Allgemeines

Die vorgesehene Bearbeitungszeit für das Übungsblatt endet am Freitag, 12.02.2021, um 12:00 Uhr. Kurze Zeit später wird das nächste Übungsblatt hochgeladen.

Falls Sie anonymes Peer-Feedback geben und erhalten wollen, laden Sie Ihre Lösung zu den Programmieraufgaben innerhalb der Frist in das ILIAS-Übungsobjekt "Abgaben" hoch. Feedback kann erst nach Ablauf der Abgabefrist gegeben werden. Geben Sie das Feedback bis zum Mittwoch, 17.02.2021, 23:55 Uhr. Erst nach Ablauf dieser Feedbackfrist wird dann das erhaltene Feedback sichtbar. Weitere Details zum Peer-Feedback finden Sie in der Datei "Feedback_geben.pdf" im ILIAS.

1 Online-Test 5

Der fünfte Online-Test steht ab dem 18.01.2021, 00:00 Uhr im ILIAS bereit. Er beschäftigt sich mit den Vorlesungskapiteln 9 (Clustering) und 10 (Identifikation von Outliern).

2 Clustering

Ziel dieser Aufgabe ist es, verschiedene Clustering-Verfahren auf den eingebauten `faithful`-Datensatz anzuwenden und zu vergleichen.

a) [**Visualisierung**] Machen Sie sich mit den `faithful`-Daten vertraut. Normalisieren Sie die Daten mit `scale()` und erstellen Sie einen Scatterplot. Können Sie bereits Cluster erkennen? Warum haben wir die Daten überhaupt normalisiert? *(Basis)*

b) [**Partitionierendes Clustering**] Wenden Sie mittels `kmeans()` ein partitionierendes Clustering-Verfahren an. Visualisieren Sie die Ergebnisse (indem Sie z. B. die Punkte im Scatterplot gemäß der Clusterzugehörigkeit einfärben) für verschiedene Werte von k. Warum führt k-means zu diesen Ergebnissen? *(Basis)*

c) [**Silhouettenkoeffizient**] Berechnen Sie mittels `cluster::silhouette()` die Qualität der k-means-Ergebnisse. Sie können die Ergebnisse der Silhouettenberechnung mit `plot()` visualisieren. Wie sind Silhouetten-Plots zu interpretieren? Wie kann Ihnen der Silhouettenkoeffizient helfen, ein geeignetes k zu finden? *(Basis)*

d) [**Hierarchisches Clustering**] Wenden Sie mittels `hclust()` ein hierarchisches Clustering-Verfahren an. Wozu dient der Parameter `method`? Visualisieren Sie das Dendrogramm durch Plotten des Ergebnis-Objektes. Wie bekommen Sie eine Cluster-Zuordnung aus dem Dendrogramm? Vergleichen Sie die Ergebnisse zu denen von k-means. *(Vertiefung)*

e) [**Dichtebasiertes Clustering (1)**] Wenden Sie mittels `dbscan::dbscan()` ein Dichtebasiertes Clustering-Verfahren an und visualisieren Sie das Ergebnis. Warum erhalten Sie so ein Ergebnis? Wie können Sie das Ergebnis verbessern? Ist der Silhouettenkoeffizient hier ein geeignetes Qualitätsmaß? *(Basis)*

f) [**Dichtebasiertes Clustering (2)**] Verwenden Sie `dbscan::optics()` zum Dichtebasierten Clustern. Visualisieren Sie das Ergebnis-Objekt mittels `plot()` als Erreichbarkeitsgraphen. Lässt sich damit ein geeigneter Wert für ϵ bestimmen? Wandeln Sie das OPTICS-Ergebnis über `extractDBSCAN()` in ein DBSCAN-Ergebnis um und visualisieren Sie es. *(Vertiefung)*

g) **[Mixture Models]** Verwenden Sie `mclust::Mclust()` zum Clustern auf Basis von Gaussian Mixture Models. Welche Hyper-Parameter hat das Modell und wie werden deren Werte bestimmt? Wie erklären Sie sich das Clustering-Ergebnis? Die `plot()`-Funktion kann Ihnen bei der Beantwortung helfen. *(Vertiefung)*

3 Outlier Detection

Ziel dieser Aufgabe ist es, verschiedene Arten von Ausreißern in den `faithful`-Daten zu finden. Vergessen Sie nicht, die Daten zuvor zu normalisieren.

a) **[Visualisierung]** Visualisieren Sie die Verteilung der Attribute im `faithful`-Datensatz mit univariaten und bivariaten Plots, z. B. mit Dichteplots oder Histogrammen. (In `ggplot2` eignen sich z. B. `geom_density()`, `geom_histogram()` und `geom_bin2d()`.) Welche Punkte könnten Ausreißer sein? *(Basis)*

b) **[Ausreißererkennung]** Wenden Sie `KNN_SUM()` aus dem Paket `DDoutlier` als einen kNN-Distanz-basierten Ansatz und `lof()` aus dem Paket `dbscan` als einen Dichte-basierten Ansatz an. Übergeben Sie hierbei den ganzen Datensatz, d.h. mit beiden Spalten. Visualisieren Sie die Datenpunkte zusammen mit Ihrem Ausreißer-Score. In `ggplot2` können Sie zum Beispiel `color` und `size` zur Hervorhebung verwenden. Für welche Datenpunkte unterscheiden sich die Ergebnisse der beiden Ausreißererkennungsverfahren stark und warum? *(Basis)*

c) **[Teilraum-Ausreißer]** Berechnen Sie die Ausreißer-Scores nun jeweils einzeln auf den Attributen (Spalten) des Datensatzes. Vergleichen Sie die Ergebnisse zum vorigen zweidimensionalen Ansatz. Gibt es Punkte, die nur in einem Teilraum Ausreißer sind? Gibt es nicht-triviale Ausreißer in den Daten? *(Vertiefung)*

4 Bring Your Own Theoretical Task

Denken Sie sich für die nächste Übungssitzung eine Frage aus, wie sie in der mündlichen Prüfung gestellt werden könnte. Sie können sich von der Art her an den möglichen Prüfungsfragen am Ende der Vorlesungskapitel orientieren. Inhaltlich sollte sich die Frage auf eines der Vorlesungskapitel 9 (Clustering) bis 10 (Identifikation von Outliern) beziehen. Vom Niveau her sollte die Frage nicht einfach Wissen 1:1 abfragen, sondern Verständnis fordern. Beispielsweise könnte es darum gehen, Sachverhalte zu vergleichen, einzuordnen, zu analysieren etc. Wir werden die Fragen in der Übungssitzung zunächst in Kleingruppen diskutieren und danach die interessantesten Fragen im Plenum besprechen. Sie müssen die Frage nicht zusammen mit Ihrem Code hochladen.