

## Allgemeines

Die vorgesehene Bearbeitungszeit für das Übungsblatt endet am Freitag, 15.01.2021, 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, 20.01.2021, 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 4

Der vierte Online-Test steht ab dem 21.12.2020, 00:00 Uhr im ILIAS bereit. Er beschäftigt sich mit den Vorlesungskapiteln 6 (Association Rules), 7 (Schnelles Bestimmen der Frequent Itemsets) und 8 (Pattern Mining in Gegenwart von Constraints).

## 2 Association Rules

Ziel dieser Aufgabe ist es, verschiedene Varianten von Frequent Itemset Mining und Association Rule Mining anzuwenden. Hierzu setzen wir das Paket `arules` ein, welches den Datensatz `Groceries` mitliefert.

- a) [**Transaktionsdaten**] Laden Sie den Datensatz mit `data(Groceries)`. Wie ist der Datensatz aufgebaut? (*Basis*)
- b) [**Frequent Itemset Mining**] Bestimmen Sie mittels `apriori()` alle Frequent Itemsets mit einem Support von mindestens 5%. Mittels `inspect()` können Sie die Itemsets betrachten. Was passiert, wenn Sie alle maximalen Frequent Itemsets mit dieser Support-Grenze bestimmen? Maximalität lässt sich wahlweise in `apriori()` einstellen oder nachträglich herausfiltern. (*Basis*)
- c) [**Association Rules Mining**] Bestimmen Sie mittels `apriori()` alle Association Rules mit einem Support von mindestens 1% und einer minimalen Confidence von 40%. Welche fünf Regeln haben die höchste Confidence? (Tipp: `head()`-Funktion) Welche Regeln enthalten *yogurt* (als eines der Items) auf der linken Seite und haben eine Confidence größer als 50%? (Tipp: `subset()`-Funktion) (*Basis*)
- d) [**Visualisierung**] Nutzen Sie das Paket `arulesViz`, um die extrahierten Regeln zu visualisieren. Die CRAN-Homepage des Pakets enthält einen ausführlichen Artikel (‘vignette’), welcher die verschiedenen Plot-Typen darstellt. (*Vertiefung*)
- e) [**Constraints (1)**] Behalten Sie die bisherigen Werte für Support und Confidence bei. Bestimmen Sie alle Regeln, die ausschließlich *yogurt* auf der rechten Seite haben. Versuchen Sie hierbei, die Constraints bereits in `apriori()` selbst zu berücksichtigen. Bestimmen Sie nun alle Regeln, die ausschließlich *yogurt* auf der linken Seite haben. Warum unterscheiden sich die Ergebnisse? (*Basis*)
- f) [**Constraints (2)**] Erhalten Sie dieselben Ergebnisse, wenn Sie die Constraints wie in c) erst nach Ablauf des Algorithmus anwenden? Wann und warum würden Sie Mining mit Constraints im Vergleich zu Post-Processing bevorzugen? (*Basis*)

- g) [**Multi-Level (1)**] Nutzen Sie die Funktion `aggregate()`, um die **Groceries**-Daten auf **level2** zu aggregieren. Extrahieren Sie alle Regeln mit einem minimalen Support von 10% und einer minimalen Confidence von 40%. Warum ist die höhere Support-Grenze im Vergleich zu vorigen Teilaufgaben sinnvoll? (*Vertiefung*)
- h) [**Multi-Level (2)**] Nutzen Sie die Funktion `addAggregate()`, um eine Level-Crossing-Repräsentation des Datensatzes aus den ursprünglichen Items und **level2** zu erstellen. Bestimmen Sie alle Regeln mit den Schwellwerten der vorigen Teilaufgabe. Was fällt Ihnen auf? (*Vertiefung*)
- i) [**Klassifikation (1)**] Association Rules können auch für simple Klassifikationsmodelle verwendet werden. Hierzu nutzen wir den **iris**-Datensatz und das Paket **arulesCBA**. Diskretisieren Sie zunächst die numerischen Attribute (**arulesCBA** stellt entsprechende Funktionen bereit) und erstellen Sie ein **transactions**-Objekt. Extrahieren Sie nun Regeln, die **Species** als rechte Seite haben. Sind die Regeln sinnvoll? Was sind Vor- und Nachteile eines solchen Ansatzes im Vergleich zu anderen Klassifikationsmethoden? (*Vertiefung*)
- j) [**Klassifikation (2)**] Nutzen Sie die Funktion `CBA()`, um ein vollwertiges Regel-basiertes Klassifikationsmodell zu trainieren. Was ändert sich im Vergleich zur vorigen Teilaufgabe? Machen Sie eine Vorhersage und bewerten Sie die Vorhersagequalität. (*Vertiefung*)

### 3 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 4 (Klassifikation mit Entscheidungsbäumen) bis 8 (Pattern Mining in Gegenwart von Constraints) 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.