Seminar
Anonymisierung von Mikrodaten

Thema:
Experimentelles SAFE-Verfahren
mit Clusteringalgorithmen

Bearbeitung: Boris Wiegand (36030332), Sommersemester 2015
Seminarleiter: Prof. Dr. rer. nat. habil. Rainer Lenz, Dipl.-Vw. Emanuel Weiß

Inhaltsverzeichnis

1 Aufgabe:

Nach der Grundidee von SAFE wird eine Basisdatei derart verändert, dass in der anonymisierten Datei jeder Merkmalsträger mindestens dreimal vorkommt. Da SAFE in der ursprünglichen Form praktisch schwierig umsetzbar ist, wird ein experimenteller Ansatz mit Clusteringalgorithmen verfolgt.

2 Anforderungsbeschreibung:

Ein bereits existierendes Javaprojekt mit Verfahren zur statistischen Anonymisierung wird folgendermaßen erweitert:

Der Anwender kann in einem Swing-GUI eine Basisdatei im CSV Format einlesen. Zur Vorbereitung des Anonymisierungsverfahrens wählt er aus, welchen Typ die Spalten haben (metrisch oder kategorial). Ein naives Scannen des ersten Datensatzes könnte diesen Schritt teilweise automatisieren.

Das „Cluster-SAFE-Verfahren“ soll auf folgende Weise funktionieren:

> Zuerst wird nach den kategorialen Merkmalen mit verschiedenen Verfahren geclustert.
> Die Datensätze der einzelnen Cluster werden durch ein oder verschiedene Verfahren (es wird zunächst nur ein Verfahren implementiert, die Austauschbarkeit wird aber im Codedesign berücksichtigt) auf gleiche Merkmalsausprägung gebracht. Datensätze in Cluster der Größe eins und zwei werden verworfen.
>
> Analog werden die Originaldaten nach den metrischen Merkmalen geclustert und auf gleiche Merkmalsausprägung innerhalb der Gruppen gebracht.
>
> Analog werden die Originaldaten nach allen Merkmalen geclustert und auf gleiche Merkmalsausprägung innerhalb der Gruppen gebracht.
>
> Die anonymisierte Lösung, die nach der Defintion von SAFE am nächsten an die Originaldatei herankommt, wird dem Benutzer als anonymisierte Datei vorgeschlagen.

Die im Verfahren verwendeten Clusteringverfahren können durch den Anwender an- und abgeschaltet werden (bei sehr großen Datensätzen aufgrund der Performance eventuell notwendig) und parametrisiert werden. Auch kann pro Verfahren ausgewählt werden, welcher der drei Durchgänge (kategorial, metrisch, kategorial + metrisch) ausgeführt wird.

Zuerst wird das Verfahren implementiert, dann das GUI.
Das Verfahren wird im Laufe der Arbeit weiter dokumentiert.

Hinweis:

Es handelt sich hier lediglich um eine Anforderungsbeschreibung. Im implementierten Prototypen konnten zeitbedingt nicht alle Anforderungen umgesetzt werden.

3 Verfahrensbeschreibung

3.1 SAFE

3.1.1 Grundidee

"Grundidee des SAFE-Verfahrens ist die Vereinheitlichung von Merkmalsträgern im Mikrodatenbestand deart, dass mindestens 3 Merkmalsträger völlig identisch sind."[1]

Die folgende Abbildung veranschaulicht die Idee graphisch.

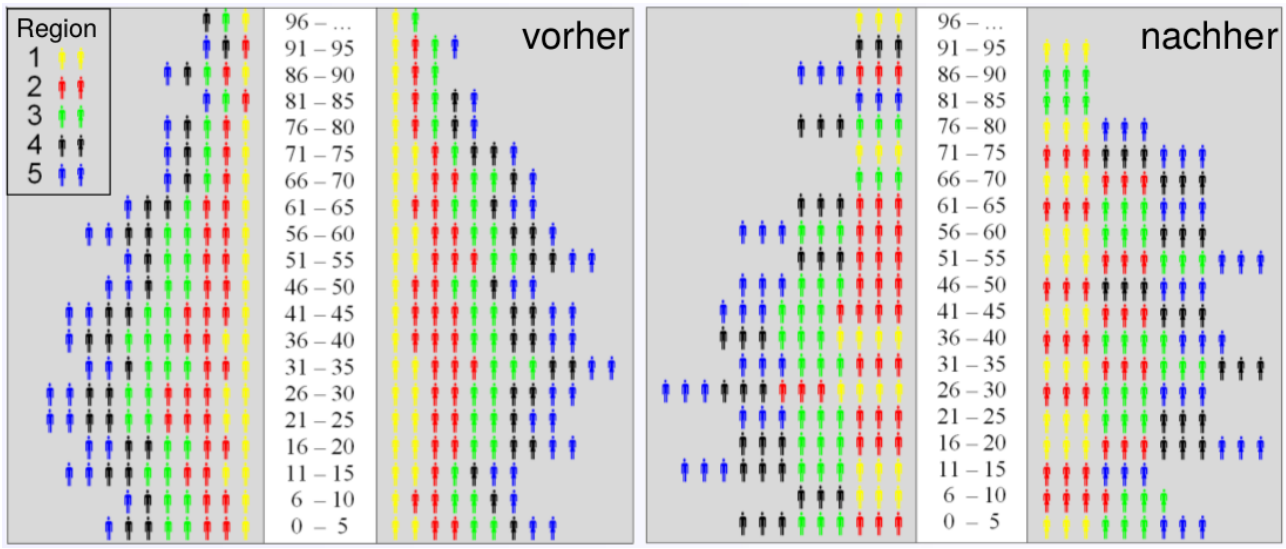


Abbildung 1: Safe-Grundidee. Quelle: Höhne (2012)

Ein tabellarisches Beispiel wäre folgendes:

Vorher:

Merkmal 1	Merkmal 2
A	10
B	12
A	12
A	15
B	12
A	11
C	10
A	25
A	2

Nachher:

Merkmal 1	Merkmal 2
A	11
A	11
A	11
A	11
A	11
A	11
B	12
B	12
B	12

1 Höhne (2010), S-77

3.1.2 Mathematisches Modell

Beispieltabelle:

Alter	Gehalt	Ort	Geschlecht
51	1500	A-Stadt	m
21	550	B-Dorf	w
32	3500	C-Weiler	m

Jeder Datei lässt sich genau eine **metrische Matrix (X)** zuordnen:

$$X = \begin{pmatrix} 51 & 1500 \\ 21 & 550 \\ 32 & 3500 \end{pmatrix}$$

Der Aufbau der Datei ist trivial.

Jedem kategorialen Merkmal i der Datei wird eine sogenannte **Zuordnungsmatrix (Z_i)** zugeordnet. Die Zeilen entsprechen den Zeilen der Originaldatei. Jede Spalte steht für eine Merkmalsausprägung. Deshalb besteht eine Zuordnungsmatrix aus Zeilen, die genau eine Eins und sonst nur Nullen enthalten.

Für den Ort:

$$Z_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Für das Geschlecht:

$$Z_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Die **Kombination von kategorialen Merkmalen** wird durch kombinierte Zuordnungsmatrizen dargestellt. Theoretisch kann eine solche Kombination durch die zeilenweise Berechnung des Kronecker-Produktes berechnet werden:

$$Z_{1,2} = \begin{pmatrix} Z_{1_{Zeile1}} \otimes Z_{2_{Zeile1}} \\ Z_{1_{Zeile2}} \otimes Z_{2_{Zeile2}} \\ Z_{1_{Zeile3}} \otimes Z_{2_{Zeile3}} \end{pmatrix} = \begin{pmatrix} 1\cdot(10) & 0\cdot(10) & 0\cdot(10) \\ 0\cdot(10) & 1\cdot(10) & 0\cdot(01) \\ 0\cdot(10) & 0\cdot(10) & 1\cdot(10) \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Auf diese Weise kann man auch die Kombination von mehr als zwei kategorialen Merkmalen berechnen. Auffällig ist die Entstehung von Nullspalten, weil in der Praxis nicht jede theoretische Merkmalskombination auftritt. "Die Zuordnungsmatrizen sollten deshalb aus Gründen der Größe des mathematischen Modells und somit der Effektivität bei der Berechnung für

Merkmalskombinationen aus den bestehenden Ausprägungskombinationen und nicht aus den theoretisch möglichen hergeleitet werden."[2]

Zur Bestimmung der Ähnlichkeit einer anonymisierten Datei werden zwei Arten von Auswertungen definiert. Die **Auswertung der kategorialen Merkmale** geschieht mittels der Formel:

$$A_j := Z_j^T \cdot Z_j$$

Für jedes kategoriale Merkmal und für jede Kombination aus kategorialen Merkmalen kann man eine solche Auswertungsmatrix berechnen. Ergebnis ist eine Diagonalmatrix, deren Dimension gleich der Anzahl der verschiedenen Merkmalsausprägungen ist. Die Diagonalelemente geben an, wie oft eine Merkmalsausprägung auftritt.

Beispiel:

$$A_2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$$

A_2 kann folgendermaßen interpretiert werden: Es gibt zweimal die Merkmalsausprägung männlich und einmal die Merkmalsausprägung weiblich.

Die **Auswertung der metrischen Merkmäle** geschieht in Kombination mit den kategorialen Merkmalen:

$$T_j := Z_j^T \cdot X$$

Auch hier gibt es theoretisch pro kategorialem Merkmal beziehungsweise pro Kombination aus kategorialen Merkmalen eine solche Auswertungsmatrix.

Beispiel:

$$T_2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 51 & 1500 \\ 21 & 550 \\ 32 & 3500 \end{pmatrix} = \begin{pmatrix} 83 & 5000 \\ 21 & 550 \end{pmatrix}$$

Interpretation: Die Alterssumme der männlichen Datensätze ist 83, der weiblichen 21. Die männlichen Datensätze verdienen zusammen 5000€, die weiblichen 550€.

Das SAFE-Verfahren ist ein **Minimierungsproblem**:

Gegeben ist eine **Basisdatei**, die durch die Matrizen $X^0, Z_1^0 bis Z_k^0$ beschrieben werden kann.

Gesucht ist eine anonymisierte Datei, bei der jede Merkmalskombination mindestens k-mal vorkommt (k > 2). Sie wird beschrieben durch die Matrizen: $X^a, Z_1^a bis Z_k^a$

$X^a, Z_1^a bis Z_k^a$ sollen so gewählt sein, dass die Differenzen der Auswertungsmatrizen minimial sind.

Zusammengefasst:

$Gegeben: X^0, Z_1^0 bis Z_k^0$
$Gesucht: X^a, Z_1^a bis Z_k^a, sodass$
$Differenz\ aller\ A_j^a\ und\ A_j^0\ minimal\ und$
$Differenz\ aller\ T_j^a\ und\ T_j^0\ minimal$

2 Höhne (2010), S-79

3.2 SAFE-Clustering

3.2.1 Konzept

M1	M2
A	10
B	12
A	12
A	15
B	12
A	11
C	10
A	25
A	2

Clustering →

M1	M2
A	10
B	12
A	12
A	15
B	12
A	11
C	10
A	25
A	2

Vereinheitlichung → SAFE!

M1	M2
A	15
A	15
A	15
A	15
A	15
B	12
B	12
B	12

Die Basisdatei wird geclustert. Jede Clustergruppe wird vereinheitlicht, sodass die Datei der SAFE-Anonymität genügt.

3.2.2 Pseudocode

Parameter:

- Basisdatei (abgebildet durch X und Z_1 bis Z_j, vgl. mathematisches Modell von SAFE)
- Menge an Clustering-Algorithmen (C)
- Wie oft soll in der Datei jede Merkmalsausprägung identisch vorkommen? (k)
- Menge an Vereinheitlichungsalgorithmen (V)

Pseudocode:

M = {};

Für jedes c in C:
{
    M = M ∪ (c($Z_{1,2,3,4,...,k}$) \ Cluster mit Größe < k);
    M = M ∪ (c(studentisiere(X)) \ Cluster mit Größe < k);
}
// M hat nun die Form { { {1,3,5}, {2,4,6} } , { {1,2,3}, {4,5,6} } }
// M enthält also für jedes Clusterverfahren in C eine Menge mit den Clustergruppen, die durch das
// Clusterverfahren gebildet wurden.

N = {};

Für jedes m in M:
{
    Für jedes v in V
    {
        N = N ∪ v(m);
    }
}

//N enthält also für jedes Clusteringverfahren eine Vereinheitlichung.

//wird über die Differenz der Auswertungsmatrizen zu B bestimmt.
return besteVereinheitlichung(N);

4 Entwicklerdokumentation

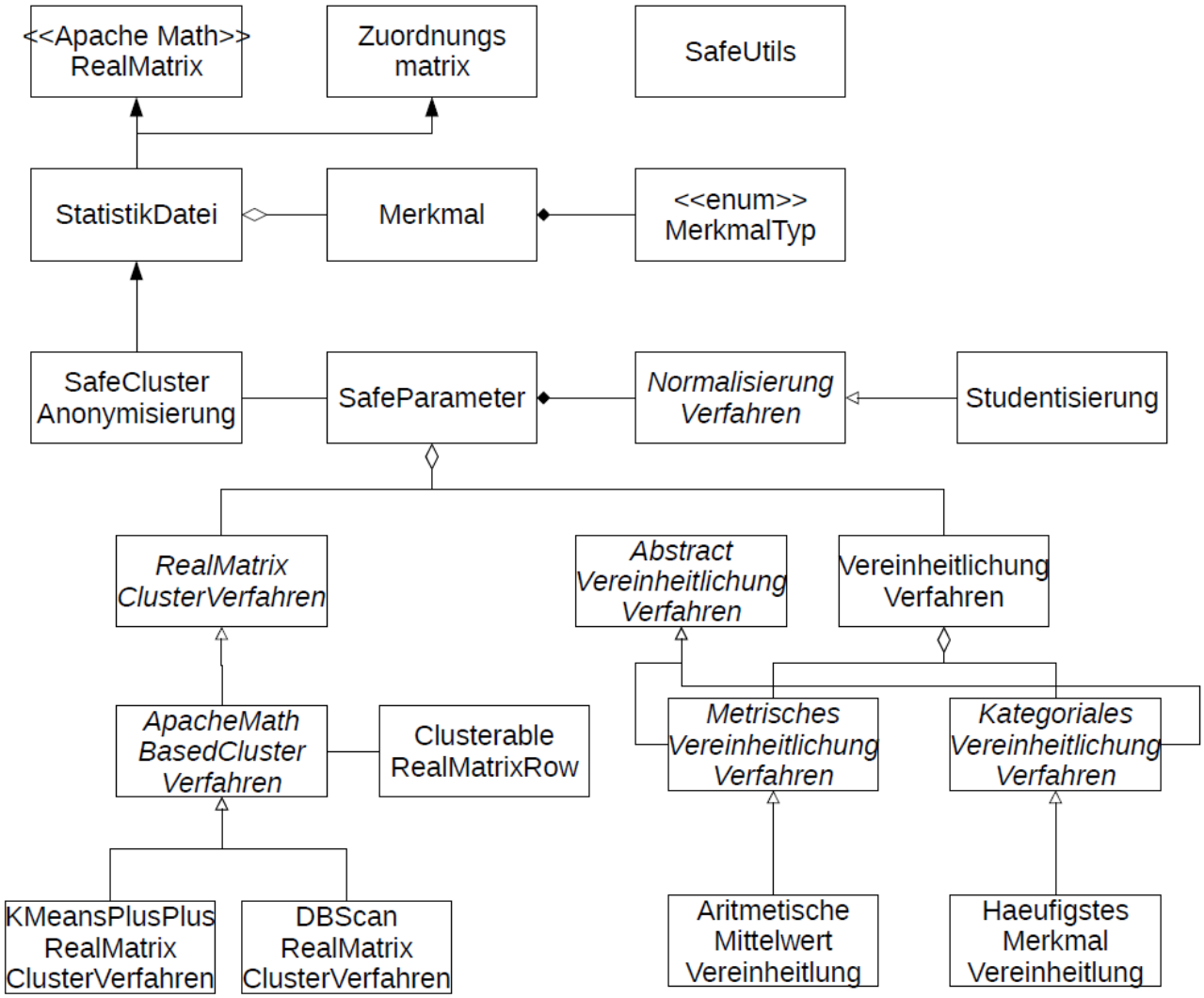
4.1 Wichtige Bibliotheken

4.1.1 Apache Commons Math

Definiert wichtige Datentypen der linearen Algebra (z.B. Matrizen)[3], aber enthält auch einige Clusteringalgorithmen[4]. Die Bibliothek wäre aber sicherlich noch für andere Anwendungsfälle interessant.

3 http://commons.apache.org/proper/commons-math/userguide/linear.html
4 http://commons.apache.org/proper/commons-math/userguide/ml.html

4.2 Wichtige Backend-Klassen



Neue Clusterverfahren, Vereinheitlichungsverfahren oder Normalisierungsverfahren können ganz einfach durch Vererbung hinzugefügt werden.

Zum besseren Verständnis der Funktionsweise der wichtigen Klassen empfiehlt sich ein zusätzlicher Blick in den Testordner (src/test/java).

4.2.1 SafeUtils

Diese Klasse enthält nützliche Hilfsimplementierungen:

- Kroneckerprodukt
- Überprüfung, ob in einer Datei jede Zeile mindestens k-mal vorkommt
- Berechnung des Abstands zweier Dateien (mit Hilfe der in SAFE definierten Auswertungsmatrizen)

4.3 GUI-Entwicklung

Die Basis der GUI-Entwicklung für das SAFE-Verfahren ist die Klasse **de.htw.sg.safe.gui.SafePanel**.

4.4 Lauffähiges Jar-File erzeugen

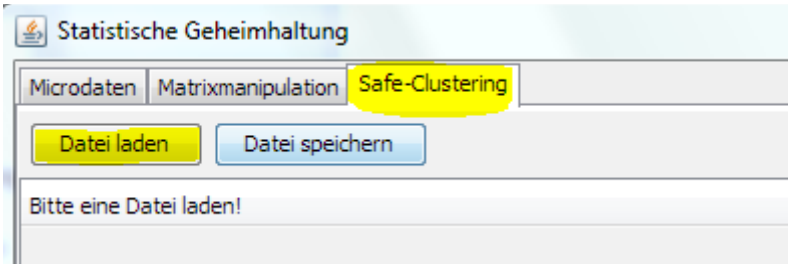
Dazu einfach folgenden Mavenbefehl ausführen:

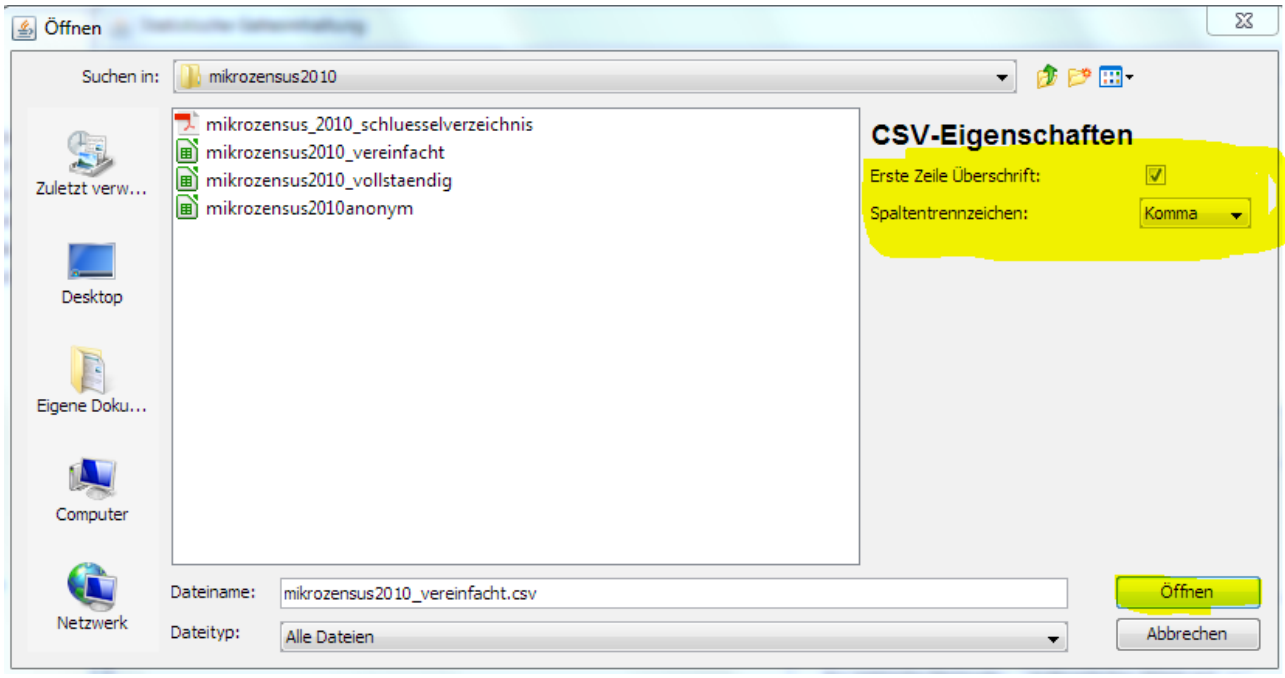
mvn clean compile assembly:single

5 Benutzerdokumentation

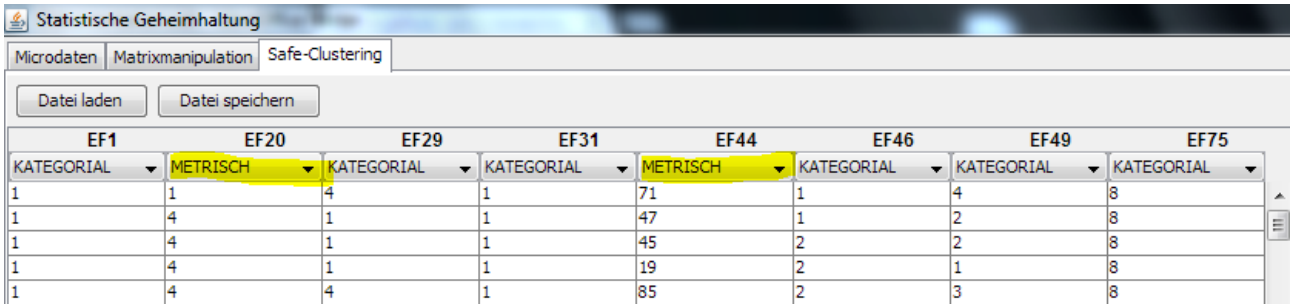
5.1 Beispielanwendung

5.1.1 Laden einer Datei



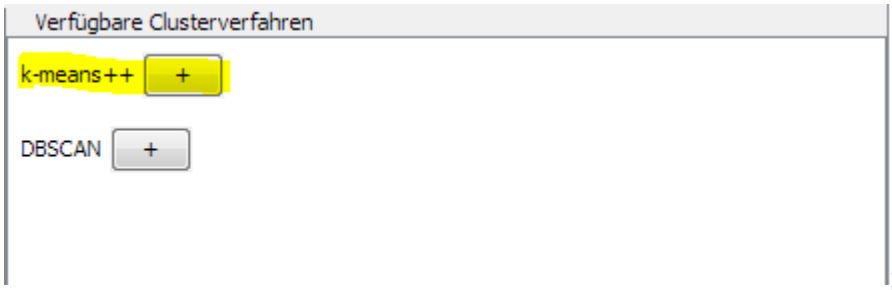


5.1.2 Auswahl der metrischen Merkmale



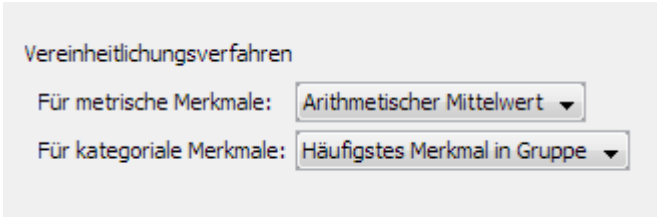
EF1	EF20	EF29	EF31	EF44	EF46	EF49	EF75
KATEGORIAL	METRISCH	KATEGORIAL	KATEGORIAL	METRISCH	KATEGORIAL	KATEGORIAL	KATEGORIAL
1	1	4	1	71	1	4	8
1	4	1	1	47	1	2	8
1	4	1	1	45	2	2	8
1	4	1	1	19	2	1	8
1	4	4	1	85	2	3	8

5.1.3 Auswahl der zu verwendenen Clusterfahren



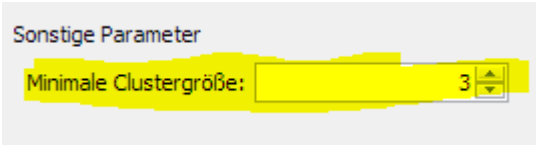
5.1.4 Konfiguration der zu verwendenen Clusterverfahren

5.1.5 Vereinheitlichungsverfahren auswählen

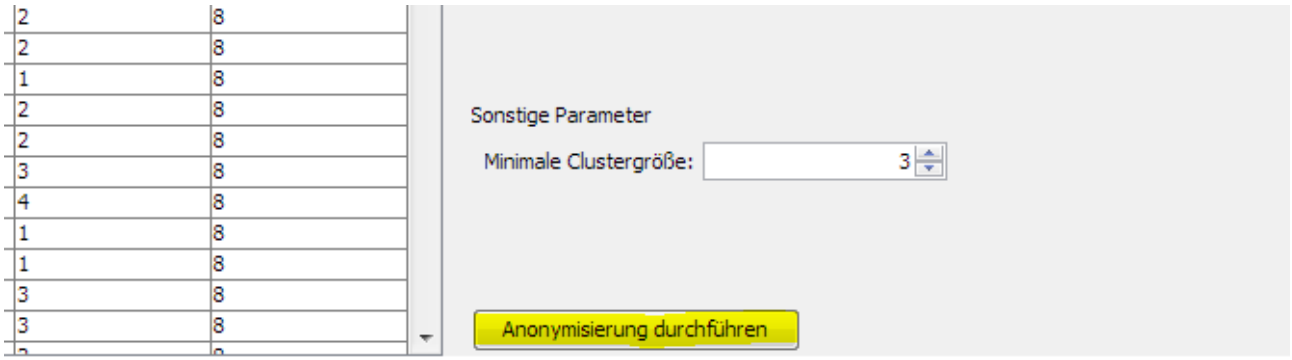


5.1.6 Bestimmung der minimalen Clustergröße

Dieser Parameter gibt an, wie oft jede Merkmalsausprägung in der anonymisierten Datei mindestens vorkommen muss.



5.1.7 Anonymisierung starten



Dieser Prozess kann eine Weile in Anspruch nehmen.

5.2 Verwendete Clusteralgorithmen

5.2.1 K-Means++

Es handelt sich um eine verbesserte Version des K-Means-Algorithmus.

Die Grundidee von K-Means ist die Minimierung der Funktion $\varphi = \sum_{x \in X} \min_{c \in C} \| x - c \|^2$

X bezeichnet dabei die Menge der zu clusternden Punkte. C ist die gesuchte Menge der Clusterzentren, wobei k Clusterzentren gewählt werden sollen (|C|=k).

K-Means möchte die Clusterzentren so wählen, dass der quadrierte Abstand der Punkte eines Clusters zum Clusterzentrum minimal ist. Da dieses Problem zur Klasse NP gehört, kann nur eine Näherungslösung bestimmt werden.

Die Standardimplementierung wählt k zufällige Start-Clusterzentren und versucht in einem iterativen Verfahren eine solche Näherungslösung zu finden. Der K-Means++-Algorithmus durchläuft im wesentlichen dieselben Schritte, versucht die Startpunkte aber intelligenter zu wählen.

Details lassen sich im Originalpaper von Arthur und Vassilvitskii nachlesen.[5]

Nach der Idee sollten höhere Werte für k zu besseren Ergebnissen für das Anonymisierungsverfahren führen.

Ein Beispiel ist unter http://commons.apache.org/proper/commons-math/userguide/ml.html zu finden.

5 Arther und Vassilvitskii (2007)

5.2.2 DBScan

DBScan verfolgt eine einfache Annahme: Punkte in einem Cluster liegen dicht beieinaner.

Deshalb werden folgende Begriffe definiert:

- A ist von B **erreichbar**, wenn $d(A,B) < \varepsilon$
- A ist **Kernpunkt**, wenn A *minPts* erreichbare Nachbarn besitzt. Kernpunkte liegen bildlich gesehen in der Mitte eines Clusters
- A ist **Dichte-erreichbar**, wenn A von einem Kernpunkt erreichbar ist, selbst aber kein Kernpunkt ist. Dichte-erreichbar liegen bildlich gesehen am Rand eines Clusters
- Ansonsten ist A **Rauschpunkt (Noise)**
- Kernpunkte und Dichte-erreichbare Punkte bilden einen **Cluster**

Details sind im Originalpaper von Ester et al. zu finden.[6]

minPts sollte ungefähr 3 sein (mindestens 3 identische Merkmalsträger in der anonymisierten Datei)

ε ist schwieriger zu bestimmen, da es sehr datenabhängig ist. Wählt man ε zu klein, so werden alle Punkte als Noise erkannt und es wird kein Cluster gebildet. Wählt man ε zu groß, so sind alle Punkte in einem Cluster.

Ein Beispiel ist unter http://commons.apache.org/proper/commons-math/userguide/ml.html zu finden.

6 Ester et al. (1996)

6 Offene Aufgaben

6.1 Neue Funktionen

- Einbau eines SplitPanels zwischen Dateiansicht und Parameterpanels, sodass man die Ansichtsgröße frei verändern kann
- Implementierung weiterer Clustering-Verfahren (Single-Linkage, Complete-Linkage, Average-Linkage, etc.)
- Behandlung von Missings (momentan führen Missings bei metrischen Merkmalen zu Programmabstürzen)
- Clustering nach metrischen und kategorialen Merkmalen gleichzeitig (durch Kombination der Zeilenvektoren von X und $Z_{1..k}$)
- Wenn mehrere Ergebnisse produziert werden, wäre es schön, die angezeigt zu bekommen und nicht nur das nach SAFE präferierte

6.2 Funktionsverbesserung

- Speichereffizienz bei den Zuordnungsmatrizen (bei vielen kategorialen Merkmalen kommt es zu Abstürzen, weil der Arbeitsspeicher nicht ausreicht)
- Laufzeitanalyse (Verdacht: Vereinheitlichungen sind sehr langsam)
- Delimiter (Trennzeichen für CSV-Spalten) sollte frei wählbar sein
- Warnung/Fehlermeldung, wenn keine Cluster gebildet werden konnten (z.B. ε zu klein bei DBScan)
- Mehrfachauswahl von Clusteringverfahren (Es wäre schön beispielsweise zweimal k-means mit unterschiedlichem k laufen zu lassen)
- Fortschrittsanzeige oder zumindest Hinweis, wenn dass das Verfahren noch läuft

6.3 Bugs

- Lädt man eine Datei, bei der nicht die erste Zeile Überschrift ist, so kann man nicht auswählen, welche Merkmale metrisch und welche kategorial sind.

6.4 Sonstiges

- Intensivere Auswertungen zur Qualitätsbestimmung des Verfahrens (vgl. Präsentationsfolien)
- Umbenennung des Parameters minimale Clustergröße in minPts beim DBScanPanel

6.5 Mängel in der Dokumentation

- Informationen, wie genau bei den 3 möglichen Durchgängen (kategorial, metrisch, kategorial + metrisch) vorgegangen wird, wären wünschenswert. Überlegung meinerseits: Müsste nicht "kategorial + metrisch" nicht generell zu den "besten" Ergebnissen führen, da so die Heterogenität in den Gruppen am geringsten sein sollte?
  => Prinzipiell ja. Darauf wurde in der Präsentation auch kurz eingegangen.
- Die Clusteralgorithmen könnten ausführlicher erklärt sein (evtl. auch durch anschauliche Beispiele)//insbesondere keinerlei Ansatz Wahl der Start-Clusterzentren bei k-means++ zu

erklären (zeitbedingt nicht mehr geschafft)

7 Literaturverzeichnis

Arthur, David, and Sergei Vassilvitskii. *k-means++: The advantages of careful seeding*. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 2007.

Ester, Martin, et al. *A density-based algorithm for discovering clusters in large spatial databases with noise.* Kdd. Vol. 96. No. 34. 1996.

Höhne, Jörg. *Methoden und Potenziale des Zensus 2011*, 2012.
Vortrag auf den Statistik-Tagen Bamberg 2012. Folien verfügbar unter
https://www.statistik.bayern.de/medien/wichtigethemen/st_vortrag_hoehne_27072012.pdf
(letzter Aufruf am 17.06.15)

Höhne, Jörg.*Verfahren zur Anonymisierung von Einzeldaten*. HGV Servicecenter Fachverlage, 2010.