4. Zufallszahlen

- werden nach einem determinist. Algorithmus erzeugt $\Rightarrow$ Pseudozufallszahlen

- wirken wie zufäll. Zahlen (sollen sie jedenfalls)

Algorithmus:

Startwert x_0 $\quad x_{n+1} = f(x_n)$ (z.B. Kongruenzen)

Der Generator von SAS

$$x_{n+1} = \underbrace{397204094}_{2 \cdot 7 \cdot 7 \cdot 4053103} x_n \text{mod}(2^{31} - 1)$$

liefert gleichverteilte ganze Zufallszahlen auf $(0, 2^{31} - 1)$.

Zufallszahlen

auf $(0,1)$ gleichverteilte Zufallsgrössen, $U_n \sim R(0,1)$

$$U_n = \frac{x_n}{2^{31}-1}$$

```
seed = -1;  /* zufaelliger Startwert */
x=ranuni(seed) /*auf (0,1) gleichvert.ZZ. */
x=rannor(seed) /*Standard-Normal-ZZ.*/
```

Der interne Startwert wird dann durch x_1 ersetzt, der folgende Aufruf von rannor(seed) liefert eine neue Zufallszahl.

Zufallszahlen

vorgegebene stetige Verteilung

wird z.B. aus gleichverteilter Zufallsvariable U_i mittels Quantilfunktion ($F^{-1}(U_i)$) gewonnen.

diskrete Verteilungen

werden erzeugt durch Klasseneinteilung des Intervalls $(0, 1)$ entsprechend der vorgegebenen Wahrscheinlichkeiten p_i, also

$$(0, p_1], (p_1, p_1 + p_2], (p_1 + p_2, p_1 + p_2 + p_3],$$

$$\ldots, (p_1 + \cdots + p_{k-1}, 1)$$

Zufallszahlen

Wünschenswerte Eigenschaften

- Einfacher Algorithmus, wenig Rechenzeit.
- möglichst viele verschieden Zufallszahlen sollen erzeugbar sein

$\Rightarrow$ lange Periode.

- k-Tupel $(U_1, \ldots, U_k) \sim R(0,1)^k$, $k \leq 10$

$\Rightarrow$ Test auf Gleichverteilung.

- “Unabhängigkeit”

Test auf Autokorrelation (z.B. Durbin-Watson Test, vgl. Regression)
Plot der Punkte (U_i, U_{i+k}), $k = 1, 2...$
es sollten keine Muster zu erkennen sein.

`Zufallszahlen_test.sas`
`Zufallszahlen_Dichte.sas`

Clusteranalyse

Ziel: Zusammenfassung von
- “ähnlichen” Objekten zu Gruppen (Clustern),
- unähnliche Objekte in verschiedene Cluster.

Cluster sind vorher nicht bekannt.

20 Patienten, Blutanalyse

Merkmale: Eisengehalt X_1, alkalische Phosphate X_2

Umweltverschmutzung in verschiedenen Städten

Merkmale: Schwebeteilchen, Schwefeldioxid

Byzantinische Münzen

Lassen sich gesammelte Münzen verschiedenen Epochen zuordnen?

Clusteranalyse

Wir unterscheiden:

partitionierende Clusteranalyse

Zahl der Cluster ist vorgegeben
(MAXCLUSTERS=)
PROC FASTCLUS (k-means),
PROC MODECLUS (nichtparam. Dichteschätzung)

hierarchische Clusteranalyse

PROC CLUSTER, gefolgt von
PROC TREE und evtl.
PROC GPLOT

Fuzzy Clusteranalyse

Clusteranalyse

Abstandsdefinitionen (p: # Merkmale)

Euklidischer Abstand (das ist Standard)

$$d_E^2(x,y) = \sum_{i=1}^{p} (x_i - y_i)^2$$

City-Block Abstand (Manhattan-Abstand)

$$d_C(x,y) = \sum_{i=1}^{p} |x_i - y_i|$$

Tschebyscheff-Abstand

$$d_T(x,y) = \max_i |x_i - y_i|$$

Clusteranalyse

- Nichteuklidische Abstände müssen selbst berechnet werden.

Macro %DISTANCE

- Abstandsmatrix kann in der DATA-Anweisung angegeben werden.

DATA=name (TYPE=DISTANCE)

- Die Variablen sollten i.A. vor der Analyse standardisiert werden, da Variablen mit großer Varianz sonst großen Einfluß haben (Option STANDARD oder die Prozedur ACECLUS zuvor laufen lassen).

davor: Ausreißer beseitigen.

Hierarchische Clusteranalyse

Methoden (1)

Die Methoden unterscheiden sich durch die Definition der Abstände $D(C_i, C_j)$ zwischen Clustern C_i und C_j.

Single Linkage

$$D_S(C_i, C_j) = \min\{d(k,l), k \in C_i, l \in C_j)\}$$

Complete Linkage

$$D_C(C_i, C_j) = \max\{d(k,l), k \in C_i, l \in C_j)\}$$

Centroid

$D_{CE}(C_i, C_j) = d(\overline{X}_i, \overline{X}_j)$ Abstände der Schwerpunkte

Hierarchische Clusteranalyse

Methoden (2)

Average Linkage

$D_A(C_i, C_j) = \frac{1}{n_i n_j} \sum_{k \in C_i, j \in C_j} d(k, l)$

Ward

ANOVA-Abstände innerhalb der Cluster minimieren, außerhalb maximieren. Nach Umrechnen erhält man
$D_W(C_i, C_j) = \frac{n_i n_j}{n_i + n_j} D_{CE}(C_i, C_j)$.

Density Linkage

beruht auf nichtparametrischer Dichteschätzung (DENSITY, TWOSTAGE)

Hierarchische Clusteranalyse

Tendenzen

WARD:	Cluster mit etwa gleicher Anzahl von Objekten
AVERAGE:	ballförmige Cluster
SINGLE:	große Cluster, “Ketteneffekt”, langgestreckte Cluster
COMPLETE:	kompakte, kleine Cluster

Im Mittel erweisen sich **Average Linkage** und **Ward** sowie die nichtparametrischen Methoden als die geeignetsten Methoden.

Hierarchische Clusteranalyse

Agglomerative Verfahren

1. Beginne mit der totalen Zerlegung, d.h.
   $Z = \{C_1, ..., C_n\}, C_i \cap C_j = \varnothing \quad C_i = \{O_i\}$
2. Suche C_r, C_l : $d(C_r, C_l) = \min_{i \neq j} d(C_i, C_j)$
3. Fusioniere C_r, C_l zu einem neuen Cluster:
   $C_r^{new} = C_r \cup C_l$
4. Ändere die r-te Zeile und Spalte der Distanzmatrix durch Berechnung der Abstände von C_r^{new} zu den anderen Clustern! Streiche die l-te Zeile und Spalte!
5. Beende nach n-1 Schritten, ansonsten fahre bei 2. mit geänderter Distanzmatrix fort!

Hierarchische Clusteranalyse

- Alle von SAS angebotenen hierarchischen Methoden sind agglomerativ.
- Es gibt auch divisive Methoden.
- Fall großer Datensätze:

PROC FASTCLUS: Vorclusteranalyse mit großer Anzahl von Clustern
PROC CLUSTER: mit diesen Clustern.

Hierarchische Clusteranalyse

zu WARD:

ANOVA Abstände innerhalb eines Clusters i

$$D_i = \frac{1}{n_i} \sum_{l \in C_i} d^2(O_l, \overline{X}_i)$$

Fusioniere die Cluster C_i und C_j, wenn

$$D_{CE}(C_i, C_j) - D_i - D_j \longrightarrow \min_{i,j}$$

Clusteranalyse

Durchführung

```
PROC CLUSTER /*hierarchische Clusteranalyse*/
   METHOD=methode
   STANDARD /*Standardisierung*/
   OUTREE=datei;/*Eingabedatei fuer Proc Tree*/
RUN;
PROC TREE DATA=datei
   OUT=out /*Ausgabedatei z.B.f. PROC GPLOT*/
   NCLUSTERS=nc /*Anz. Cluster*/
   COPY vars /*vars in die Ausgabedatei*/
RUN;
PROC GPLOT;
   PLOT variablen=cluster;/*Symbol-Anweis.
         vorher definieren*/
RUN;
```

Hierarchische Clusteranalyse

Die Ausgabedatei OUTTREE=

NAME	Bezeichnung der Cluster
	≥ 2 Beobachtungen: CLn
	1 Beobachtung: OBn
NCL	Anzahl der Cluster
FREQ	Anzahl der Beobachtungen im jeweiligen Cluster

n: Clusternummer (CLn) oder
Beobachtungsnummer (OBn = _N_)

```
Cluster_Air.sas
Cluster.sas
Cluster_Banknoten.sas
Cluster_Muenzen.sas
```