Arbeitsblatt 3

Aufgabe 1: MNIST

Wir betrachten noch einmal den MNIST-Datensatz (`mnist_2k.Rdata`) mit 2000 zufällig ausgewählten handgeschriebenen Nummern von 0 bis 9 (Vergleiche Aufgabe 3 Arbeitsblatt2). Nach dem Laden des Datensatzes steht das Dataframe `x` zur Verfügung. Jede Zeile des Dataframes enthält die linearisierten Pixel-Grauwerte eines Bildes `pixel0` bis `pixel783`. Zudem gibt es eine Spalte `label` mit den Labeln.

a) Führen Sie mit dem K-Means Ansatz mit 10 Clustern (`kmeans(...,centers=10)`) ein Clustering durch. Visualisieren Sie das Resultat im 2D t-SNE-Plot. Kommentieren Sie das Resultat. Code zur Visualisierung:

```
library(Rtsne)
restSNE <- Rtsne(x[, -1], perplexity = 20)
df <- as.data.frame(restSNE$Y)
names(df) <- c("tsne1","tsne2")
df$label <- as.factor(x$label)
df$cluster <- as.factor(res_cl$cluster) #res_cl -> Ergebnis kmeans
ggplot(data = df, aes(x = tsne1, y = tsne2, col = cluster, label=label)) +
  geom_text(size=3)
```

b) Erzeugen Sie mit dem Befehl `table(df$label, LETTERS[df$cluster])` eine Konfusionsmatrix. Was sehen wir? Was sagen uns die Zahlen?

Aufgabe 2: Abstimmungen

Wir wollen nun noch einmal mit dem Datensatz der eidgenössischen Abstimmungen (`abst.Rdata`) arbeiten.

a) Wir versuchen nun die Kantone zu clustern. Verwenden Sie dazu den K-means Algorithmus. Bestimmen Sie die optimale Anzahl Cluster mit Hilfe der "Within-cluster Variation".

b) Visualisieren Sie das Ergebnis der Clusteranalyse in einem 2D-Plot der ersten beiden Hauptkomponenten einer PCA. Im Standardplot können Sie dazu die Cluster farblich wie folgt übergeben (`col=c("green", "blue", ...)[res.km$cluster]`). Häufig ist es informativ, wenn man zusätzlich die Zentren der Cluster plottet. Dazu muss man die Daten in die PCA-Projektion transformieren (`predict(abst.pca, newdata=res.km$centers)`)).

c) Analysieren Sie Ihr Ergebnis mit einem Silhouetten-Plot `plot(silhouette(...))`. Dazu müssen Sie vorgängig das Paket `cluster` laden (`library(cluster)`).

d) Bestimmen Sie die optimale Anzahl Cluster mit mittlerer Silhouttenbreite. (Tipp: R-Funktion `fviz_nbclust(...,method='silhouette')` aus dem Paket `factoextra`).

Aufgabe 3: Nationalrat

In dieser Aufgabe arbeiten wir noch einmal mit dem Abstimmungsverhalten des Schweizer Parlaments (Nationalrats). Datensatz `voting_NR.rdata` mit der "Distanz"-Matrix: `NR_voting` und den Metainformationen (Fraktion und Kanton): `NR_meta` (vergleiche Arbeitsblatt 2 Aufgabe 2).

a) Führen Sie mit dem K-Medoids/PAM Ansatz das Clustering für die Distanzmatrix `NR_voting` durch. Wie viele Klassen sind hier angebracht?

b) Was sind die repräsentativen Mitglieder der Cluster (Medoid)?

c) Visualisieren Sie das Ergebnis der Clusteranalyse mit MDS (`cmdscale`) oder t-SNE (`Rtsne`).

d) Erzeugen Sie eine Konfusionsmatrix zwischen dem Clustern und den Fraktionen `table(NR_meta$Fraktion, res.pam$clustering)`. Passen die Cluster zu den Fraktionen?

Aufgabe 4: Hierarchisches Clustering

Die Matrix `CD.dis` im File `CountriesDis.RDA` (mit `load()` laden) enthält Unähnlichkeiten zwischen Ländern. Die Daten stammen aus einer etwas älteren Studie, in der Studierende aufgefordert waren, paarweise die Unähnlichkeit zwischen den 12 Ländern Belgien (BEL), Brasilien (BRA), Chile (CHI), Kuba (CUB), Ägypten (EGY), Frankreich (FRA), Indien (IND), Israel (ISR), Vereinigte Staaten (USA), Sowjetunion (USS), Jugoslawien (YUG) und Zaire (ZAI) anzugeben. In der Unähnlichkeitsmatrix sind die durchschnittlichen Bewertungen der Studierenden festgehalten.

a) Führen Sie mit dieser Unähnlichkeitsmatrix hierarchische Cluster-Analysen durch, verwenden Sie verschiedene Linkage-Methoden. Vergleichen Sie dabei die Resultate der Cluster-Methoden `single`, `complete`, `average` und `ward.D2` bezüglich der Gruppenbildung.

b) Inwiefern finden Sie in einer multidimensionalen Skalierung die Resultate der hierarchischen Cluster-Analyse wieder?

c) Analysieren Sie das Ergebnis des hierarchischen Clusters mit dem Complete-Linkage Ansatz mit dem Silhouetten-Plot. Um aus dem Dendogramm die Cluster für eine spezifische Anzahl Cluster oder eine spezifische Höhe zu erhalten, können Sie die R-Funktion `cutree` verwenden. Können Sie mit der Silhouetten-Methode auch die optimale Anzahl Cluster bestimmen.

d) Führen Sie das Clustern auch noch mit einem Partitionsverfahren (K-means oder PAM) durch. Bestimmen Sie die optimale Anzahl Cluster und visualisieren Sie das Ergebnis. Welches sind die typischen Vertreter der einzelnen Cluster?

Aufgabe 5: Heatmap

Betrachten Sie noch einmal die Abstimmungsdaten und visualisieren Sie die Daten mit einer Heatmap (`heatmap`). `heatmap` erwartet als Input eine numerische Matrix. Sie müssen die Daten darum mit `as.matrix(...)` noch anpassen. Spielt es eine Rolle, wie Sie die Daten skalieren (Argument `scale`)? Wenn Sie Lust haben, können Sie auch die Alternative `pheatmap` aus dem Paket `pheatmap` testen.

Aufgabe 6: Diabetes

In dieser Aufgabe arbeiten wir mit dem Datensatz diabetes aus dem Paket `mclust`. Dazu müssen Sie zuerst das Paket installieren (`install.packages("{}mclust")` und laden (`library(mclust)`). Die Daten stehen Ihnen dann zur Verfügung. Informationen zu den Daten gibt es mit `?diabetes`.

a) Stellen Sie die Daten graphisch dar. Was sehen Sie? Macht es Sinn die Daten für das Clustering zu normalisieren?

b) Führen Sie mit einem herkömmlichen Ansatz (K-Means, pam oder hierarchisches Clustering) ein Clustering der standardisierten Variablen `glucose`, `insulin` und `sspg` durch

(`scale(diabetes[, -1]`). Vergleichen Sie Ihr Clusterergebnis mit den tatsächlichen Gruppen in der Variable `class` mit einer Konfusionsmatrix (`table(cluster=res$cluster, diabetes$class`). Was sehen Sie?

c) Führen Sie nun die Clusteranalyse mit dem modellbasierten Clustering durch.

Aufgabe 7: Zusatz DBscan

In dieser Aufgabe wollen wir noch den DBscan Algorithmus für Dichteverbundenes Clustern anschauen. Dazu betrachten wir den Datensatz `multishapes` aus dem Paket `factorextra`. Das ist ein synthetischer Datensatz mit verschiedenen Formen, der generiert wurde, um die Fähigkeit von Dbscan zu demonstrieren. Die ersten beiden Spalten enthalten x- und y-Werte. Die dritte Spalte ist die Form Nummer.

a) Visualisieren Sie den Datensatz.

b) Versuchen Sie die x- und y-Variablen mit K-Means in 5 Gruppen zu clustern.

c) Nun versuchen wir die Daten mit DBSCAN zu clustern. Verwenden Sie dazu die Funktion dbscan aus dem Paket `dbscan` (muss vorgängig installiert werden). Neben den Daten müssen bei diesem Algorithmus die Argumente `minPts` (minimale Anzahl Punkte, die um einen Punkt liegen müssen, damit dieser ein Cluster wird) und `eps` (Umgebung welche für eps verwendet wird) definiert werden. Verwenden Sie hier die Werte `eps=0.15` und `minPts=5`. Die Clustereinteilung kann man aus dem Resultatobjekt mit `res$cluster` ziehen. Beachten Sie, dass Rauschpunkte mit 0 codiert sind. Will man die auch farblich darstellen, dann muss man in der plot-Funktion `col= res$cluster+1` schreiben.

d) Nun können Sie auch noch versuchen die Daten mit einem modellbasierten Clustering zu identifizieren. Standardmässig werden in der Funktion `mclust` 1 bis 9 Gruppen gesucht. Das ist hier allenfalls nicht ausreichend. Um mehr Gruppen zu suchen, müssen Sie das Argument `G` anpassen.

Aufgabe 8: Leistungsnachweis

- Versuchen Sie Ihren Datensatz (oder Teile davon) mit einer geeigneten Methode zu clustern.
- Geben Sie an, wie Sie die Anzahl Cluster festlegen.
- Visualisieren Sie Ihre Cluster.
- Interpretieren Sie Ihre Ergebnisse.