WMuEMaL

Friedrich-Schiller-Universität Jena

Fakultät für Mathematik und Informatik
INSTITUT FÜR INFORMATIK
Prof. E.G. Schukat-Talamazzini

WERKZEUGE MUSTERERKENNUNG & MASCHINELLES LERNEN

Aufgabenblatt 6

(Ausgabe am Fr 24.5.2019 — Abgabe bis So 2.6.2019)

Aufgabe 1 — 8 P

Diese Aufgabe behandelt die maschinelle Gruppierung landessprachlicher Texte nach einem informationstheoretischen Distanzkriterium (Skript[1] „Stochastische Grammatikmodelle“ *VIII.6, S. 13–15).*

(a) *Laden Sie die Liste (*`zip.rda`*) der Zeichenkettenvektoren von 43 Übersetzungen des UDHR-Dokuments (Menschenrechtedeklaration der UN).*

(b) *Schreiben Sie eine Funktion* `bits(x,compress=TRUE)`*, die einen Textvektor* `x` *mit dem* GZIP*-Verfahren ('R'-Funktion* `memCompress`*) komprimiert und als Ergebnis die Anzahl der erzeugten Bits abliefert. Für* `compress=FALSE` *geben Sie die Bitzahl des Originaltextes zurück.*

(c) *Erzeugen Sie eine Cleveland-Grafik (*`?dotchart`*) mit den absteigend sortierten Kompressionsfaktoren für alle Landessprachen.*

(d) *Nach Shannon benötigt ein Komprimierer $\mathcal{H}(p)$ Bits/Zeichen (Entropie), um einen p-verteilten Text x_p zu kodieren, wenn er die Verteilung p zum Verschlüsseln verwendet. Verschlüsselt er mit abweichender Verteilung q, so werden es $\mathcal{H}(p\|q)$ Bits/Zeichen (Kreuzentropie). Schreiben Sie eine Funktion* `cross(xp,xq)`*, welche näherungsweise die Kreuzentropie $\mathcal{H}(p\|q)$ für die Verteilungen p und q der Texte* `xp` *und* `xq` *berechnet. Die Bitzahl einer q-Verschlüsselung von* `xp` *sollten Sie durch Aufrufe* `bits(c(xq,xp))` *und* `bits(xq)` *ermitteln können.*

(e) *Schreiben Sie den Einzeiler* `divergence(xp,xq)` *zur Berechnung der Kullback-Leibler-Divergenz $\mathcal{D}(p\|q) = \mathcal{H}(p\|q) - \mathcal{H}(p\|p)$ sowie die Funktion* `distance(X)`*, die für die Textliste* `X` *eine Distanzmatrix (Klasse* `dist`*) mit allen wechselseitigen Textdistanzen $d_{ij} = \mathcal{D}(p_i\|p_j) + \mathcal{D}(p_j\|p_i)$ (symmetrische Divergenz) erzeugt. Vergessen Sie bitte nicht die Mitnahme der Textprobenamen aus* `X`*.*

(f) *Und nun clustern Sie die Textproben, indem Sie ihre Distanzmatrix den Methoden* `agnes` *bzw.* `diana` *('R'-Paket* `cluster`*) zur agglomerativen/divisiven Gruppierung übergeben und die Dendrogrammgrafiken ausgeben.*

[1] URL: `http://www.minet.uni-jena.de/fakultaet/schukat/SGM/Scriptum/lect08-NLP.pdf`

(g) *Distanzdaten lassen sich auch näherungsweise in der $\mathbb{R}^2$-Ebene visualisieren. Konsultieren Sie die Handbuchseiten zu den MDS-Verfahren* `cmdscale` *und* `sammon` *(für Letzteres mit* `library(MASS)` *das Statistikpaket laden!) und produzieren Sie je eine Grafik. Die Landessprachennamen tragen Sie bitte mit der Grafikfunktion* `text` *in den Plot ein.*

Abzugeben ist die Datei `zip.R` *mit Ihrem Programmcode sowie die Grafikausgabe* `zip.pdf`*.*

Aufgabe 2 — 12 P

In dieser Aufgabe geht es um ***etikettierte*** *Merkmaldaten, ihre graphische Darstellung und ihre Transformation nach Karhunen-Loéve (PCA, ME-Skript V.5).*

Wir stellen Merkmaldaten in 'R' als `data.frame` *mit $N+1$ Spalten dar; jede Zeile entspricht einem Muster; die Spalten $1, 2, \ldots, N$ enthalten die Merkmalwerte (Typ* `numeric`*) und die letzte Spalte zeigt die wahre Klassenzugehörigkeit (Typ* `factor`*) an.*

(a) *Laden Sie den Irisdatensatz mit dem Kommando* `data(iris)` *und lesen Sie die sieben Datensätze aus* `load('pca.rda')` *($\rightsquigarrow$ Aufgabenwebseite) ein.*

(b) *Schreiben Sie eine Grafikausgabefunktion* `plot.ldf(x, ...)` *zur Scatterplotdarstellung (siehe* `?plot.data.frame`*) der multivariaten Datensätze. Die Punkte der Zeichnung sind nach Klassenzugehörigkeit einzufärben. Es bezeichne* `x` *den Datensatz (mit Klassenfaktor in der letzten Spalte) und* `...` *die an* `plot` *zu delegierende Restparameterliste.*

(c) *Testen Sie* `plot.ldf()` *mit den vier Teildatensätzen* `iris[c(1:j,5)]` *für $j = 1, 2, 3, 4$.*

(d) *Schreiben Sie nun eine Konstruktorfunktion* `PCA (x, n=?)`*, die ein Objekt der Klasse* `PCA` *erzeugt mit den Listenelementen* `mean`*,* `eigenval`*,* `eigenvec` *für den Mittelwertvektor und die ersten* `n` *Eigenwerte bzw. Eigenvektoren des Datensatzes* `x`*. In der Voreinstellung für Argument* `n` *sollen* ***alle*** *Hauptachsen eingespeichert werden.*

(e) *Dann schreiben Sie für die neue Klasse eine Funktion* `predict.PCA (o, newdata)`*, die auf den Eingabedatensatz* `newdata` *die (i.a. unvollständige) Hauptachsentransformation des* `PCA`*-Objekts* `o` *anwendet. Die Eingabevektoren $\boldsymbol{x}$ sind also gemäß der Skriptformel $\boldsymbol{D}^{-1/2} \cdot \boldsymbol{U}^\top \cdot (\boldsymbol{x} - \boldsymbol{\mu})$ zu zentrieren, zu rotieren und dann zu skalieren.*
Achtung! Die Eingabe `newdata` *und auch die Rückgabe sind* `data.frame`*-Objekte; der Klassenfaktor ist von Eingabe zu Ausgabe durchzuschleusen!*

Brauchbare 'R'-Funktionen für den Konstruktor sind `colSums`*,* `cov` *und* `eigen`*, für den Prädiktor z.B.* `apply`*,* `sweep` *oder* `scale`*.*

(f) *Erzeugen Sie nun zur Kontrolle die Grafikausgaben* `plot.ldf(predict (PCA (iris,j), iris))` *mit $j = 1, 2, 3, 4$ für die vier möglichen (un)vollständigen Transformationen.*

(g) *Starten Sie eine (2×2)-Leinwand und visualisieren Sie nun die Transformierten* `predict (PCA (ldata, n=2), tdata)`*. Für* `ldata` *und* `tdata` *setzen Sie wahlweise* `iris` *ein und den Teildatensatz* `iris.part`*, der lediglich die 50* `versicolor`*-Muster enthält.*

(h) *Starten Sie nun eine Schleife über die acht Datensätze (inklusive* `iris`*) mit je einer (2×2)-Leinwand und den vier Scatterplots für (1) die beiden ersten Originalmerkmale, (2) die beiden letzten Originalmerkmale, (3) die beiden ersten Hauptkomponentenmerkmale, (4) die beiden letzten Hauptkomponentenmerkmale.*

Abzugeben ist die Datei `pca.R` *mit Ihrem Programmcode.*

Hinweise zum Übungsablauf

- Die wöchentliche WMM-Vorlesung findet am Mittwoch um 12:15 Uhr statt.
  Das Aufgabenblatt gibt es immer am Freitag (PDF im Netz).
  Der späteste Abgabetermin ist Sonntag 23:59 Uhr.
- Die Übungsaufgaben dürfen natürlich (und sollten sogar) in Gruppenarbeit (2 Mitglieder) gelöst werden.
- Schriftliche Lösungen (*„Textantworten“*) sind als PDF beizufügen oder direkt im e-Mail-Textkörper unterzubringen.
- Alle anderen Lösungen (Programmieraufgaben, Daten und Grafiken) sind als elektronischer Anhang der Lösungs-e-Mail abzuliefern.
- Programmcode (Dateien `*.R`) muss auch wirklich in 'R' ausführbar sein.
  (Kommando `Rscript «name.R»` auf einem der Rechner des FRZ-Pools)
- Ganz wichtig:
  Schriftliche Antworten werden von mir gedruckt, gelesen, kommentiert und korrigiert. Deshalb diese Textteile bitte **niemals** im abgegebenen Programmcode verstecken!
- Je Gruppe und je Aufgabenblatt ist **genau eine** e-Mail zu senden:
  - Vermerk »**WMM**/n« und Gruppenname im `subject`-Feld
    ($n \in \mathbb{N}$ ist die laufende Nummer des Übungsblattes)
  - die Namen der beteiligten Gruppenmitglieder im Textrumpf
  - Tabellen, Bilder, Programmcode, Sensordaten als Attachments
    (elektronische Anlagen)
  - etwaige schriftliche Antworten im Textrumpf der Post oder als Attachment
    (Text/PDF)
- Einige Aufgabentexte verweisen Sie zum Nachschlagen von Details auf das Folienskript zur Vorlesung Mustererkennung; Sie finden es unter der URL
  `http://www.minet.uni-jena.de/fakultaet/schukat/ME/Scriptum/`.
  Die Angabe *ME-Skript II.6* bedeutet: Kapitel II, Abschnitt 6