UNIVERSITÄT LEIPZIG

Eine Einführung in R: Hochdimensionale Daten: $n << p$ Teil II

Bernd Klaus, Verena Zuber
Institut für Medizinische Informatik, Statistik und Epidemiologie (IMISE),
Universität Leipzig

19. Januar 2012

1. Fragestellung: Supervised vs Unsupervised

2. Klassifikation: Diskriminanzanalyse
   - Fragestellung
   - Vorgehensweise
   - Weitere Klasifikatoren

3. Ähnlichkeitsmaße und Clustern
   - Ähnlichkeitsmaße
   - Clustern
   - Visualisierung

Grundsätzliche Fragestellung

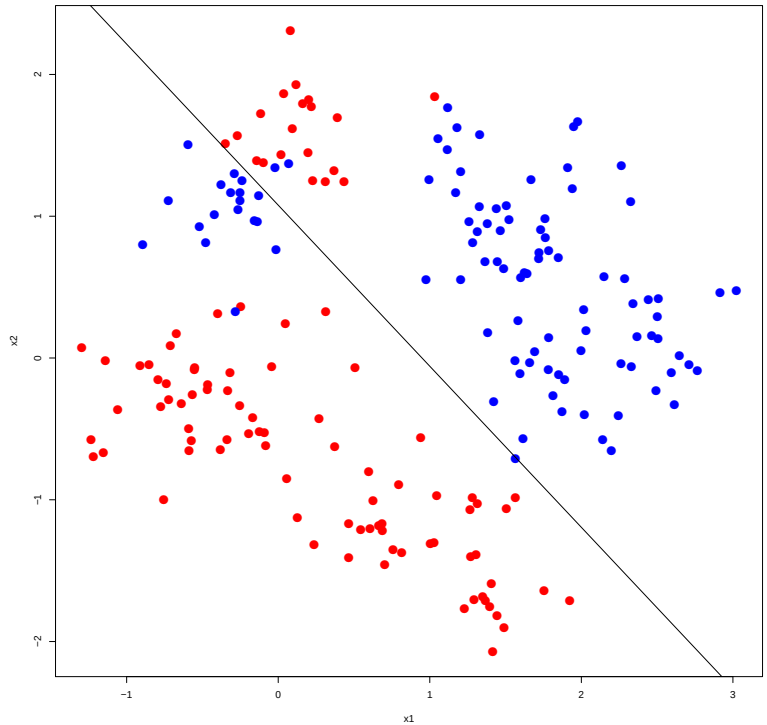
- *Supervised*:
  Mit Daten X soll eine interessierende Variable Y erklärt werden.
  Beispiele:
  - Y kategorial: Klassifikation / Diskriminanzanalyse
  - Y metrisch: Lineares Modell
- *Unsupervised*:
  Welche Struktur findet sich in den Daten X?
  Eine interessierende Variable Y **soll nicht** (oder erst in der weiteren Analyse) untersucht werden.
  Beispiele:
  - Clusterverfahren
  - Netzwerke
  - Principal Component Analysis (PCA)

Klassifikation

- Fragestellung: Erstellen einer Vorhersageregel auf Basis bestehender (Trainings-)Daten
- Daten:
  - Y: kategoriale Zielgröße, interessierende Variable; im folgenden $Y = 0, 1$
  - X: metrische Prediktoren, erklärende Variablen (mehr/hoch-dimensional)
- Ziel:
  1. Variablenselektion: Finde Variablen in X die Y möglichst gut vorhersagen können, bzw möglichst gut zwischen den Gruppen in Y trennen/diskriminieren.
  2. Vorhersageregel: Mittels einer Vorhersageregel kann zu gegebenen X-Daten Y vorhergesagt werden.
- Methoden: Diskriminanzanalyse, Logit-Regression

Darstellung in zwei Dimensionen

- $Y = 0, 1$ in blau und rot dargestellt
- (Lineare) Entscheidungsgrenze (*Decision Boundary*), um die Gruppen blau/rot zu trennen



Diskriminanzanalyse: Vorgehensweise

1. Teilen des Datensatz in Trainings- und Testdaten
2. Variablenselektion auf Basis der Trainingsdaten
3. Erstellen der Vorhersageregel mit den gewählten Variablen
4. Evaluierung der Vorhersageregel an den Testdaten

1. Teilen des Datensatz in Trainings- und Testdaten

- Zufällige Aufteilung der Beobachtungen in Trainings- und Testdaten
- Ziel: Beurteilung der Vorhersageregel, die auf die Trainingsdaten angepasst wurde, auf einem unabhängigen Testdatensatz
- Gütekriterien: Misklassifikationsrate, oder andere Kombinationen aus:

	wahr: 0	wahr: 1
klassifiziert: 0	True Negatives	False Negatives
klassifiziert: 1	False Positives	True Positives

2. Variablenselektion

- Strategien:
  - Erst Variablen selektieren, z.B. mit Mittelwertsdifferenz oder t-score, dann auf den selektierten Variablen die Vorhersageregel berechnen
  - Regularisierte Verfahren verwenden, die automatisch Variablenselektion betreiben.
  - Vorsicht: Optimal wäre ein weiterer unabhängiger Validierungsdatensatz, um die Variablen zu selektieren, bzw. Parameter der Regularisierung zu wählen.

3. Erstellen der Vorhersageregel

Die Form der Diskriminanzanalyse ist abhängig von der Kovarianz von X in den Gruppen 0 und 1:

- Diagonal: $\Sigma_0 = \Sigma_1 = I$
- Linear: $\Sigma_0 = \Sigma_1 \neq I$
- Quadratisch: $\Sigma_0 \neq \Sigma_1$

Vorsicht: In der quadratischen müssen zwei Kovarianzmatrizen der Dimension $p \times p$ berechnet werden. Deswegen wird bei großem p meist nur diagonal oder linear gerechnet.

R-paket: `MASS`

```
lda(formula, data, ..., subset, na.action)
qda(formula, data, ..., subset, na.action)
```

4. Vorhersage auf Testdaten

R-paket: `MASS`

```
predict.lda(object, newdata, ...)
predict.qda(object, newdata, ...)
```

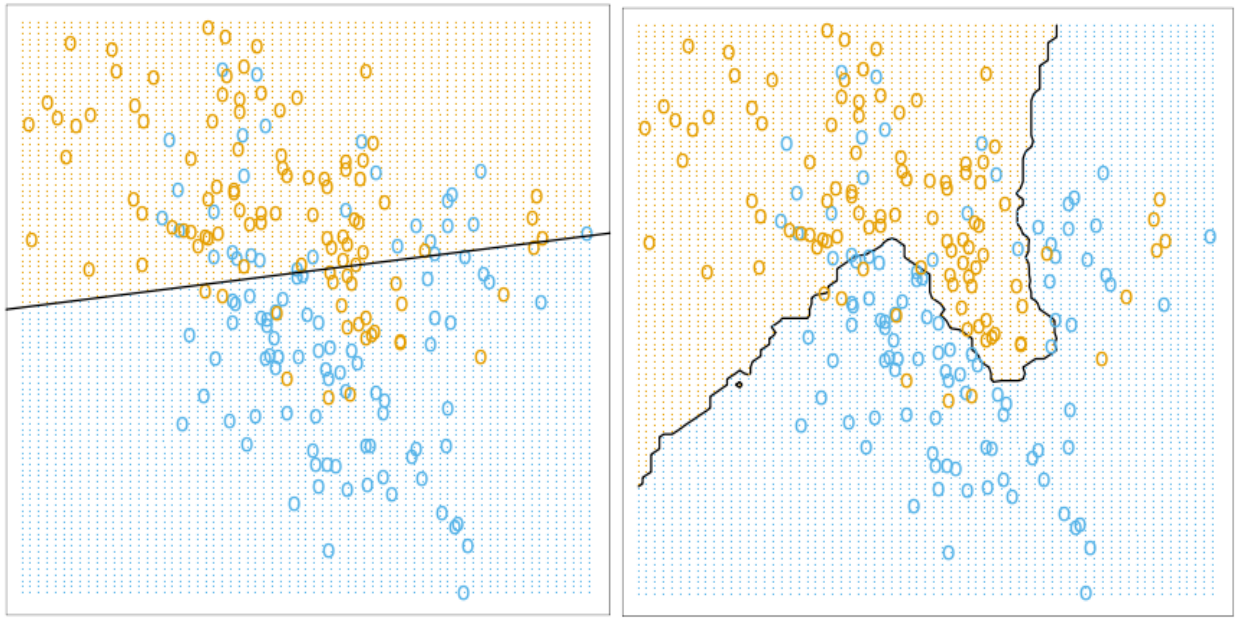
- `object`: Ein Objekt vom Typ `lda, qda`
- `newdata`: Neue X-Daten, ohne Y

Ausgabe:

- `$class`: Prognose bezüglich der Klasse
- `$posterior`: (Posteriori) Wahrscheinlichkeiten für die Zugehörigkeit zu einer Klasse

Weitere Verfahren zur Klassifikation

- *Decision Trees* bzw *Random Forest*
- *Neural Network*
- *Support Vector Machines*
- *Nearest Neighbor*



links: LDA; rechts: Nearest Neighbor Klassifikation

1. Ähnlichkeitsmaße: Wann sind Variablen ähnlich?

Ähnlichkeitsmaße

Wie kann Ähnlichkeit quantifiziert werden?

Dafür eignen sich Distanzmaße, wie:

- **Euklidische Norm:**

$$\|x\| = \sqrt{x_1^2 + x_2^2 + ... + x_p^2}$$

- **Absolute (Manhatten) Norm:**

$$\|x\|_1 = |x_1| + |x_2| + ... + |x_p|$$

- p-**Norm:**

$$\|x\|_p = \left(x_1^p + x_2^p + ... + x_p^p\right)^{1/p}$$

Clustern

Ein Cluster ist eine Gruppe von Variablen oder Beobachtungen.

Ziel des Clusterns

- Varianz in einem Cluster **so gering** wie möglich
- Varianz zwischen Clustern **so stark** wie möglich

Beispiel: Genexpressionsdaten

- Bei genetischen Prozessen ist häufig nicht ein Gen alleine beteiligt, sondern ein Netzwerk, das in komplexer Weise interagieren kann -> Gennetzwerke
- Finden sich die AML und ALL Beobachtungen aus den Golub Daten in identischen Clustern wieder oder sind diese bunt gemischt?

Cluster-Algorithmen: Single Linkage

- **Start:** Jedes der zu clusternden Objekte bildet ein eigenes Cluster
- **Repeat:** Verbinden der Cluster, die am ähnlichsten sind
- **Stop:** Ein großes Cluster
- **Ergebnis**: Baumstruktur, Dendrogramm

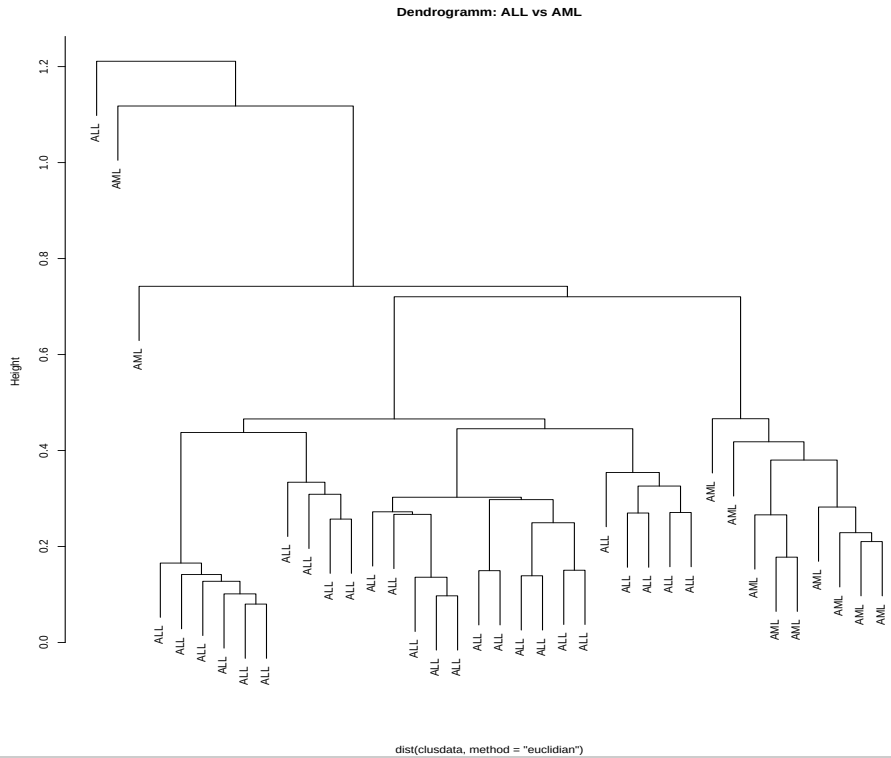
Vorteil: Anzahl der Gruppen muss nicht a priori festgelegt werden.

R-Befehl:

```
plot(hclust(dist(data,method="euclidian"),
method="single"), main="Dendrogramm: ALL vs AML",
                    labels=gol.fac)
```

Cluster-Algorithmen: Single Linkage

- Expression von $p = 2$ Genen **"CCND3 Cyclin D3"** und **"Zyxin"**
- Frage: Können die $n = 38$ Beobachtungen **getrennt nach Läukemie-Typen** (ALL vs AML) geclustert werden?



Cluster-Algorithmen: k-means

- **Start:** Beginne mit k (zufälligen) Clustern und berechne die k-Mittelwerte
- **Repeat:** Assoziere jede Variable **neu mit dem Mittelwert, zu dem sie am nächsten liegt**. $\Rightarrow$ neue Mittelwerte
- **Stop:** Wenn sich die k Mittelwerte nicht mehr stark verändern
- **Ergebnis:** k Cluster

Nachteil: Anzahl der Gruppen muss a priori festgelegt werden.

R-Befehl:

```
cl <- kmeans(data, centers=2, nstart = 10)
```

Cluster-Algorithmen: k-means

- Expression von $p = 2$ Genen **"CCND3 Cyclin D3"** und **"Zyxin"**
- Frage: Können die $n = 38$ Beobachtungen **getrennt nach Läukemie-Typen** (ALL vs AML) geclustert werden?

```
cluster.2 <- kmeans(data, centers=2, nstart = 10)
```

- `K-means clustering with 2 clusters of sizes 11, 27`
- `Cluster means:`

	CCND3 Cyclin D3	Zyxin
1	0.6355909	1.5866682
2	1.8938826	-0.2947926

- `Clustering vector: 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1`
- `Within cluster sum of squares by cluster: 4.733248 19.842225 (between SS / total SS = 62.0 %)`

Eine Heatmap stellt die Expression aller p Gene aller n Beobachtungen dar. Die Spalten (und/oder Zeilen) werden nach ihrerer Ähnlichkeit angeordnet. Zusätzlich wird an den Spalten (und/oder Zeilen) ein Dendrogramm geplotet.

R-Befehl: `heatmap( ))` nur bedingt empfehlenswert

- `heatmap(x, Rowv=NULL, Colv=NULL,))`
- `x`: Datenmatrix
- `Rowv=NA`: Option falls Ordnen der Zeilen nicht erwünscht
- `Colv=NA`: Option falls Ordnen der Spalten nicht erwünscht

Weitere Pakete:

- heatmap.plus mit Befehl `heatmap.plus( )`
- gplot mit Befehl `heatmap.2( )`
- compHclust mit Befehl `compHclust.heatmap( )`

Heatmap der Golub-Daten