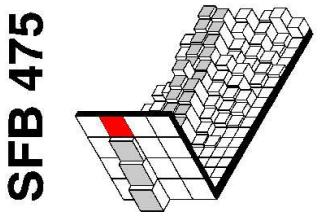
klaR: A Package Including Various Classification Tools

Christian Röver, Nils Raabe, Karsten Luebke and Uwe Ligges



Universität Dortmund
44221 Dortmund
Germany

May 21, 2004

Overview:

1. Example data
2. Classification tools
3. Comparing classification results
4. Variable selection
5. Illustrating discrimination
6. Visualization of data structure

B3 data: "West German business cycles"

- data on 14 economic variables observed quarterly over 39 years (157 observations)

- each quarter was assigned to one out of 4 phases:

  1. upswing
  2. upper turning point
  3. downswing
  4. lower turning point

- wanted: classification rule for phases

RDA: Regularized Discriminant Analysis[1]

- **generalization** of LDA and QDA
- assumptions similar to QDA
  (differences in means and covariances)
- **covariance matrices** are manipulated using two parameters (γ and λ)
- more robust against **multicollinearity**
- parameters are determined by minimizing (estimated) misclassification rate

[1] Friedman, J.H. (1989): Regularized Discriminant Analysis. *Journal of the American Statistical Association* 84, 165-175.

RDA: special cases

- (γ=0, λ=0): **QDA** — individual covariances for each group.
- (γ=0, λ=1): **LDA** — a common covariance matrix.
- (γ=1, λ=0): **Conditional independence**, identical variances within class (similar to Naive Bayes).
- (γ=1, λ=1): Objects are assigned to class with **nearest mean** (euclidean).

RDA: examples

- set parameters manually...

```
> x <- rda(PHASEN~., data=B3[train,], gamma=0.05, lambda=0.1)
```

- ...or optimize misclassification rate.

```
> x <- rda(PHASEN~., data=B3[train,])
```

- prediction etc. as usual

```
> predict(x, B3[test,])
$class
 [1] 3 3 3 4 4 4 4 1 3 1 1 1 1 1 1 1 4 4 4 1 1 4 4 4 1 1
```

SVMlight[2]

- interface to T. Joachims' Support Vector Machine implementation
- supports **loss parameters** and **1-against-all** classification
- returns comparable **membership scores** ('posterior probabilities')
- example:

```
> x <- svmlight(PHASEN ~ ., data=B3[train,])
> predict(x, B3[test,])
```

[2] Joachims, T. (2004): SVM^{light}. `http://svmlight.joachims.org/`

Comparing classifications

- looking at **misclassifications**:

```
> errormatrix(true.phase, rda.prediction)
       predicted
true     dn ltp up utp -SUM-
  dn      2   7  0   0     7
  ltp     2   4  0   0     2
  up      1  12 14   0    13
  utp     0   5  0   1     5
  -SUM-   3  24  0   0    27
```

- 27 out of 48 are misclassified, worst rates for (true) "utp", most misclassifications go into class "ltp",. . .

Comparing classifications

- looking at **posterior assignments**:

```
$posterior
          up   utp    dn   ltp
 [1,] 0.000 0.000 0.978 0.022
 [2,] 0.001 0.000 0.995 0.005
 [3,] 0.077 0.000 0.151 0.772
 [4,] 0.249 0.000 0.000 0.750
 [5,] 0.256 0.000 0.005 0.739
```

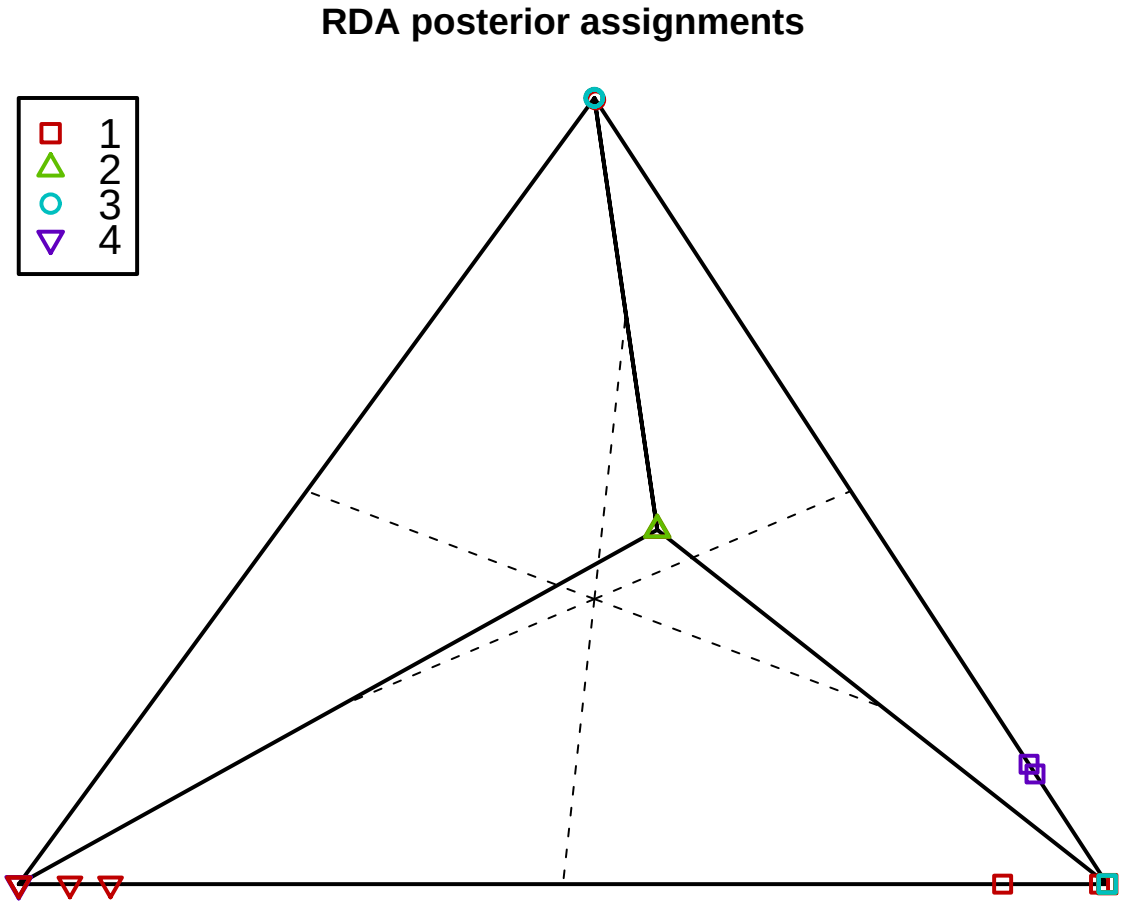
  each observation is assigned to every class with a certain **posterior probability** or **membership**

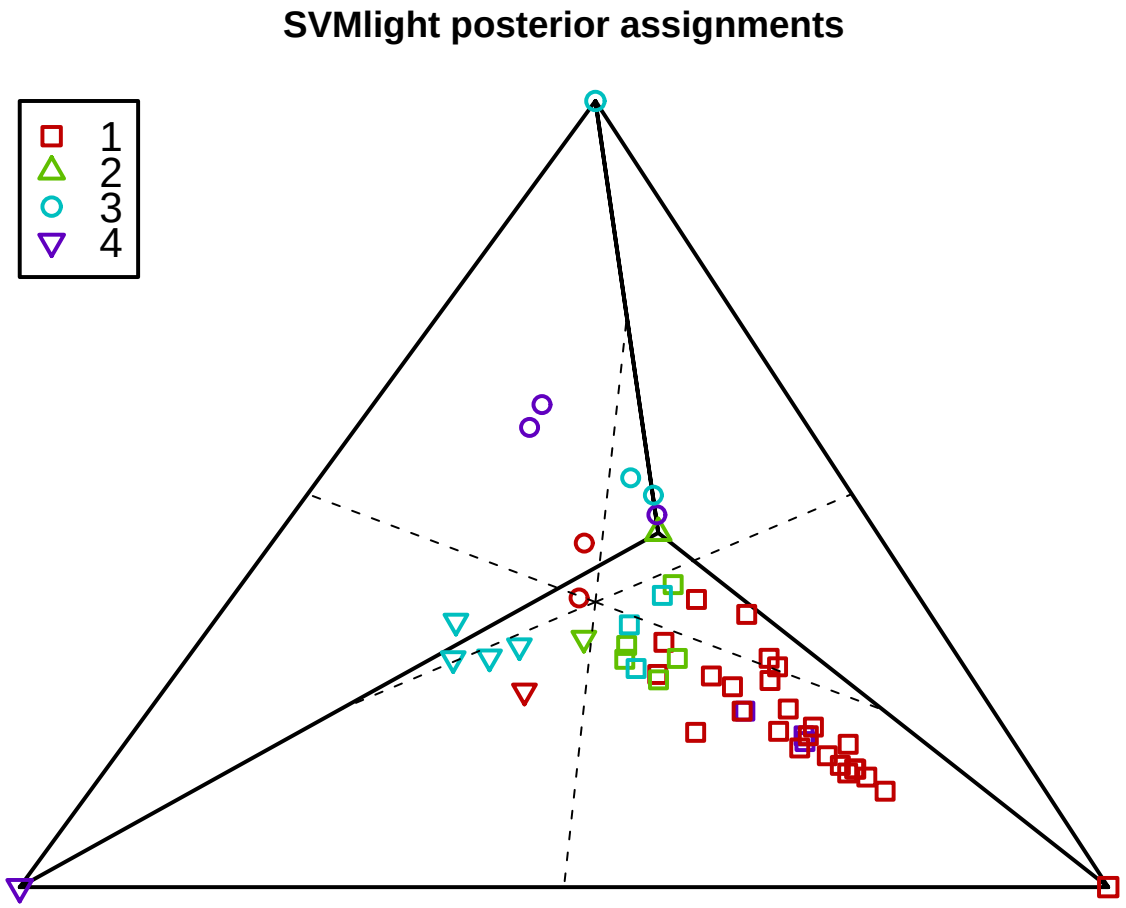
Comparing classifications

- probability distribution over 4 classes may be illustrated by a point in a **3-dimensional simplex** (tetraeder, '**barycentric plot**'):
  - each **corner** corresponds to one class,
  - probability for certain class proportional to **distance to opposite side**

- example:

```
> quadplot(rdapred$posterior, [...] )
```


RDA posterior assignments
1
2
3
4


SVMlight posterior assignments
1
2
3
4

Comparing classifications

- **RDA**: greater posterior probabilities
  (points on edges and corners)

- **SVMlight**: more uncertainty
  (points inside simplex)

➜ measure these features for comparison

Comparing classifications

- derive[3]
  - **Correctness rate**: 1 - error rate
  - **Accuracy**: distance to 'true' corner
  - **Ability to separate**: distance to classified corner
  - **Confidence**: mean membership of assigned class (either by class or average)

[3] Garczarek, U. and Weihs, C. (2003): Standardizing the Comparison of Partitions. *Computational Statistics* 18, 143-162.

```
> ucpm(m=rdapred$posterior, tc=B3$PHASEN[test])
$CR
[1] 0.5833333

$AC
[1] 0.3250307

$AS
[1] 0.981954

$CF
[1] 0.9889456

$CFvec
        1         2         3         4
0.9912088 1.0000000 0.9999684 0.9511723
```

Comparing classifications

	LDA	**RDA**	**SVM**
Correctness rate (1 - error rate)	0.44	0.58	0.54
Accuracy (distance to *true* corner)	0.03	0.33	0.17
Ability to separate (distance to *classified* corner)	0.75	0.98	0.29
Confidence (mean membership of assigned class)	0.83	0.99	0.47

Variable selection

- `stepclass`: stepwise selection using (estimated) misclassification rate
  - *forward selection*: add variables to model
  - *backward selection*: throw variables out
  - or *both directions*

- works for most classification methods

Variable selection

- example:

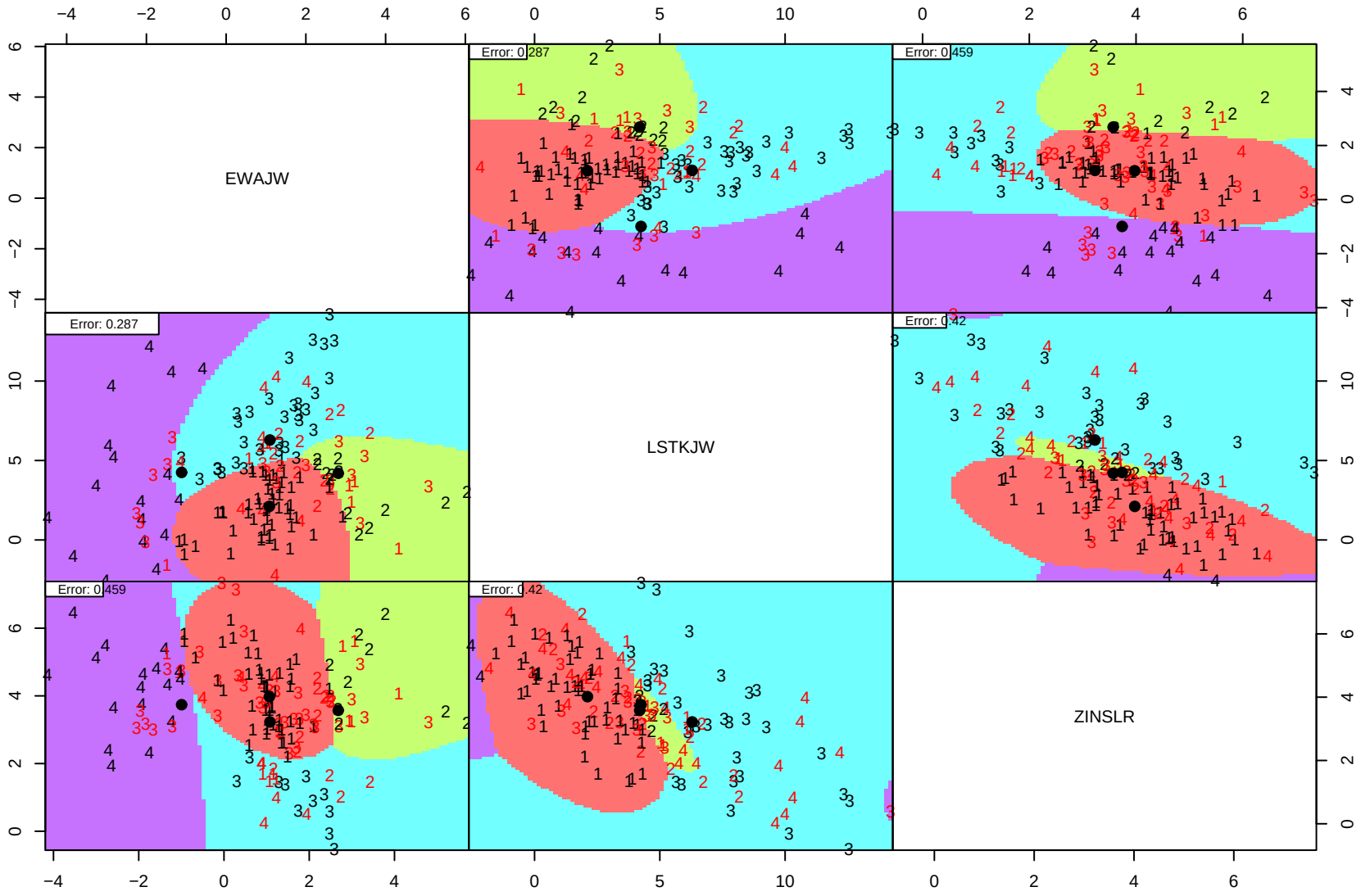
```
> x <- stepclass(PHASEN~., data=B3[train,],
+                method="qda", prior=rep(1/4,4))
> x
 method       : qda
 final model  : EWAJW, LSTKJW, ZINSLR
 error rate   : 0.3265
```

- error rate for test set is 29% (71% correct)

Visualization of partitionings

- how are classes located / separated?
- look at **partitioning** for every pair of variables...

```
> partimat(B3[,x$model$name], B3[,"PHASEN"],
+          method="qda", plot.matrix=TRUE)
```

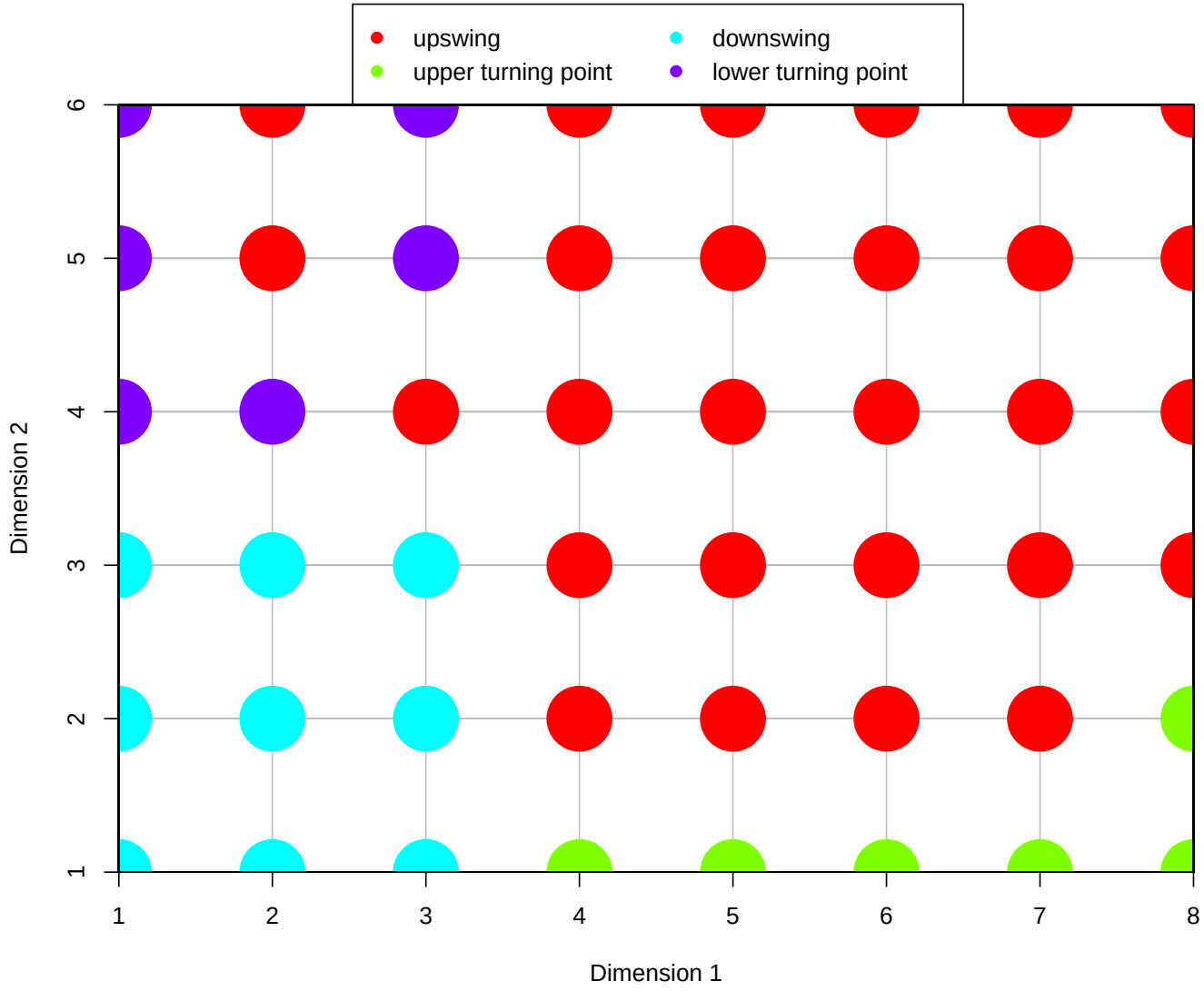

EWAJW
Error: 0.287
Error: 0.459
Error: 0.287
LSTKJW
Error: 0.42
Error: 0.459
Error: 0.42
ZINSLR

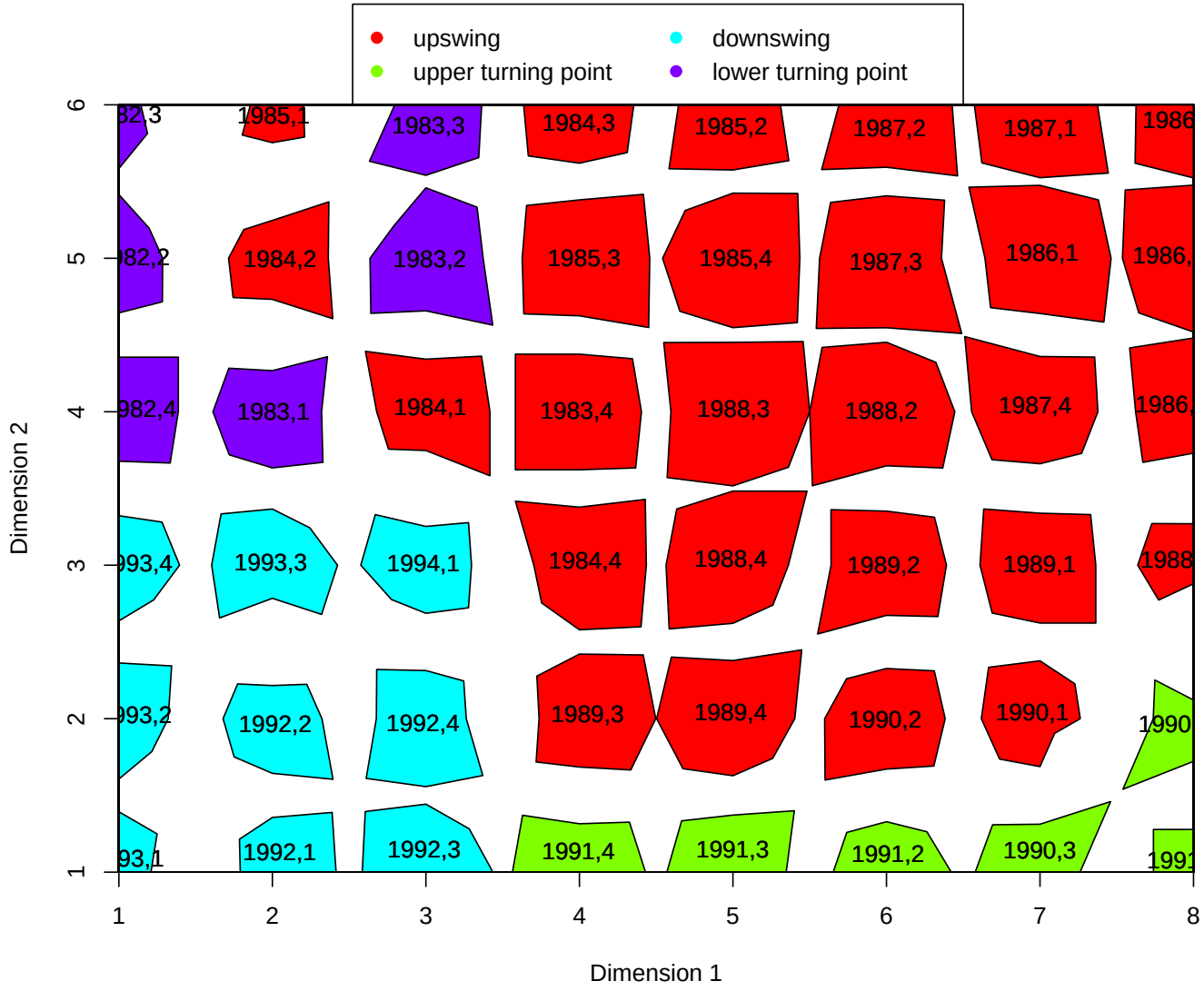
Visualization of data structure

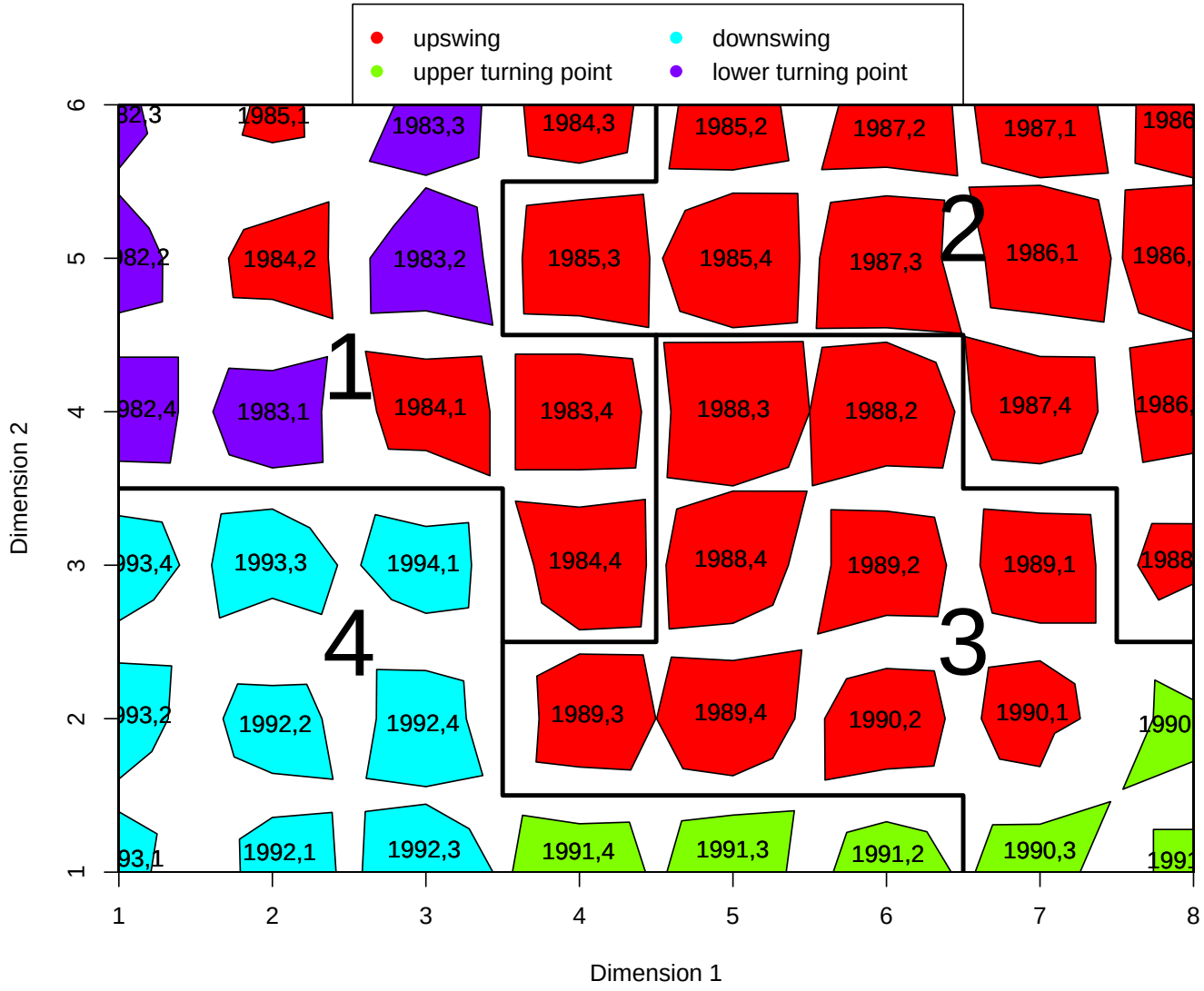
- `EDAM`: Eight Directions Arranged Map[4]
- similar to Self Organizing Maps
- observations (and gaps between) are arranged on a **2D-grid** in order to **reflect distances**
- example:

```
> lcEDAM <- EDAM(B3[test,-1], classes=B3$PHASEN[test],
+                  standardize = TRUE, iter.max = 20)
```

[4] Raabe, N. (2003). *Vergleich von Kohonen Self-Organizing-Maps mit einem nichtsimultanen Klassifikations- und Visualisierungsverfahren*. Diploma Thesis, University of Dortmund.


upswing
downswing
upper turning point
lower turning point
Dimension 2
Dimension 1
1
2
3
4
5
6
7
8


upswing
downswing
upper turning point
lower turning point
Dimension 2
Dimension 1
1
2
3
4
5
6
7
8
1985,1
1983,3
1984,3
1985,2
1987,2
1987,1
1984,2
1983,2
1985,3
1985,4
1987,3
1986,1
1983,1
1984,1
1983,4
1988,3
1988,2
1987,4
1993,3
1994,1
1984,4
1988,4
1989,2
1989,1
1992,2
1992,4
1989,3
1989,4
1990,2
1990,1
1992,1
1992,3
1991,4
1991,3
1991,2
1990,3


upswing
downswing
upper turning point
lower turning point
Dimension 2
Dimension 1
1
2
3
4
1985,1
1983,3
1984,3
1985,2
1987,2
1987,1
1984,2
1983,2
1985,3
1985,4
1987,3
1986,1
1983,1
1984,1
1983,4
1988,3
1988,2
1987,4
1993,3
1994,1
1984,4
1988,4
1989,2
1989,1
1992,2
1992,4
1989,3
1989,4
1990,2
1990,1
1992,1
1992,3
1991,4
1991,3
1991,2
1990,3

`> require(klaR)`

- seen:
  - Classification tools: `rda`, `svmlight`
  - Comparing classifications: `errormatrix`, `ucpm`
  - 3D barycentric plots: `quadplot`
  - Variable selection: `stepclass`
  - Illustrating classifications: `partimat`
  - Data visualization: `EDAM`
- further features:
  - 2D barycentric plots: `triplot`
  - Hidden Markov Modelling: `hmm.sop`
  - Simple k-Nearest Neighbour: `sknn`