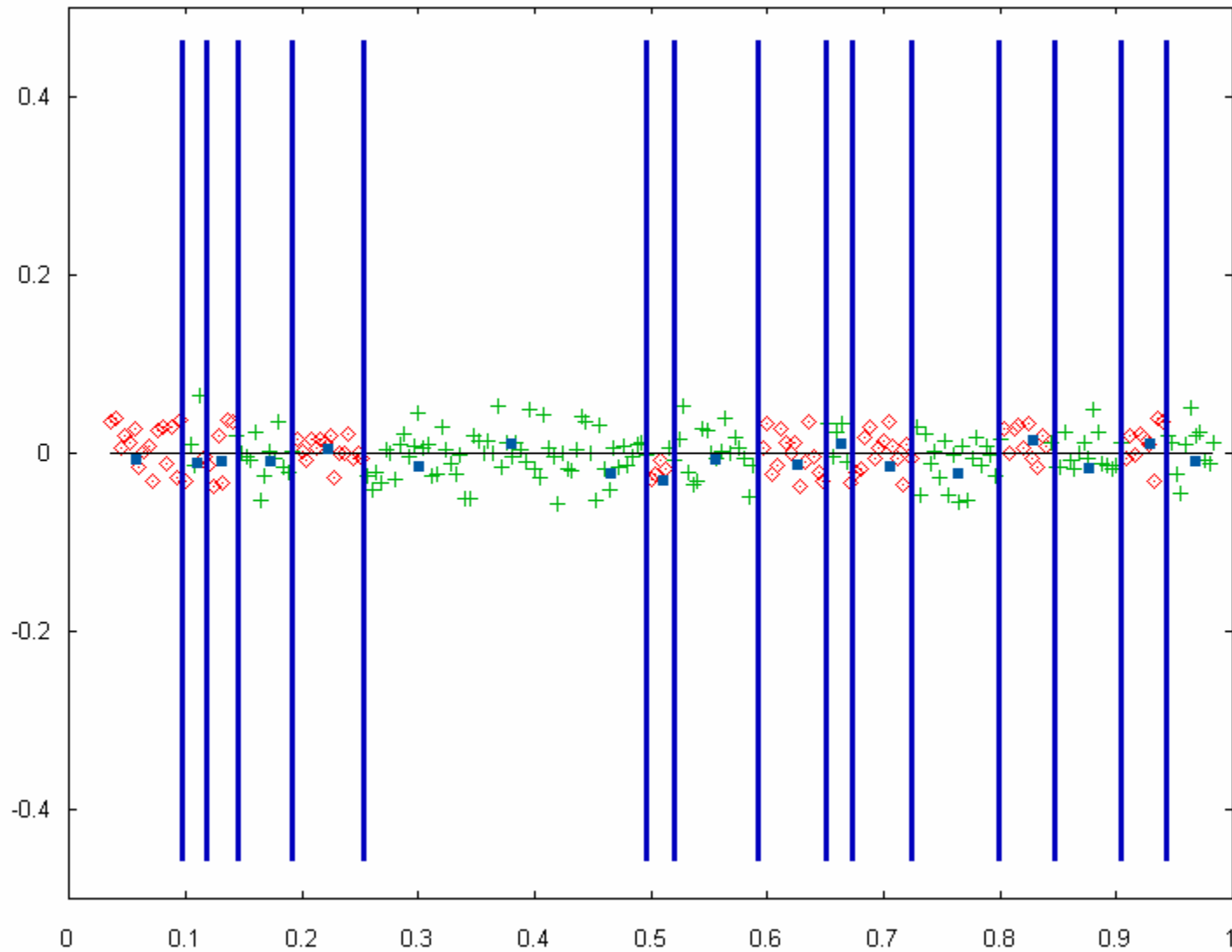


Decision tree: Toy example - 2 classes in 1.23D



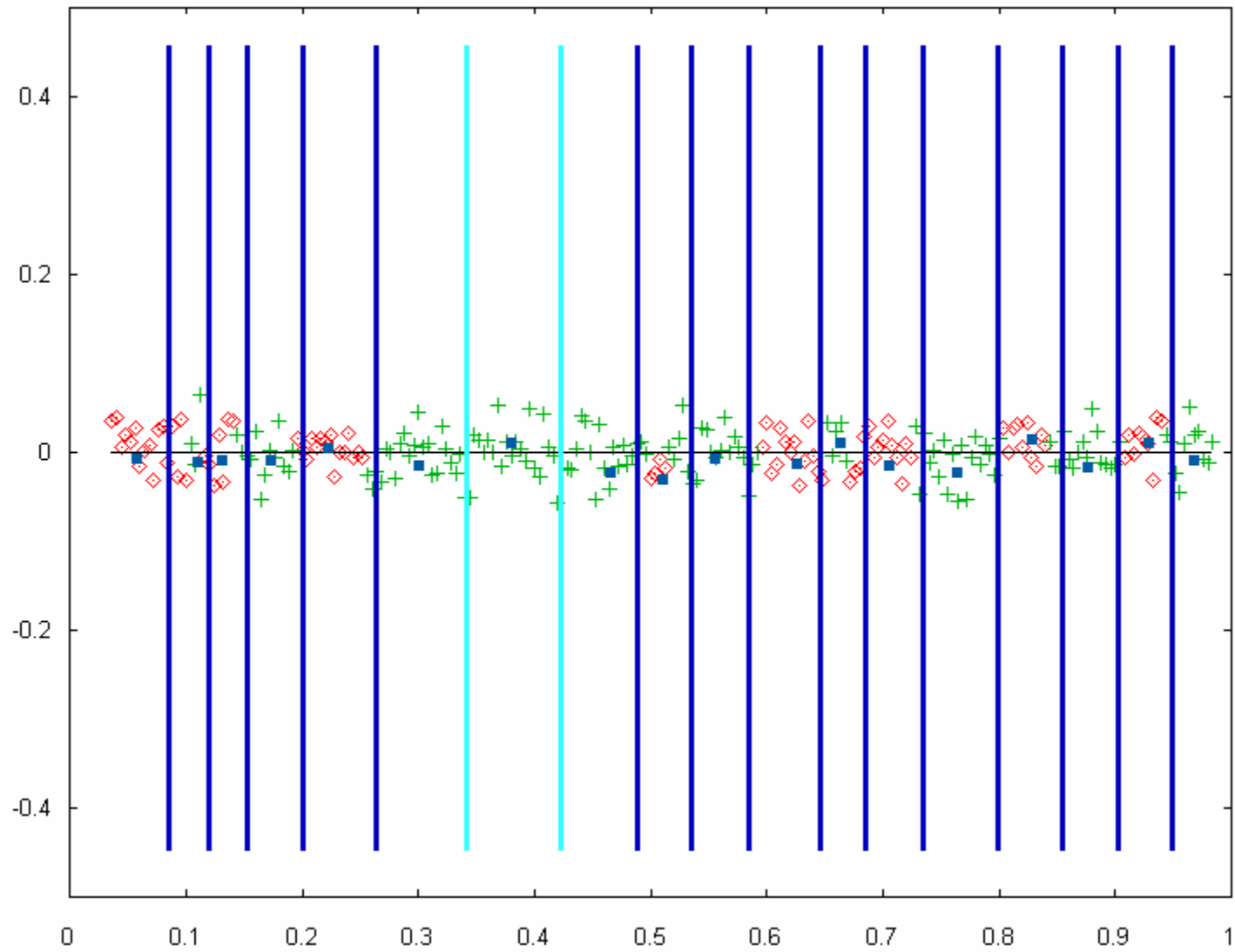


Decision tree: Toy example - 2 classes in 1.23D



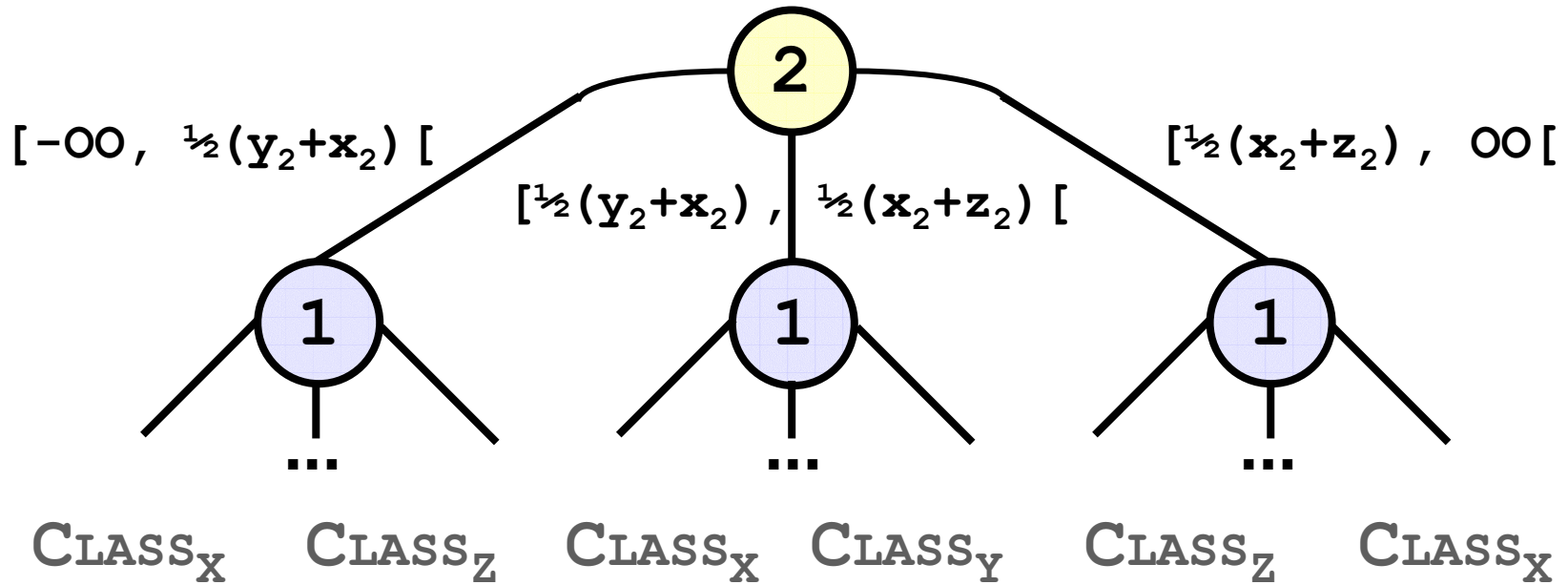


Decision tree: Toy example - 2 classes in 1.23D





$$\begin{array}{l} \text{prototypes} \left\langle \begin{array}{l} \mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N, \text{CLASS}_X) \\ \mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N, \text{CLASS}_Y) \\ \mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N, \text{CLASS}_Z) \end{array} \right. \\ \text{relevances} - \lambda = (0.3, 0.4, \dots, 0.1) \end{array}$$

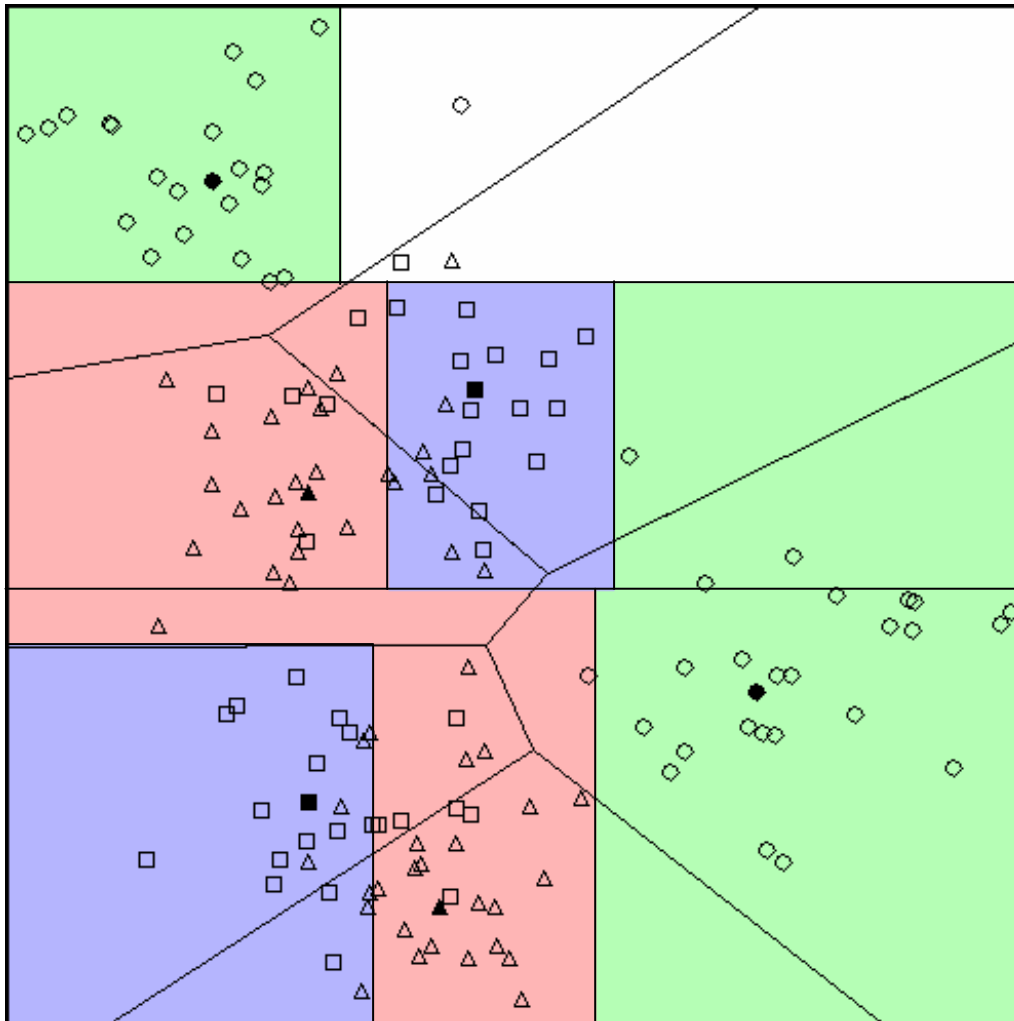




BB-tree-algorithm (BabsiBaum 😊)

Java Code by Andreas Rechten, LNM.

Again: 10-D artificial data with 3 classes.



BB-Splits and Receptive Fields

Correct Labels for $BB_{h=2}$: 79.5%

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



BB-Tree (X, W, Λ) :

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

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

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

compile a sorted list $[a_1, \dots, a_W]$ from $\{w_{I^N}^i \mid w^i \in W\}$

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

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

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



Classical decision tree methods:

CART (Breiman, Friedmann, Olshen, Stone, 1983)

C4.5 (Quinlan, 1993)

CAL5 (Müller and Wysotzki, 1997)

Are greedy approaches using divide and conquer methods with information measures or statistic properties for the tree construction.

BB: *Simultaneous* adaptation of relevances.



SIG*, UMM (Ultsch, 1991)

rule extraction from unsupervised SOM;
post-labelling in a 2D or 3D target map.

BB: Active class separation and
no a priori dimensionality reduction.

Literature for feed forward networks, e.g. Duch et al.



Demonstration: Convergence of relevances.

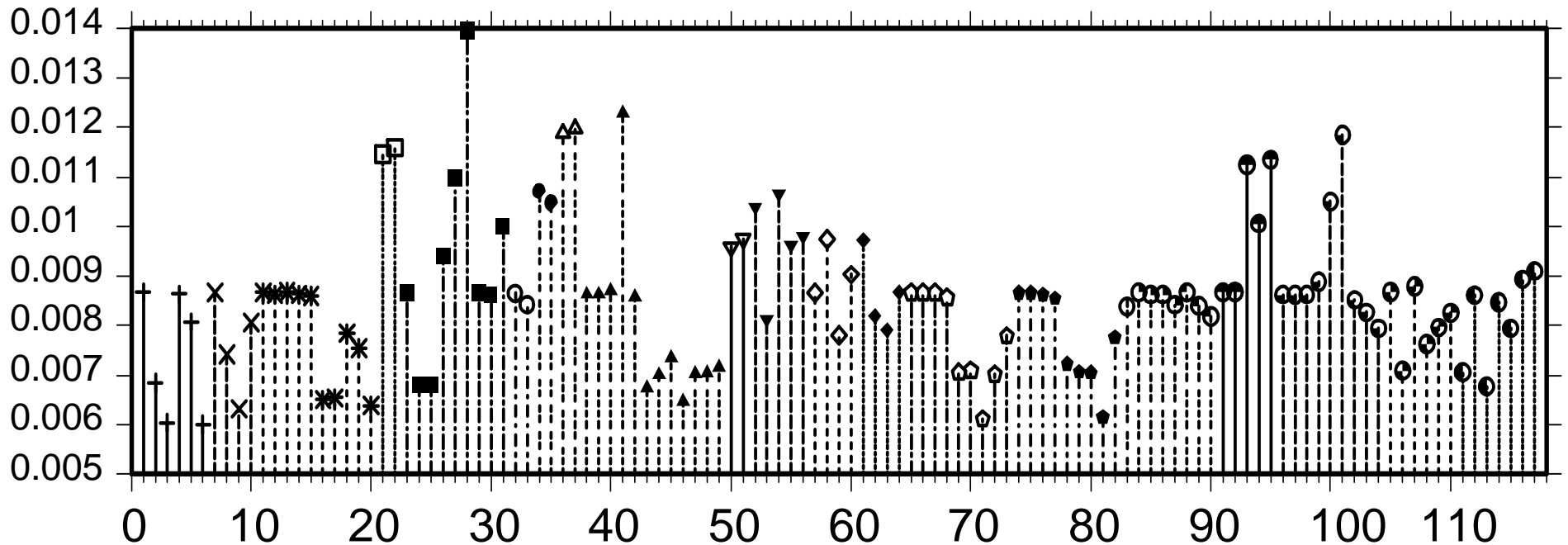
Data

Features: 117 unary coded dimensions.

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

Patterns: 8124.

Classification accuracy: 97.5% on test set.



Top 6 most relevant features:

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



Mushroom data: Extracted decision table

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 % on training set.

odor = none => minimum conflict

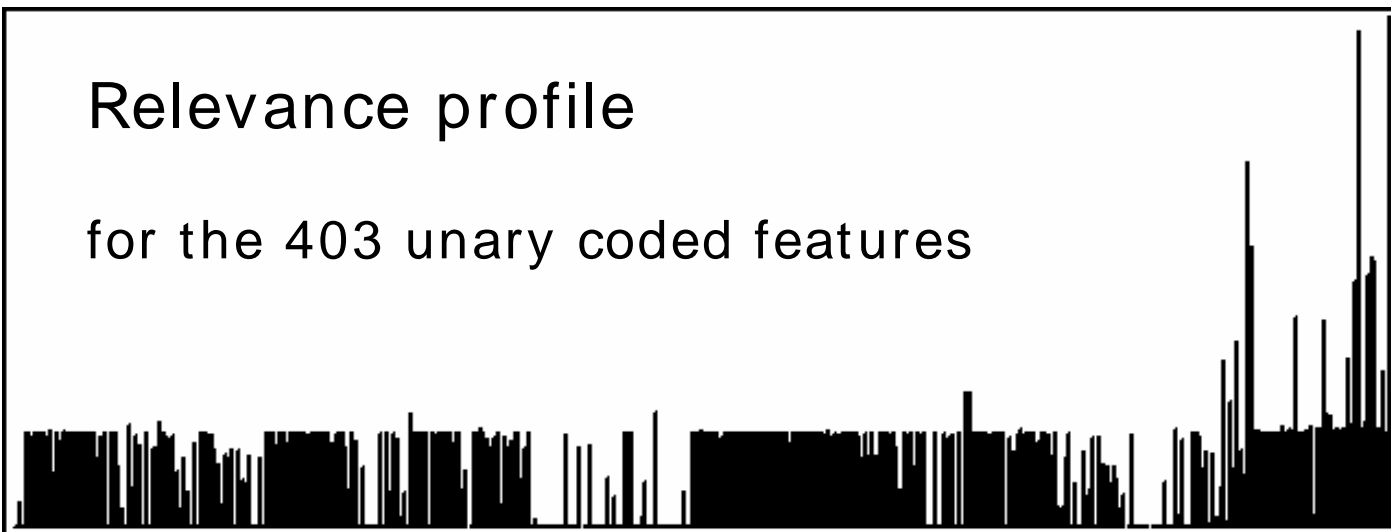


Which syllable information is most relevant for the prediction of one of five diminutive forms in Dutch ?

+ , = , A , s , - , f , A , d , - , b , a , n , T
= , = , = , = , + , r , e , = , - , G , @ , n , T
- , k , o , = , - , l , O , m , + , b , K , n , T
+ , h , o , = , - , n , I , N , - , b , A , k , J

Relevance profile

for the 403 unary coded features





Comparison to other methods

	training set (2999)	test set (950)
SRNG :	92.6 % ,	92.3 %
BB _{h=25} :	95.5 % ,	95.2 % (117 rules)
TiMBL :		96.6 %
C4.5 :	97.5 % ,	97.1 % (71 rules)



Improving the rules

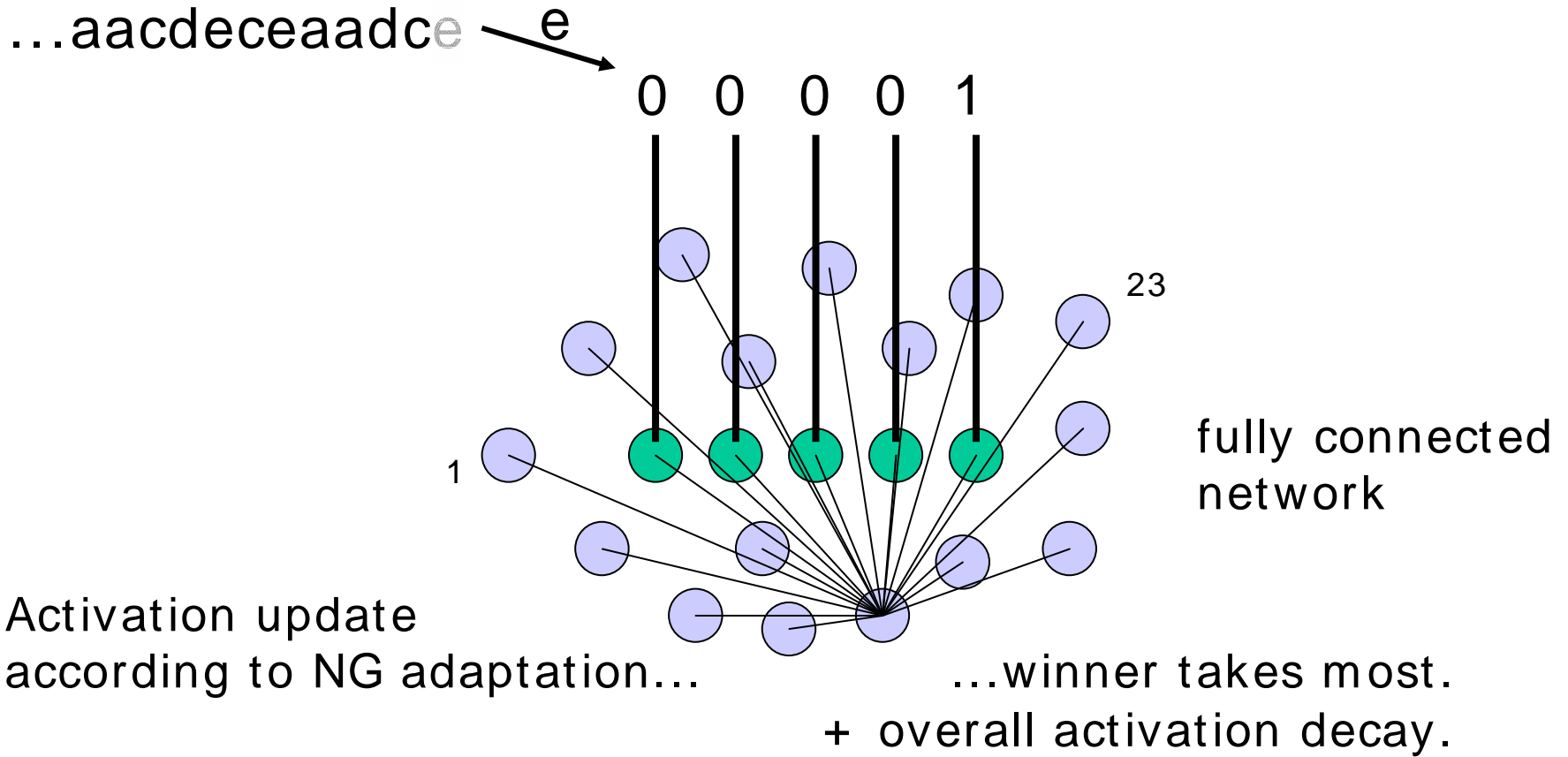
1. *Decision boundaries*: ret(r)ain interior node prototypes.
2. *Rule simplification*: hypercube rearrangements; merges.
3. *Hybrid model*:

INTERIOR NODES- > characteristic rules,
logical AND- chains in BB tree: symbolic;

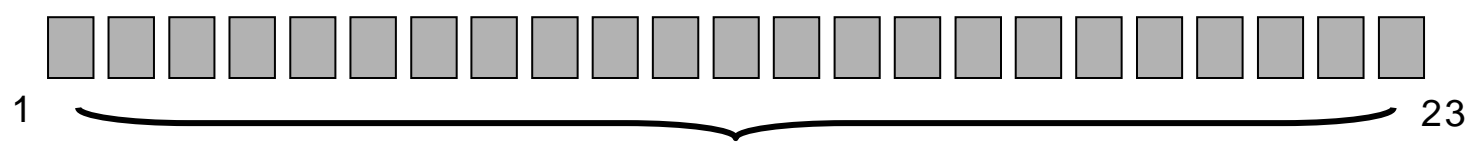
LEAVES- > differentiating analogies,
prototype majority vote: subsymbolic.



- Vector quantization methods have been extended to:
 1. stable learning by gradient descent on cost function,
 2. adaptive metric, making dimension relevances available.
- From trained net, i.e. prototypes and metric, a BB – decision tree, thus, rules can be extracted.
- Good results for both methods; sometimes data transforms like logarithmic or z-transform are necessary.
- Many ideas to still improve the results 😊



Context vector containing real valued activations



e.g. fed into vector quantizer



Ongoing work for tree processing SOMs.

Hammer, Micheli, Sperduti (2001):

A general framework for processing structured data.



(Prevents policemen from being squeezed :o)



That's it for today.

Thanks for attention !

Questions??