WMuEMaL

Friedrich-Schiller-Universität Jena

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

WERKZEUGE MUSTERERKENNUNG & MASCHINELLES LERNEN

Aufgabenblatt 7

(Ausgabe am Fr 1.6.2018 — Abgabe bis So 10.6.2018)

Aufgabe 1 — 12 P

Die Anwendung der PCA auf Grauwertbilder erfordert wegen der enormen Eingangsdimension (Anzahl der Bildpixel) einen besonderen Kunstgriff (Singulärwertzerlegung, SVD, ME-Skriptum V.7) bei der Implementierung.

(a) *Laden Sie die Bilddaten aus* `faces.rda` *— je 9 Porträts (Bogart&Merkel&Gysi) im* (120×160)*-Raster — in eine benamte 27-er Liste und zeichnen je 9 davon auf eine* 3×3*-Seite. Verwenden Sie* `mapply` *und* `plot.array`*.*

(b) *Implementieren Sie eine kleine Hilfsfunktion* `impack(x)`*, welche zur Bilderliste* `x` *eine breite Datenmatrix mit je einer Zeile pro Eingabebild erzeugt, die alle Bildpixel in Folge enthält.*

(c) *Implementieren Sie eine zweite Hilfsfunktion* `plot.flatpic(x,dim,...)`*, die den Pixelvektor* `x` *in eine Matrix mit Ausmaßen* `dim` *zurückwandelt, die Einträge mittels* $(x-\min)/(\max-\min)$ *auf das Grauwertintervall* $[0,1]$ *abbildet und anschließend (unter Weitergabe der Restparameter* `...`*) mit* `plot.array` *zeichnet.*

(d) *Testen Sie die beiden Funktionen, indem Sie die Bildausgabe (a) unter Verwendung der Bildstapelmatrix* `impack(x)` *wiederholen.*

(e) *Jetzt wird es ernst! Implementieren Sie eine Funktion* `eigenface(X,n=nrow(X))`*, die zur Datenmatrix* `X` *(Zeilen = Mustervektoren) die ersten* `n` *Hauptachsen und Hauptkomponenten berechnet.*

> *TIPP: Das traditionelle Vorgehen, mittels* `eigen()` *die Eigenwertaufgabe für die Kovarianzmatrix* $\boldsymbol{S} = {}^{1}\!/_{T}\cdot\boldsymbol{X}^{\top}\boldsymbol{X}$ *der* ***zentrierten*** *Datenmatrix — für mittelwertbehaftete Daten gilt diese vereinfachte Formel nicht! — zu lösen, ist für hochdimensionale Daten nicht praktikabel. Nutzen Sie deshalb die Singulärdarstellung* $\boldsymbol{X}=\boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^{\top}$ *der Datenmatrix und die resultierende Beziehung* $T\cdot\boldsymbol{S}=\boldsymbol{U}\boldsymbol{D}^{2}\boldsymbol{U}^{\top}$*, um die Hauptachsen von* $\boldsymbol{S}$ *und die Hauptkomponentendarstellung von* $\boldsymbol{X}$ *über einen effizienteren Umweg (es gilt* $\boldsymbol{X}\boldsymbol{U}=\boldsymbol{V}\boldsymbol{D}$*) zu berechnen!*

Lesen Sie sorgfältig `?svd` *durch. Resultat sei die Liste* `list(PC,PA,singval)` *mit den Hauptachsen als Spalten der Matrix* `PA`*, der transformierten Eingabe in der Matrix* `PC` *und den* `n` *Singulärwerten im Vektor* `singval`*.*

(f) *Rufen Sie* `eigenface` *mit der Bildstapelmatrix auf und visualisieren Sie die Singulärwerte in einem* `barplot`.

(g) *Entnehmen Sie dem Eintrag* `PC` *die beiden ersten Hauptkomponenten und zeichnen die 27 Bilder als PC_1/PC_2-Punkte in die Ebene. Setzen Sie den Bildnamen neben die Punktmarkierung — siehe 'R'-Funktion* `text()`. *Wiederholen Sie die Grafikausgabe für die Hauptkomponenten PC_3/PC_4, dann PC_5/PC_6 usw. bis PC_{23}/PC_{24}. Wählen Sie ein 3×3-Layout für diese 12 Grafiken.*

(h) *Entnehmen Sie dem Eintrag* `PA` *die 27 Hauptachsen („Eigenfaces") und visualisieren sie mit* `plot.flatpic` *auf drei 3×3-Seiten.*

(i) *Wiederholen Sie dieses mit (den Zeilen!) der Datenmatrix* `PC %*% t(PA)` *und erklären Sie das Resultat.*

Abzuliefern ist ist die Datei `faces.R` *mit Ihrem Programmcode und die Antwort zu Teil (i).*

Aufgabe 2 — 8 P

Wir implementieren Lern- und Testphase eines einfachen statistischen Klassifikators — der naiven Bayesregel mit klassenweise normalverteilten Merkmalen (ME-Skript VI.4 und VII.2).

(a) ***Lernphase:*** *Die Konstruktorfunktion* `naivegauss(x)` *erwartet einen Lerndatensatz* `x` *(Klasse* `data.frame`*) mit der Etikettierung (Klasse* `factor`*) in letzter Position. Sie erzeugt ein Listenobjekt der Klasse* `naivegauss`*, das alle nötigen Informationen zur Klassifikation enthält, also z.B. die Klassenwahrscheinlichkeiten und die gelernten Normalverteilungsparameter.*

(b) ***Abrufphase:*** *Die Funktion* `predict.naivegauss(o,newdata)` *erwartet ein Listenobjekt* `o` *der Klasse* `naivegauss` *sowie einen Testdatensatz* `newdata` ***ohne*** *Etikettierung. Sie retourniert einen Faktorvektor, der zu jedem Eingabemuster (Zeilenvektoren von* `newdata`*) die geratene Klasse enthält.*

HINWEIS: Stellen Sie sicher, dass `predict` *auch unter Extrembedingungen (Datensätze mit einem Merkmal und/oder einem Muster) funktioniert!*

(c) ***Fehlertest:*** *Die Funktion* `heldout (x, newdata=x, method, ...)` *erwartet je einen etikettierten Lern- und Testdatensatz. Sie lernt aus* `x` *und klassifiziert damit* `newdata`*. Dabei verwendet sie das Klassifikationsverfahren, das in der 'R'-Klasse* `method` *(mit gleichnamigem Konstruktor, dem wir auch* `...` *weiterleiten) implementiert ist. Nach Vergleich mit den wahren Klassenzugehörigkeiten der Testmuster liefert sie die (geschätzte) Fehlerwahrscheinlichkeit als Rückgabewert. Diesem* `numeric[1]`*-Objekt sei als Attribut (Name:* `confused`*) die Matrix der absoluten Klassenverwechslungshäufigkeiten beigefügt.*

(d) *Laden Sie die Iris-Daten und starten Sie* `heldout (iris, iris, naivegauss)`*. Die Reklassifikationsfehlerrate sollte 4 Prozent (6/150) betragen.*

(e) *Lesen Sie die Datensätze* `vehicle.lern` *und* `vehicle.test` *ein. Starten Sie alle vier möglichen Aufrufkombinationen (Lern/Test) von* `heldout()` *für diese Daten. Erklären Sie, inwiefern die Größenrelationen zwischen den Fehlerraten der vier* `vehicle`*-Läufe exakt Ihren Erwartungen entsprechen (ME-Skript VI.7).*

Abzugeben sind die Datei `naivegauss.R` *mit dem Programmcode sowie schriftlich die $5 = 1+4$ Fehlerraten zu (d,e) und der Kommentar zu (e).*

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