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 Fr 9.6.2017 — Abgabe bis So 18.6.2017)

Aufgabe 1 — 10P

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,sigma)`*, 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)`*, der 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* `rugs()`*. 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* `sigma`*-Werte* s^m *mit* $m \in \mathbb{Z}$ *zwischen* 12 *und* -5 *und der Basis* $s = 0.8$ *(3 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 werden. (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 mit* `points()` *in die Grafik einbezogen werden. Wiederholen Sie nun die Grafikaufrufe aus (d).*

(g) *Ergänzen Sie den Konstruktor* `parzen`*, so dass im Fall* `sigma=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 — 10P

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 (train, test=train, method, ...)` *erwartet je einen etikettierten Lern- und Testdatensatz. Sie lernt aus* `train` *und klassifiziert damit* `test`*. 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.*

(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).*

(f) ***Kreuzvalidierung:*** *Schreiben Sie eine Funktion* `leave1out (x, method, ...)`*, welche die „leave-one-out"-Fehlerrate eines Datensatzes* `x` *berechnet. Wie* `heldout` *soll auch* `leave1out` *für jeden syntaktisch wie* `naivegauss` *ausgelegten Klassifikatortyp* `method` *anwendbar sein.*

(g) *Wie groß ist der* L^1O*-Fehler für die Iris-Daten? (Tipp:* 7/150*) Und für die* `vehicle.test`*-Daten? Und welches Phänomen beobachten Sie beim Datensatz* `rbind(iris,iris)`*?*

Abzugeben sind die Datei `naivegauss.R` *mit dem Programmcode sowie schriftlich die* $8 = 1 + 4 + 3$ *Fehlerraten zu (d,e,g) und der Kommentar zu (e,g).*

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)
- *Pfingstfrieden:* Am Freitag 2.6. gibt es kein Übungsblatt. Die Lösungen für das Übungsblatt vom Freitag 26.5. müssen erst am Sonntag 11.6. abgeliefert werden.
- 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