WMuEMaL

Friedrich-Schiller-Universität Jena

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

WERKZEUGE MUSTERERKENNUNG & MASCHINELLES LERNEN

Aufgabenblatt 8

(Ausgabe am So 16.6.2019 — Abgabe bis So 23.6.2019)

Aufgabe 1 — 10 P

Wir implementieren eine Klasse `parzen` *für eine* ***univariate*** *Parzenschätzung (ME-Skriptum VI.6, Blatt 12–13) mit je einer Gaußglocke* $\mathcal{N}(x \mid z_i, s^2)$ *als Potentialfunktion (Skalenfaktor* s*) für Lernprobenwerte* $z_1, \ldots, z_n \in \mathbb{R}$.

(a) *Schreiben Sie einen Konstruktor* `parzen(x,sd)`*, der ein Objekt der Klasse* `parzen` *mit Komponenten* `o$support` *und* `o$sigma` *für Lernprobe und Skalenfaktor abliefert.*

(b) *Schreiben Sie eine Abrufmethode* `predict.parzen(o,newdata=NULL)`*, die den Vektor der Dichtewerte des Parzenobjekts* `o` *für die Eingabedaten des Vektors* `newdata` *zurückgibt. Verwenden Sie dafür die 'R'-Implementierung* `dnorm()` *der Gaußdichte!*

(c) *Schreiben Sie eine Funktion* `plot.parzen(o,xlim=?,...)` *zur Grafikdarstellung der Parzendichte* `o` *im Intervall* `xlim`*. Verwenden Sie* `curve()` *und zur Fransendarstellung der Lernprobewerte* $z_1, \ldots, z_n$ *die Funktion* `rug()`*. Die* `xlim`*-Voreinstellung wähle einen sinnvollen Bereich um alle Stützstellen. Den Skalenfaktor* s *platzieren Sie bitte an der Grafiknordseite.*

(d) *Laden Sie jetzt* `parzen.rda` *und zeichnen Sie den Parzendichteverlauf der Datenprobe* `samples` *für alle* `sd`*-Werte* s^m *mit* $m \in \mathbb{Z}$ *zwischen* 6 *und* -5 *und der Basis* $s = 3/4$ *(2 Grafikseiten im Format* 3×2*).*

(e) *Ergänzen Sie* `predict.parzen`*, so dass im Fall* `newdata=NULL` *der Vektor aller Leave-One-Out-Dichtewerte für die Stützstellen in* `o$support` *berechnet und zurückgegeben wird. (Der Dichtewert für* z_j *wird auf der Basis der Parzendichte mit den Stützstellen* $\{z_1, \ldots, z_n\} \setminus \{z_j\}$ *ermittelt.)*

(f) *Ergänzen Sie* `plot.parzen`*, so dass auch die oben implementierten* L^1O*-Dichtewerte als graue Balken in die Grafik einbezogen werden. Wiederholen Sie die Grafikaufrufe aus (d).*

(g) *Ergänzen Sie den Konstruktor* `parzen`*, so dass im Fall* `sd=NULL` *der Skalenfaktor mit maximaler (logarithmierter!)* L^1O*-Zielgröße (Produkt der* L^1O*-Dichtewerte aller Stützstellen) berechnet und verwendet wird. Realisieren Sie die Maximierung durch einen geeigneten Aufruf der 'R'-Funktion* `optimize()`*. (Die mitgelieferte Variante* `Optimize()` *erzeugt bei Bedarf eine Grafikausgabe des Suchprozesses.)*

(h) Testen Sie Ihre Implementierung mit dem Grafikaufruf `plot(parzen(samples))`.

Abzuliefern ist bitte Ihr Programmcode in `parzen.R`.

Aufgabe 2 — 10 P

Laden Sie das 'R'-Paket `class` *mit dem Kommando* `library(class)` *und lesen Sie sich die Beschreibung zu den Methoden* `knn` *und* `knn.cv` *des Nächste-Nachbarin-Klassifikators (ME-Skript VI.6, Blatt 14,15) durch, für deren etwas hausbackene Schnittstelle wir im Folgenden einige einfache Hüllfunktionen schreiben werden.*

(a) Schreiben Sie eine 'R'-Konstruktorfunktion `kNN(x,neighbours=1)` *für einen k-NN-Regel-Klassifikator mit Lerndaten* `x` *und* `neighbours` *nächsten Nachbarn. Rückgabe ist ein Listenobjekt der Klasse* `kNN` *mit den benötigten Daten und Parametern.*

(b) Schreiben Sie eine 'R'-Prädiktorfunktion `predict.kNN(o,newdata)`*, welche die Zeilenvektoren der Datenmatrix* `newdata` *(Matrix oder Dataframe; ohne Faktor!) mit der k-NN-Regel* `o` *klassifiziert. Rückgabe ist der Klassenfaktor.*

(c) Erweitern Sie `predict.kNN()`*, so dass bei Aufruf mit* `newdata=NULL` *die Leave-One-Out-Klassifikation der Lerndaten des* `o`*-Objekts berechnet wird. Konsultieren Sie* `?knn.cv`.

(d) Reanimieren Sie die Auswertefunktion `heldout(x,newdata=x,method,...)` *vom letzten Aufgabenblatt und modifizieren Sie ihren 'R'-Kode, so dass bei Aufruf mit* `newdata=NULL` *die Leave-One-Out-Fehlerrate des* `method`*-Klassifikators für die* `x`*-Daten berechnet wird.*

(e) Programmieren Sie einen Testlauf `run.1st(x,y,choice=1+2*0:8)`*, der eine fünfzeilige Matrix von Fehlerraten erzeugt. In Spalte* `j` *wird die k-NN-Regel mit* `choice[j]` *Nachbarn getestet. In Zeile 1 wird* `x` *zum Lernen und* `y` *zum Testen genutzt. In Zeile 2 werden die Rollen von* `x` *und* `y` *getauscht. In Zeilen 3–5 wird wiederholt die Leave-One-Out-Fehlerrate für die Vereinigungsmenge von* `x` *und* `y` *ermittelt.*

(f) Programmieren Sie einen Testlauf `run.2nd(x,y,choice=2^(0:13))`*, der einen Vektor von Fehlerraten erzeugt. In Komponente* `j` *stehe die Fehlerrate der 1-NN-Regel mit Testdaten* `y` *und den ersten* `choice[j]` *Mustern von* `x` *zum Lernen.*

(g) Programmieren Sie einen Testlauf `run.3rd(x,choice=2:ncol(x)-1)`*, der einen Vektor von Fehlerraten erzeugt. In Komponente* `j` *stehe die Leave-One-Out-Fehlerrate der 1-NN-Regel für die Daten* `x`*, wobei alle Attribute außer einem — dem „Knock-out"-Attribut* `choice[j]` *— als Merkmalsatz zur Klassifikation genutzt wurden.*

(h) Laden Sie jetzt die drei Datensätze `vehicle`*,* `letter` *und* `australia` *aus den* `*.rda`*-Dateien und führen Sie damit (in obiger Reihenfolge zugeordnet) die drei Testreihen durch. Für die* `australia`*-Studie werden Lern- und Testdatenteil vereinigt und an* `x` *übergeben. Speichern Sie die drei Fehlertabellen mit* `save(pe.1,pe.2,pe.3,file='kNN.rda')` *ab.*

(i) Erzeugen Sie abschließend vier `barplot`*-Grafiken, zwei für die* `vehicle`*-Fehlermatrix und je eine für die beiden Fehlervektoren zu* `letter` *und* `australia`*. Gestaltungsvorschläge siehe Ausgabebeispiel* `kNN-bsp.djvu`*. Für eine ansprechende Darstellung ist darauf zu achten, dass die Funktionen aus (e,f,g) informative Beschriftungen in* `colnames` *und* `rownames` *ablegen.*

Abzugeben sind der R-Code `kNN.R` *und die Fehlertabellen in* `kNN.rda`.

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