WMuEMaL

Friedrich-Schiller-Universität Jena

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

WERKZEUGE MUSTERERKENNUNG & MASCHINELLES LERNEN

Aufgabenblatt 10

(Ausgabe am Fr 22.6.2018 — Abgabe bis So 1.7.2018)

Aufgabe 1 — 10 P

*Wir testen einen **Normalverteilungsklassifikator** (NVK; ME-Skript VII.2) und verwenden dazu die 'R'-Implementierungen* `qda()` *und* `lda()` *aus dem Paket* `MASS` *('R'-Standarddistribution).*

(a) *Schreiben Sie einen Konstruktor* `GDC(x)`*, der einen NVK aus dem Datensatz* `x` *lernt (*`qda()`*-Aufruf!) und ein geeignetes 'R'-Objekt mit Klassenkennung* `GDC` *zurückliefert.*

(b) *Schreiben Sie die zugehörige Abrufmethode* `predict.GDC(o,newdata)`*, welche den (unetikettierten) Datensatz* `newdata` *mit Hilfe des NVK im* `GDC`*-Objekt* `o` *klassifiziert und einen entsprechenden Faktor zurückgibt.*

(c) *Reanimieren Sie die* `heldout()`*-Funktion (Aufgabe 7.2c), laden Sie den Datensatz* `satimage` *und berechnen Sie für alle neun Kombinationen aus Testprobe, Lernprobe und deren Vereinigung die Fehlerrate. Implementieren Sie dazu* `heldout.KxK(data,...,plot=TRUE)`*, eine Funktion, die zur Datensatzliste* `data` *die resultierende Fehlerratentabelle (k^2 Einträge für Listenlänge k; Restparameter* `...` *werden an* `heldout` *delegiert) abliefert.* TEST LERN ALLE

(d) *Erweitern Sie nun Ihre* `GDC()`*-Implementierung um das Argument* `linear`*. Bei Aufruf mit* `linear=TRUE` *soll ein NVK mit **linearen** statt quadratischen Diskriminanten gelernt werden — das tut die* `MASS`*-Methode* `lda()` *für uns.*

(e) *Jetzt implementieren Sie die Funktion* `bootstrap(x,k,...)`*, welche* `k` *zufällige Bootstrapfehlerraten (ME-Skript XI.4) zum Datensatz* `x` *berechnet und als Vektor abliefert. Je Rate werden zum Lernen* `nrow(x)` *zufällig mit Zurücklegen aus* `x` *gezogene (siehe* `?sample`*) Musterzeilen verwendet. Zum Test werden alle* `x`*-Zeilen genutzt, die nicht beim Lernen dabei waren. Die Restargumente* `...` *gehen wieder an* `heldout`*.*

(f) *Abschließend laden Sie die vier Datensätze* `heart`*,* `satimage`*,* `vehicle`*,* `diabetes` *und erzeugen je eine Grafikseite mit drei Fehlerplots: je einen* `barplot` *(Teil (c)) für die neun Fehler des quadratischen und des linearen NVK sowie einen vierteiligen* `boxplot` *mit den Bootstrapresultaten (je $k = 32$ Ziehungen) für die lineare (alle drei Proben) und die quadratische (nur die* `ALLE`*-Probe) Variante des NVK.* TEST LERN ALLE ALLE

Abzugeben sind Ihre Programmcodedatei `GDC.R` *und schriftlich die beiden (3×3)-Tabellen für* `satimage`*.*

Aufgabe 2 — *10 P*

Wir schreiben 'R'-Funktionen zur Klassifikation mit einem ***Mehrschichtenperzeptron*** *(MLP; ME-Skript VIII.5).*

(a) Laden Sie per Kommando `library(nnet)` *das 'R'-Paket zum Lernen und Testen von MLPs mit* ***einer*** *verborgenen Schicht künstlicher Neuronen. Studieren Sie die Beschreibungstexte zur Lernmethode* `nnet` *und zur Vorhersagemethode* `predict.nnet`*. Beachten Sie die Aufrufbeispiele, die Hinweise zur Gestaltung von* `formula`*-Objekten und die Informationen zur Verwendung des MLP zu Klassifikationszwecken (Zielvariable=*`factor`*).*

(b) Schreiben Sie eine Methode `SHLP(x,hidden,norm=FALSE)` *zum Lernen eines MLP mit* $H=$`hidden` *verborgenen Neuronen aus den etikettierten Daten* `x`*. Nutzen Sie dazu die* `nnet`*-Methode unter Beibehaltung aller Defaulteinstellungen.*

(c) Schreiben Sie eine Methode `predict.SHLP(o,newdata)` *zur Vorhersage der Klassennamen (*`factor`*!) für die nicht etikettierten Daten* `newdata`*. Der 'R'-Code passt quasi in eine Zeile!*

(d) Wiederbeleben Sie Ihre Funktion `heldout()` *(Aufgabe 7/2c) und tätigen Sie nun einige Testaufrufe. Die erbrachten Resultate wären leider nicht reproduzierbar, weil die Startgewichte vom Lernverfahren* `nnet` *in Werkseinstellung mit Zufallszahlen vorbesetzt werden. Korrigieren Sie diesen Missstand, indem Sie* `nnet` *mittels Aufrufargument* `Wts = sin(10*1:m)` *von den Vorteilen deterministisch vorgegebener Startwerte überzeugen. Damit das auch reibungslos funktioniert, müssen Sie allerdings die Anzahl* `m` *zu lernender MLP-Gewichte kennen . . .*

(e) Entwickeln Sie daher eine Formel $m = \rho(D, H, K)$ *zur Berechnung der Gewichteanzahl* m *aus der Merkmaldimension* D*, der Anzahl* H *verborgener Neuronen und der Anzahl* K *der Musterklassen. (Vergessen Sie nicht die konstanten Schwellwertneuronen und die außerplanmäßige Modellierung von Zweiklassenproblemen!)*

(f) Sobald die reproduzierbare Version funktioniert, berechnen Sie bitte die Test- und die Reklassifikationsfehlerraten für die Datensätze `australia`*,* `diabetes`*,* `segment` *und* `vehicle` *für MLPs mit* $H \in \{1, 3, 6, 10, 15, 21, 28\}$ *verborgenen Neuronen.*

(g) Künstliche Neuronale Netze sind dafür berüchtigt, dass sie sensibel auf die Skalierung ihrer Eingabedaten reagieren. Nutzen Sie den Schalter `norm`*, um eine Verfahrensvariante zu realisieren, die Lern- und Testdaten durch eine* ***gemeinsame*** *Lineartransformation standardisiert. Nutzen Sie die 'R'-Methode* `scale()`*, um die Lerndatenmerkmale auf* $\mu = 0$*,* $\sigma = 1/3$ *zu normieren.*

(h) Wiederholen und tabellieren Sie nun die obige Testreihe. Erzeugen Sie aus Ihren Resultaten je Datensatz eine Grafik mit den vier Fehlerkurven (Lernfehler/Testfehler $\times$ *rohe/normierte Daten).*

Abzugeben sind bitte der 'R'-Programmcode `SHLP.R` *sowie die Gewichtanzahlformel (e) und die 2 Tabellen aus (f,h) als schriftliche Lösungskomponenten.*

Aufgabe 3 Bonuspunkte: max. 10 P

Unter Merkmalauswahl versteht man die Bestimmung einer geeigneten Merkmalteilmenge, die möglichst viel Klasseninformation und möglichst wenig Datenredundanz enthält. Für diese „Freistil"*-Aufgabe sollen Sie unter den vielen möglichen Merkmalkombinationen (D Merkmale besitzen* 2^D *Untermengen) eine mit möglichst niedriger Fehlerrate finden.*

(a) *Laden Sie den bioinformatischen Lerndatensatz aus* `data1.rda`*. Er besteht aus* 2000 *Mustern mit je* 180 *binären Merkmalen und der Klassenetikettierung (*3 *Klassen).*

(b) *Gesucht ist eine Teilmenge* $\mathcal{S} \subset \{x_1, \ldots, x_{180}\}$*, bei deren Verwendung ein Nächster-Nachbar-Klassifikator (*`class::knn`*, Einstellung* $k = 1$*) mit den geladenen Lerndaten* `data1` *eine möglichst geringe Fehlerrate auf den zukünftigen Testdaten* `data2` *(die liegen beim Dozenten im Tresor) erzielt.*

(c) *Lassen Sie Ihrer Kreativität freien Lauf und schreiben Sie ein Programm zur Berechnung einer guten Merkmalteilmenge* $\mathcal{S}$*. Alle Hilfsmittel sind erlaubt! Typische gute Lösungen bestehen aus acht bis sechzehn Merkmalen.*

(d) *Die Namen der* $\mathcal{S}$*-Merkmale schreiben Sie in einen* `character`*-Vektor. Sie dürfen bis zu maximal drei Lösungskandidaten als Liste von* `character`*-Objekten einsenden.*

Abzugeben ist der 'R'-Code `subset.R` *ihrer Implementierung, eine Datei* `subset.rda` *mit der ge*`save`*ten Namensvektorliste und ein kurzer Kommentar (Text) zu Ihrer Vorgehensweise.*

Die Lösung zur Freistil-Aufgabe 3 muss erst bis spätestens zum Sonntag, den 8.7.2018, abgeliefert werden.

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