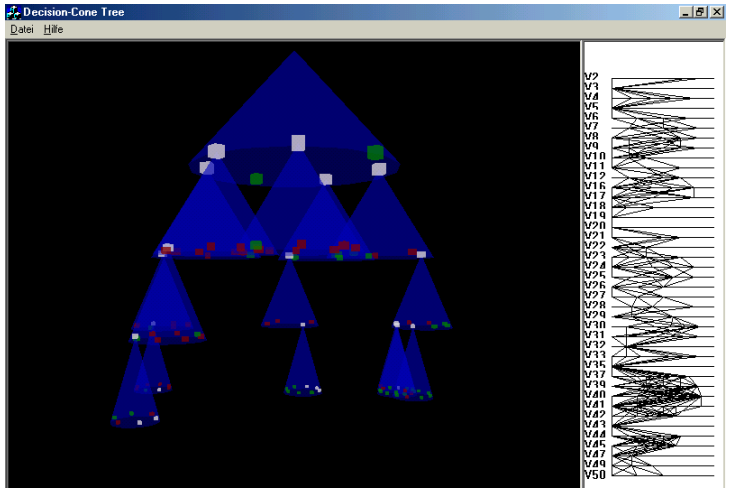
Association Rule Mining & Information Visualization



Univ.-Lektor Dr.techn. Alexander K. Seewald
Österreichisches Forschungsinstitut
für Artificial Intelligence

alex@seewald.at / alex.seewald.at

Association Rule Mining

- Find rules of the form $A \Rightarrow B$ where A and B are itemsets, (i.e. sets of *items*) and $A \cap B = \emptyset$.

- An item is a specific object which may either be part of an itemset or not.
- Rules can related to any attribute; there is no target variable (*unsupervised* as in Reinforcement Learning)

- Usually applied to binary data (*market basket analysis*); Nominal variables can be transformed in the usual way into one binary attribute per value. Each binary attribute corresponds to an item with $attr_i = 1 \Leftrightarrow$ Item *i* is present.

alex@seewald.at / alex.seewald.at

Association Rule Mining: Defs.

- T is the set of all possible items. When $A,B \subseteq T$, $A \cap B = \emptyset$: $A \Rightarrow B$ is an association rule.

  E.g. supermarket: Each distinct product is an item $\in$ T. Each transaction Tr_i corresponds to the products bought by one shopper, and thus to a specific itemset. TD = $\{Tr_i\}$.

- Rules of the form $A \Rightarrow B$ tell us about correlations between itemsets, e.g. *(milk,beer)* $\Rightarrow$ *(diapers, babyfood)*
- Rules are characterized by *Support* (how common is the rule?), and *Confidence* (how well does rule $A \Rightarrow B$ hold?)
- Algorithms to find **all** rules with given minimum *Support* and *Confidence* exist, and are space- and time-efficient.

alex@seewald.at / alex.seewald.at

Support & Confidence

$Support(A \Rightarrow B) = P(A \cup B)$ [$P(A)$ for Borgelt's Implementation]

$$Confidence(A \Rightarrow B) = \frac{P(A \cup B)}{P(A)}$$

$$P(I) = \frac{1}{|TD|} \sum_{Tr \in TD} \prod_{i \in I} (Tr_i = 1) \qquad Lift(A \Rightarrow B) = \frac{P(A \cup B)}{P(A)P(B)}$$

Itemset I is frequent $\Leftrightarrow$ *Support(I)* $\geq$ *minimum support*

Adding items: Support can only monotonically decrease ($P(I \cup X) \leq P(I)$), since we add restrictions to the itemset.

$\Rightarrow$If an itemset *I* is frequent, all its subsets must also be frequent. If any subset has lower support, then *I* cannot be be frequent. This is an efficient pruning criterion

alex@seewald.at / alex.seewald.at

The Apriori Algorithm

Computes the set of all frequent itemsets.

L_1 = {set of frequent items (=frequent itemsets of size 1)}
for (k=1; L_k!=Ø; k++) {
  C_{k+1} = {A∪B | A,B ∈ L_k, |A∪B|=k+1,
    ∀ X⊂(A∪B) ⇒ X∈(∪L_k)}
  L_{k+1} = {C | C ∈ C_{k+1} && *Support(C) > MinSupp* }
}
return ∪ L_k;

Afterwards, compute all possible rules (partitions) of frequent itemsets and output those with min. confidence.

Open-source Apriori implementation by C. Borgelt
http://fuzzy.cs.uni-magdeburg.de/~borgelt/apriori.html

alex@seewald.at / alex.seewald.at

Rules from Weather Dataset (nominal)

Minimum Support = 0.3, *Minimum Confidence* = 0.7

- **outlook=overcast ⇒ play=yes**
  supp: 0.28, conf: 1.0, lift: 1.56
- **temperature=cool ⇒ humidity=normal**
  supp: 0.28, conf: 1.0, lift: 2.0
- **humidity=normal, windy=false ⇒ play=yes**
  supp: 0.28, conf: 1.0, lift: 1.56
- **humidity=normal ⇒ play=yes**
  supp: 0.43, conf: 0.86, lift: 1.33
- **play=no ⇒ humidity=high**
  supp: 0.28, conf: 0.8, lift: 1.6

...

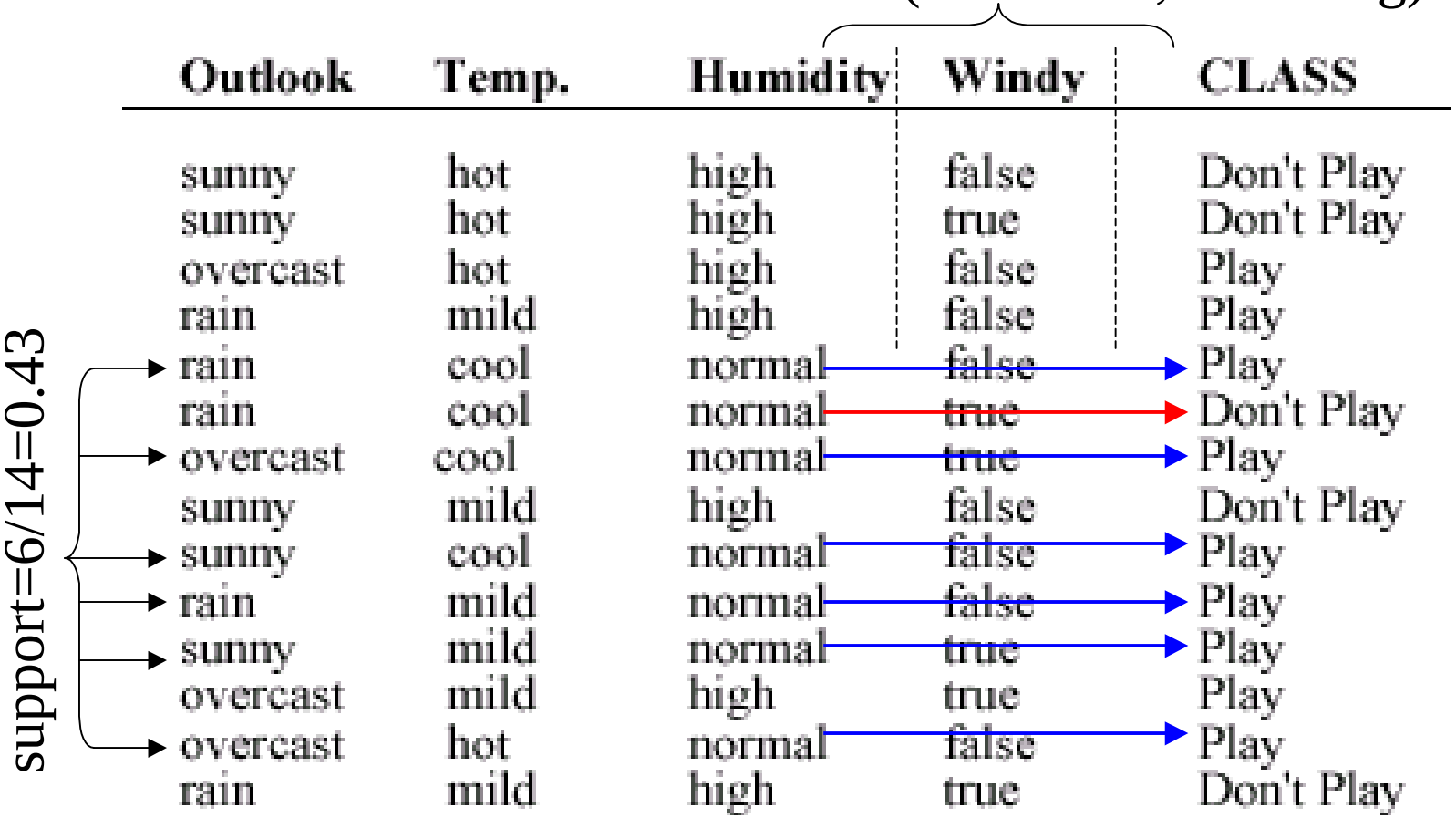
For large datasets, outputs a large set of rules (>1000), so understanding rules is more challenging than mining.

alex@seewald.at / alex.seewald.at

Ex.: Computing Support and Confidence

- **humidity=normal ⇒ play=yes**

confidence=6/7=0.86 (6x correct, 1x wrong)



Outlook	Temp.	Humidity	Windy	CLASS
sunny	hot	high	false	Don't Play
sunny	hot	high	true	Don't Play
overcast	hot	high	false	Play
rain	mild	high	false	Play
rain	cool	normal	false	Play
rain	cool	normal	true	Don't Play
overcast	cool	normal	true	Play
sunny	mild	high	false	Don't Play
sunny	cool	normal	false	Play
rain	mild	normal	false	Play
sunny	mild	normal	true	Play
overcast	mild	high	true	Play
overcast	hot	normal	false	Play
rain	mild	high	true	Don't Play

alex@seewald.at / alex.seewald.at

Information Visualization

No inherent spatial or temporal structure (contrary to *Scientific Visualization*). Heterogenous, high-dimensional data → use appropriate visual metaphors. Interaction with the user in context of real-time explorative data analysis offers the highest benefits.

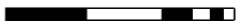
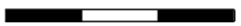
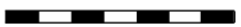
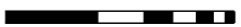
Five general techniques

- *Geometric*: e.g. scatterplots and parallel coordinates.
- *Icon-based*: e.g. chernoff faces, stick figures, glyphs.
- *Pixel-based*: e.g. recursive patterns, circle segments.
- *Hierarchical*: e.g. cone/cam trees and treemaps.
- *Graph-based*: e.g. polylines and curved lines.

alex@seewald.at / alex.seewald.at

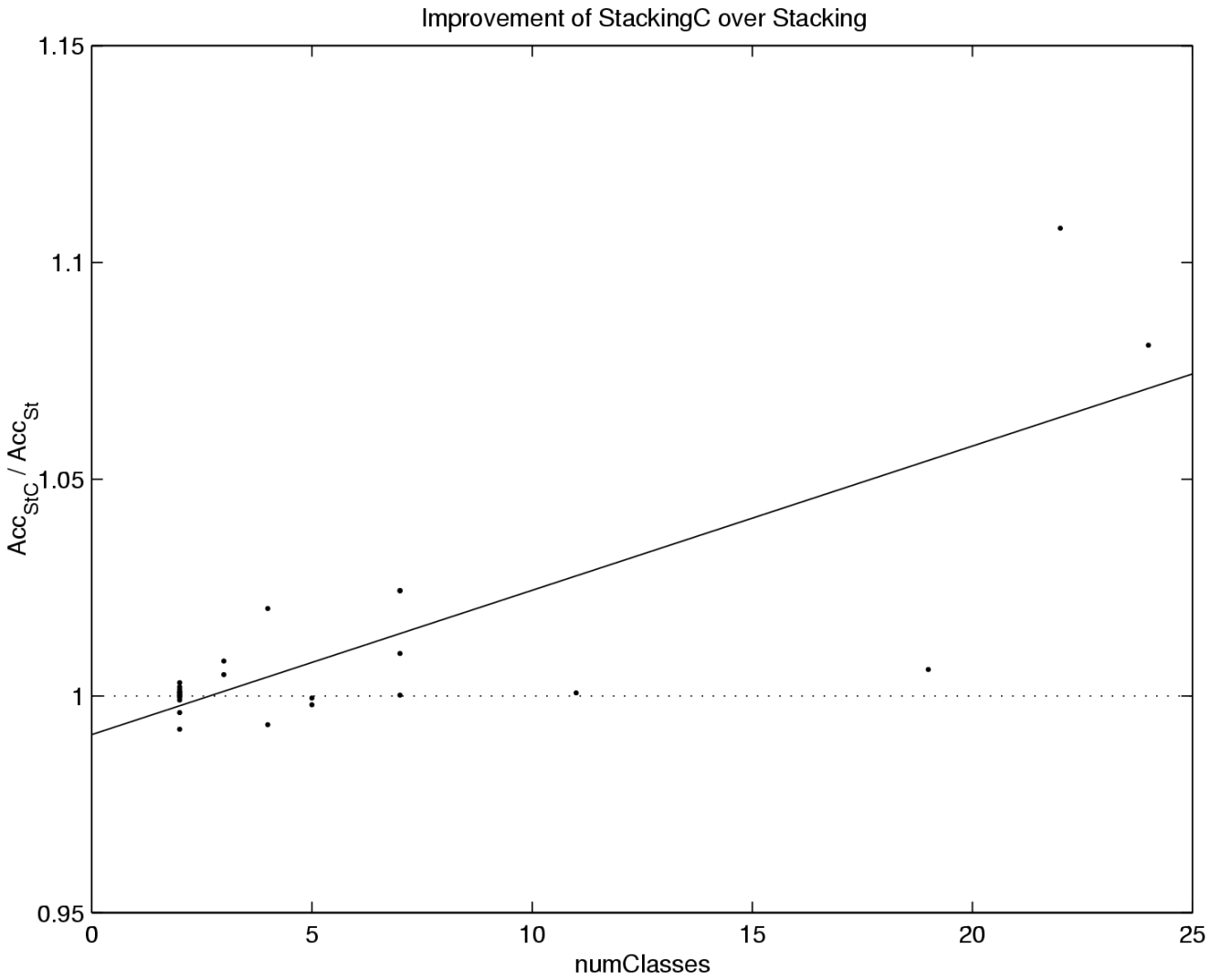
Example: Visualizing Class Distributions

Table 8.1: The used datasets with number of classes and examples, discrete and continuous attributes, baseline accuracy (%) and entropy in bits per example (Kononenko & Bratko, 1991). Class frequencies are shown in descending order; the length of every black and white bar determines one class frequency each.

Dataset	cl	Inst	disc	cont	bL	E	class frequencies
audiology	24	226	69	0	25.22	3.51	
autos	7	205	10	16	32.68	2.29	
balance-scale	3	625	0	4	45.76	1.32	
breast-cancer	2	286	10	0	70.28	0.88	
breast-w	2	699	0	9	65.52	0.93	
colic	2	368	16	7	63.04	0.95	
credit-a	2	690	9	6	55.51	0.99	
credit-g	2	1000	13	7	70.00	0.88	
diabetes	2	768	0	8	65.10	0.93	
glass	7	214	0	9	35.51	2.19	
heart-c	5	303	7	6	54.46	1.01	
heart-h	5	294	7	6	63.95	0.96	
heart-statlog	2	270	0	13	55.56	0.99	
hepatitis	2	155	13	6	79.35	0.74	
ionosphere	2	351	0	34	64.10	0.94	
iris	3	150	0	4	33.33	1.58	
labor	2	57	8	8	64.91	0.94	
lymph	4	148	15	3	54.73	1.24	
primary-t.	22	339	17	0	24.78	3.68	
segment	7	2310	0	19	14.29	2.81	
sonar	2	208	0	60	53.37	1.00	
soybean	19	683	35	0	13.47	3.84	
vehicle	4	846	0	18	25.41	2.00	
vote	2	435	16	0	61.38	0.96	
vowel	11	990	3	10	9.09	3.46	
zoo	7	101	16	2	40.59	2.41	

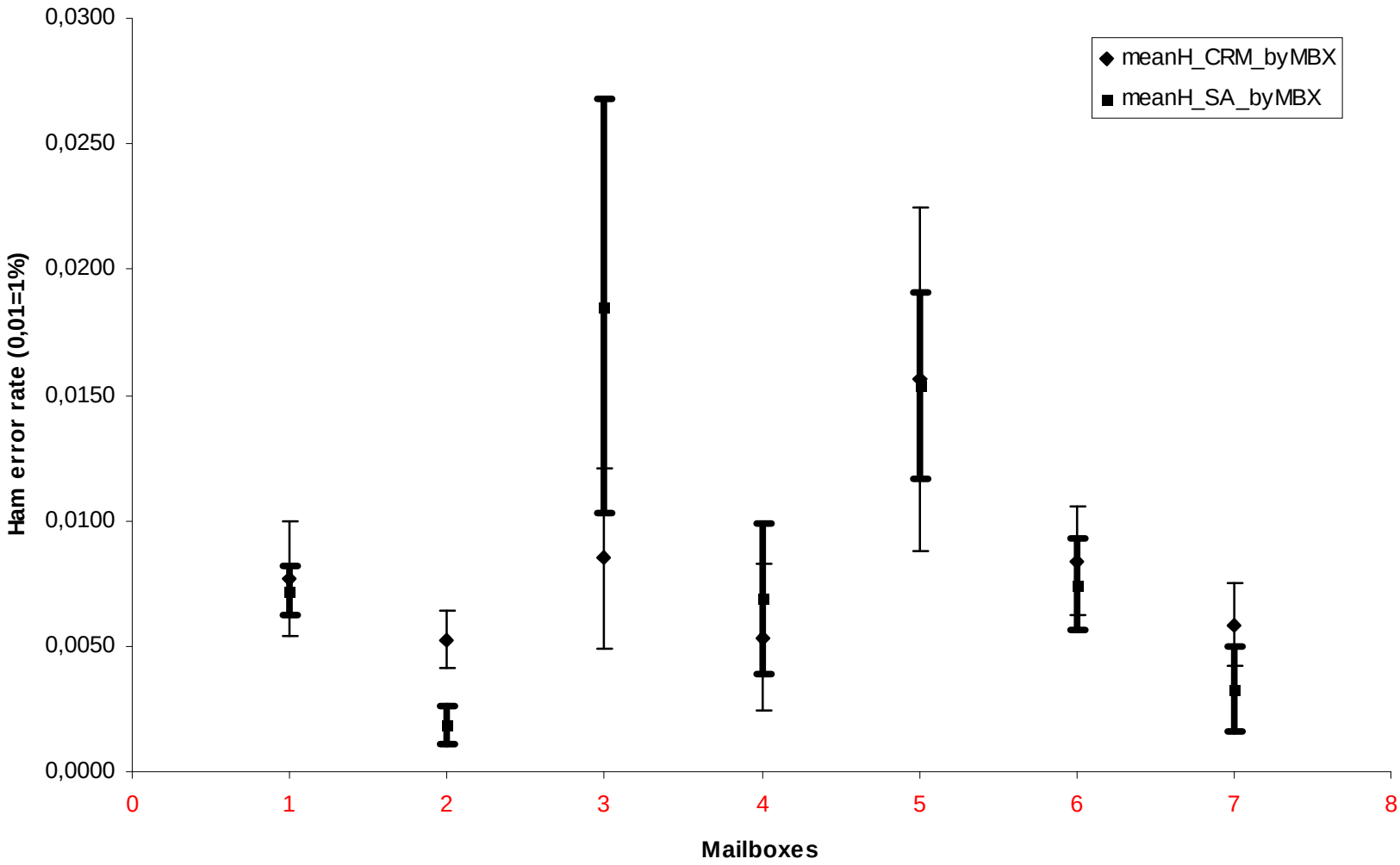
alex@seewald.at / alex.seewald.at

Example (1): Scatterplots



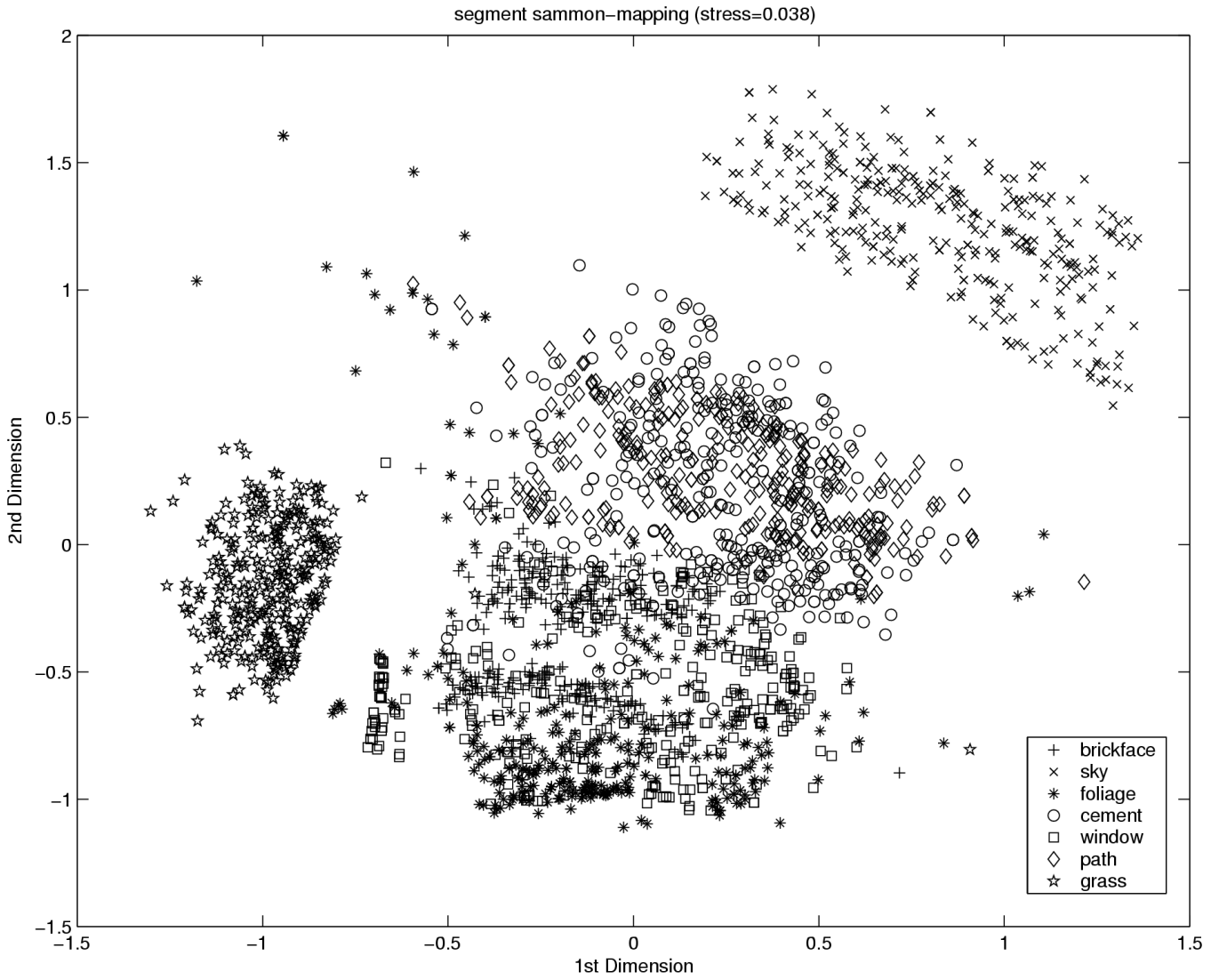
alex@seewald.at / alex.seewald.at

Example (2): Confidence Intervals



alex@seewald.at / alex.seewald.at

Example (3): Sammon-Mapping



alex@seewald.at / alex.seewald.at

Example (4): Glyph-based Visualization

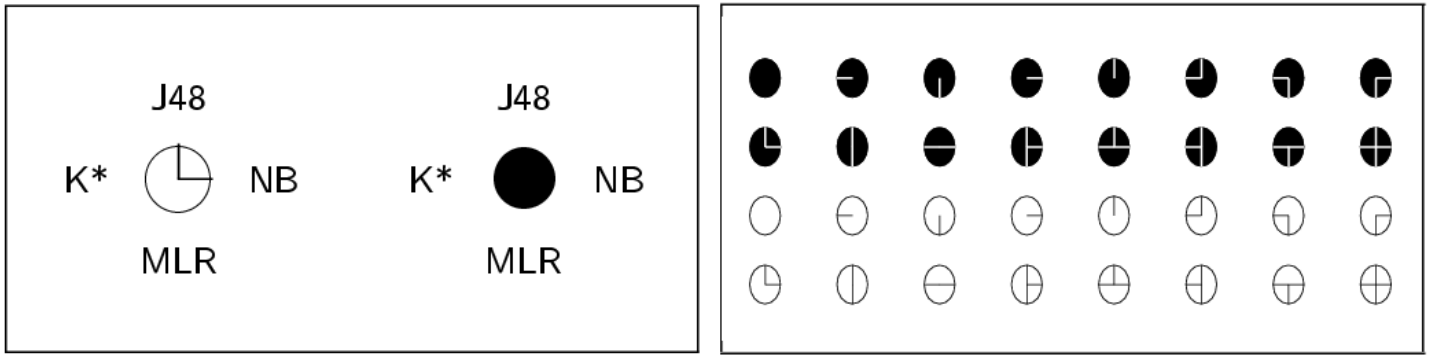
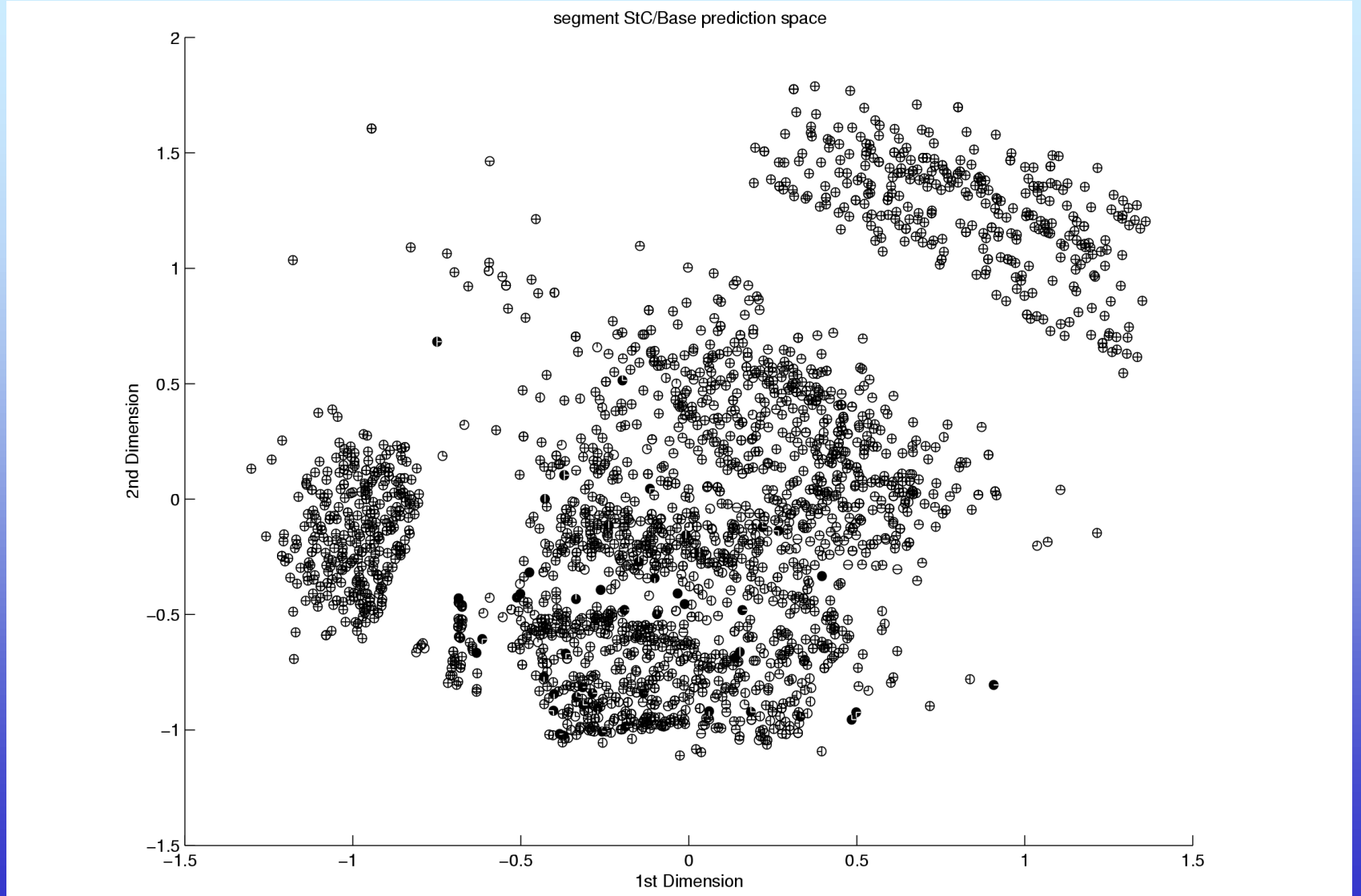


Figure 8.3: On the left side of the figure, we see two example glyphs. The main form is a circle. A filled circle indicates that StackingC predicted the wrong class; an empty circle indicates otherwise. Each direction (up, right, down and left) corresponds to a base classifier (J48, NaiveBayes, MLR and KStar) as shown, where the presence of a line indicates a correct prediction. The left glyph sample thus encodes an example where NaiveBayes, J48 and StackingC were correct; while the right glyph sample encodes an example where all base classifiers were incorrect, including StackingC. If one of the base classifiers were correct in this case, we would see a white line from the center of the filled circle into the appropriate direction, analogous to the black line for the empty circle. On the right side of this figure we see all thirty-two possible glyphs. The two upper rows correspond to instances where StackingC predicted the class wrong; the other two rows to correct predictions. The same order of glyphs is used later.

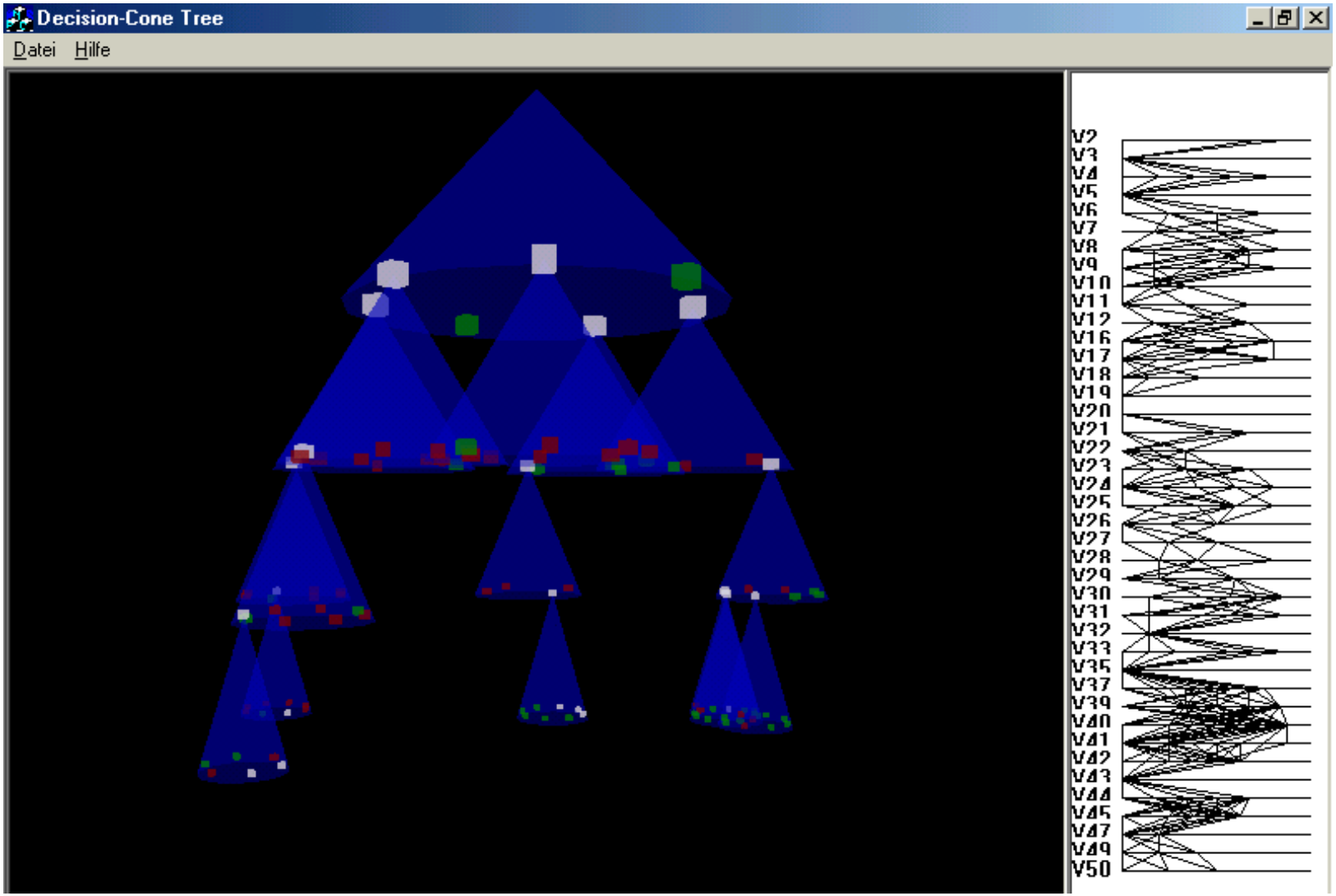
alex@seewald.at / alex.seewald.at

Example (5): Glyphs + Sammon Mapping



alex@seewald.at / alex.seewald.at

Example (6): Cone Decision Trees



alex@seewald.at / alex.seewald.at

Things We Did Not Talk About and Why

Clustering

- Groups data into clusters. Unsupervised, i.e. no class information available.
- No single field: Contains a variety different subproblems.

Neural Networks

- Unstable learning scheme, difficult to master and control.
- In most practical application, learns an almost linear model(!)
- If interested: *Neural Computation* = 2+1h lecture by Prof. Dorffner

Genetic Algorithms

- Relies on specific problem structure to work well ~ experimental technique.
- Very high computational effort, but no guarantee of useful solution (compare with *Dynamic Programming* which guarantees a global optimum solution; or *Monte Carlo Methods* which rely on random sampling in solution space)

Hands-on Machine Learning / Data Mining tasks

- Topic of next-year's lecture *AI Methoden der Datenanalyse* (hopefully...)
- We will create our own training data for digit recognition et al.

alex@seewald.at / alex.seewald.at