# Grundlagen des maschinellen Lernens

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Tobias Lang
Mathias Schickel

Andreas Schilling
Sommersemester 2017

## Übungsblatt 3

Ausgabe: 12.05.2017; Abgabe: bis 18.05.2017, 23:59 Uhr.

**Bayesscher multivariater Klassifikator** (34 Punkte)

Ein klassischer multivariater Datensatz beinhaltet Kenngrößen drei verschiedener Schwertlilienarten.[1] Mittels eines Bayesschen Klassifikators sollen die Daten den drei Arten zugeordnet werden, wobei zunächst von der *stochastischen Unabhängigkeit* der Kenngrößen der einzelnen Arten ausgegangen werden soll. Dazu stehen Trainings- und Testdaten zur Verfügung, und zwar einmal in den Files `trainingSetosa.csv`, `trainingVersicolor.csv` und `trainingVirginica.csv` sowie in `testSetosa.csv`, `testVersicolor.csv` und `testVirginica.csv`. Ein solcher Klassifikator soll anhand der folgenden Schritte erstellt werden.

a) Zunächst sollen die Trainings- und Testdaten für jede Schwertlilienart in Matlab eingelesen werden. Jede Spalte entspricht jeweils einer Messreihe (d. h. in den Zeilen finden sich die Messwerte) für eine der vier Kenngrößen[2]

   1. *sepale Länge* (Länge des Kelchblattes),
   2. *sepale Breite* (Breite des Kelchblattes),
   3. *petale Länge* (Länge des Kronblattes),
   4. *petale Breite* (Breite des Kronblattes).

   Für die Testdaten ist zwar bekannt, zu welcher Schwertlilienart sie jeweils gehören. Dieses Wissen darf beim Testen des Klassifikators aber selbstverständlich *ebensowenig* verwendet werden wie die Testdaten zum *Trainieren* des Klassifikators berücksichtigt werden dürfen.

b) Die Daten zu jeder Kenngröße jeder Schwertlilienart aus den *Trainings*daten sollen in einem Histogramm veranschaulicht werden. Höchstens sechs Sätze sollen zudem Auffälligkeiten in den Daten beschreiben. (4 Punkte)

c) Für jede Kenngröße jeder Schwertlilienart sind anschließend sowohl der Mittelwert als auch die Varianz für die *Trainings*daten zu bestimmen. (2 Punkte)

d) Nun sollen die jeweiligen *likelihoods* $p(x|\omega)$ für jede Schwertlilienarten bestimmt werden. Vorausgesetzt werden soll dazu, dass die einzelnen Kenndaten jeweils normalverteilt sind mit den in der letzten Teilaufgabe ermittelten jeweiligen Mittelwerten bzw. Standardabweichungen. Danach kann der *Satz von Bayes* zur Klassifikation verwendet werden. (10 Punkte)

[1] Vgl. dazu auch `http://en.wikipedia.org/wiki/Iris_flower_data_set` und `http://archive.ics.uci.edu/ml/datasets/Iris`.

[2] Zum näheren Verständnis der einzelnen Kenngrößen siehe etwa `http://de.wikipedia.org/wiki/Kelchblatt` und `http://de.wikipedia.org/wiki/Kronblatt`.

e) Für jede Schwertlilienart sind folgende Maßzahlen zu bestimmen:
   - Die Anzahl der Datenpunkte, die *korrekt* als *zur Art gehörig* klassifiziert wurden (*true positive*).
   - Die Anzahl der Datenpunkte, die *korrekt* als *nicht zur Art gehörig* klassifiziert wurden (*true negative*).
   - Die Anzahl der Datenpunkte, die *fälschlicherweise* als *zur Art gehörig* klassifiziert wurden (*false positive*).
   - Die Anzahl der Datenpunkte, die *fälschlicherweise* als *nicht zur Art gehörig* klassifiziert wurden (*false negative*). (4 Punkte)

f) Folgende Kenngrößen sind in Grafiken abzutragen und in maximal sechs Sätzen zu beurteilen:
   - Petrale Breite gegen sepale Breite,
   - petale Länge gegen sepale Länge,
   - petale Länge gegen petale Breite,
   - sepale Länge gegen sepale Breite. (4 Punkte)

g) Bislang ist für die einzelnen Kenndaten jeder Schwertlilienart stochastische Unabhängigkeit vorausgesetzt worden. Im Gegensatz dazu soll nun für die likelihood $p(\boldsymbol{x}|\omega)$ der Kenndatensätze jeder einzelnen Schwertlilienart multivariate Normalverteilung angenommen werden. Auf dieser Basis soll die Klassifikation erneut durchgeführt und die Änderung des Ergebnisses beschrieben werden. Wie sind die Änderungen zu erklären? Ein Beispielplot zur Veranschaulichung ist erwünscht. (10 Punkte)

**Hinweise zur Implementation in Matlab:**

a) Die spaltenweise Berechnung des Mittelwertes und der Varianz ist in Matlab mittels der Funktionen `mean(<Matrix>)` und `var(<Matrix>)` möglich. Das Ergebnis ist ein Vektor mit den Mittelwerten bzw. Varianzen jeder Kenngröße in der Matrix.

b) Die Funktion `normpdf( X, mu, sigma )` kann auf Matrizen `X` mit mehr als einer Spalte angewandt werden – die Spalten stehen dabei jeweils für eine Kenngröße und die Zeilen für Messwerte – und erzeugt die Werte der Wahrscheinlichkeitsdichte für die Normalverteilung mit Parametern `mu` und `sigma` an den durch die Zeilen der Matrix `X` gegebenen Messwerten. Dabei müssen `mu` und `sigma` jeweils selbst Matrizen sein, die in jeder Spalte den Mittelwert- und die Standardabweichung der zugehörigen Kenngröße enthalten. Die Zeilenzahl muss derjenigen der Matrix entsprechen. Nützlich dafür ist der Befehl `repmat`. Die Ausgabe ist eine Matrix `Y` mit den Werte der jeweiligen Dichte an den entsprechenden Stützstellen aus `X`.

c) Für die übersichtliche Gestaltung des Programmiercodes ist in Matlab an Stelle der Verwendung von Schleifen Vektor- bzw. Matrixschreibweise anzuraten – zudem läuft der Code dann performanter. Der vorangegangene Hinweis ist dafür hilfreich.

d) Die Kovarianzmatrix eines `nxd` Datensatzes `X` mit `n` Zeilen und `d` Kennzahlen lässt sich mittels des Befehls `cov(X)` berechnen.

e) Der Wert der Wahrscheinlichkeitsdichte für die multivariate Normalverteilung mit Mittelwertvektor `mu` und Kovarianzmatrix `C` an den durch die Matrix `X` gegebenen Datenpunkten kann in Matlab mit dem Befehl `mvnpdf( X, mu, C )` berechnet werden. Es ist eventuell nützlich, dazu die Matlab-Hilfe zu konsultieren.