Einführung in die Bioinformatik

Prof. Dr. Enno Ohlebusch, Prof. Dr. Hans Kestler, Dr. Karlheinz Holzmann,
Tobias Badura

WS 16/17

Übungsblatt 6

Abgabe bis 15.02.2017, 08:30. Lösungen bitte elektronisch (an tobias.badura@uni-ulm.de) mit Namen in Text und Quelltextdateien abgeben.

Am 15.02.17 findet, statt der Vorlesung, die Übung statt.

1. Aufgabe (10): Nächster-Nachbar-Klassifikator

Ein Klassifikator weist einem neuen unbekannten Muster eine Klasse aus vorher gelernten Daten (z.B. Gen-Expressions-Profile von Zellen bekannter Klasse) zu. Um einen Klassifikator auf seine **Generalisierungsleistung** (d.h. wie gut kommt der Klassifikator mit neuen, ungelernten Daten zurecht) zu überprüfen, werden die Daten oft in einen **Trainingsdatensatz** (TR) und einen davon unabhängigen **Testdatensatz** (TE) aufgeteilt. Mit TR wird der Klassifikator trainiert und dann die Klassifikationsleistung (d.h. die Anzahl der Fehler - falsch zugewiesene Labels) auf TE gemessen.
Einer der einfachsten und oft sehr gut funktionierenden Klassifikatoren ist der *one nearest neighbor* Klassifikator, der einem unbekannten Muster die Klasse des nächsten Datenpunktes in TR zuweist.
Implementieren Sie den *one nearest neighbor* (1-NN) Klassifikator in R. Verwenden Sie die Euklid'sche Norm als Abstandsmaß. Testen Sie das R-Skript mit dem Golub-Datensatz von der Vorlesungswebseite (Training mit `golub50.train` und Test mit `golub50.test`). Die letzte Spalte enthält die Klassenbezeichnung $(0, 1, 2)$. Geben Sie die Anzahl der falsch klassifizierten Datenpunkte auf der Trainings- und der Testmenge aus.
Hinweise:
Der 1-NN weist einem Abfrage-Datenpunkt $\mathbf{x} \in \mathbb{R}^n$ die Klasse des (bzgl. des spezifizierten Abstandsmaßes d) nächsten Nachbarn aus der Trainingsmenge $(\mathbf{x}^\mu, y^\mu) \in \mathbb{R}^n \times \{1, 2, \ldots, c\}$ zu ($\mu = 1 \ldots m$, c die Anzahl der Klassen), d.h. die Klasse y^{μ^*} mit

$$\mu^*(\mathbf{x}) = \arg \min_{\mu=1\ldots m} d(\mathbf{x}, \mathbf{x}^\mu) \quad .$$

Für den Fall des Euklid'schen Abstandes gilt $d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} (x_i - y_i)^2$ mit $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

Teilen Sie für diese Aufgabe den Klassifikator am besten auf in

1. eine **Trainingsfunktion** `classifier <- train.1nn(X,y)`, die eine Zeilenmatrix `X`$\in \mathbb{R}^{m \times n}$ (je ein Trainings-Datenpunkt pro Zeile) und einen Labelvektor `y`$\in \mathbb{R}^m$ akzeptiert und einen Klassifikator zurückgibt (beim 1-NN besteht das Training nur darin sich die Datenpunkte zu merken - z.B. in einer benannten Liste `list(train=X,labels=y)`), und in
2. eine **Testfunktion** `y <- predict.1nn(classifier,x)`, die einem n-dimensionalen Datenpunkt (Vektor) `x` mittels des trainierten Klassifikators `classifier` eine Klasse y zuweist.

2. Aufgabe(10): Single Threshold Classifiers (STC)

Der STC ist ein einfaches Beispiel eines binären Klassifikationsmodels. Es berechnet die Klasse einer Probe (z.B. Patient) $\mathbf{x} = (x^{(1)}, \ldots, x^{(n)})^T \in \mathbb{R}^n$, in Bezug auf ein einziges Merkmal.

$$c(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad d(x^{(i)} - t) \geq 0 \\ 0 & \text{else} \end{cases} \tag{1}$$

Hier bezeichnet $i \in \{1, \ldots, n\}$ den Index des gewählten Merkmals. Das Merkmal $x^{(i)}$ wird mit einem Schwellwert $t \in \mathbb{R}$ verglichen. Wenn es größer ist als t, wird die Probe als Klasse 1 kategorisiert. Der Parameter $d \in \{-1, +1\}$ wird genutzt um die Richtung des Klassifizieres zu wechseln.

- Wenn wir ein neues Klassifikationsmodel für eine neue (unbekannte) Aufgabe trainieren wollen, müssen die Parameter i, t, d anhand von Beispielen angepasst werden. Trainieren sie ihre Klassifikationsmodelle mit den Trainingsdaten ($\mathcal{S}_{tr}$) von Beispiel A (Tabelle 1) und B (Tabelle 3). Diese Klassifikatoren sollen folgendes minimieren

$$\min_{i,t,d} \sum_{(\mathbf{x},y)\in\mathcal{S}_{tr}} \mathbb{I}_{[c(\mathbf{x})=y]} \tag{2}$$

- Der Erfolg eines Klassifikators wird an seperaten Daten getestet. Testen sie die trainierten Klassifikationsmodelle an den Testdaten ($\mathcal{S}_{te}$) von Beispiel A (Tabelle 2) und B (Tabelle 4).

$$R_{emp} = \sum_{(\mathbf{x},y)\in\mathcal{S}_{te}} \mathbb{I}_{[c(\mathbf{x})=y]} \tag{3}$$

- STCs können zur Merkmalselektion eingesetzt werden. Diese können, zum Beispiel, über den minimalen Trainingsfehler bewertet und eingestuft werden. Bewerten sie die Merkmale von Beispiel B. Zeichnen Sie ein zweidimensionales Koordinatensystem für jedes Paar von Merkmalen. Ist die kombination der zwei besten Merkmale im zweidimensionalen Fall optimal?

$\mathcal{S}_{tr}$	x	y
$\mathbf{x}_1$	2	0
$\mathbf{x}_2$	3	0
$\mathbf{x}_3$	4	0
$\mathbf{x}_4$	5	0
$\mathbf{x}_5$	1	1
$\mathbf{x}_6$	6	1
$\mathbf{x}_7$	7	1
$\mathbf{x}_8$	8	1

Tabelle 1: Szenario A: Trainingsdaten $\mathcal{S}_{tr}$

$\mathcal{S}_{te}$	x	y
$\mathbf{x}_1$	2	0
$\mathbf{x}_2$	1	0
$\mathbf{x}_3$	4	0
$\mathbf{x}_4$	5	0
$\mathbf{x}_5$	3	1
$\mathbf{x}_6$	6	1
$\mathbf{x}_7$	7	1
$\mathbf{x}_8$	8	1

Tabelle 2: Szenario A: Testdaten $\mathcal{S}_{te}$

$\mathcal{S}_{tr}$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$	$x^{(5)}$	y
$\mathbf{x}_1$	4	2	1	2	8	0
$\mathbf{x}_2$	6	3	2	4	7	0
$\mathbf{x}_3$	7	4	5	6	5	0
$\mathbf{x}_4$	8	5	6	8	3	0
$\mathbf{x}_5$	1	1	3	1	6	1
$\mathbf{x}_6$	2	6	4	3	4	1
$\mathbf{x}_7$	3	7	7	5	2	1
$\mathbf{x}_8$	5	8	8	7	1	1

Tabelle 3: Szenario B: Trainingsdaten $\mathcal{S}_{tr}$

$\mathcal{S}_{te}$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$	$x^{(5)}$	y
$\mathbf{x}_1$	4	2	1	1	8	0
$\mathbf{x}_2$	5	1	7	3	6	0
$\mathbf{x}_3$	7	4	8	5	5	0
$\mathbf{x}_4$	8	5	6	7	2	0
$\mathbf{x}_5$	1	3	3	2	7	1
$\mathbf{x}_6$	2	6	4	4	4	1
$\mathbf{x}_7$	3	7	2	6	3	1
$\mathbf{x}_8$	6	8	5	8	1	1

Tabelle 4: Szenario B: Testdaten $\mathcal{S}_{te}$