Grundlagen des maschinellen Lernens

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Tobias Lang
Mathias Schickel

Andreas Schilling
Sommersemester 2017

Übungsblatt 5

Ausgabe: 01.06.2017; Abgabe: bis 15.06.2017, 23:59 Uhr.

Aufgabe 1 (Fragen zur Vorlesung) (7 Punkte)

Folgende Fragen sind zu beantworten:

a) Warum wendet man die *Principal Component Analysis* (PCA) an? (Zwei Gründe sollten genannt werden.)

b) Bei der *(Fisher) Linear Discriminant Analysis (LDA)* wird der Abstand gewisser Größen maximiert. Um welche handelt es sich? Warum genügt es hinsichtlich des Ziels der LDA nicht, *alleine* diese Größen bei der Maximierung zu betrachten? Erklärt kurz, inwiefern sich die Ziele der PCA und der LDA unterscheiden. (3 Punkte)

c) Was beschreibt die *within-class scatter matrix* (Foliensatz 5, Folie 17)?

d) Was beschreibt die *between-class scatter matrix* (Foliensatz 5, Folie 18)?

e) Man gebe sowohl für PCA als auch LDA an, ob es sich um ein *supervised* oder *unsupervised* Lernverfahren handelt.

Aufgabe 2 (Principal Component Analysis – 1) (8 Punkte)

Betrachtet die folgenden Plots zu jeweils zwei Klassen und Merkmalen und schätzt den ersten *principal component*-Vektor (für *alle* Datenpunkte ohne Klassenunterschied) und zeichnet ihn ein. Erklärt,

1. ob sich im Allgemeinen (d. h. unabhängig von der Klasse) eine Dimensionsreduktion auf den PC-Vektor anbietet und
2. ob dies für die Klassifikation förderlich ist oder nicht.

Die folgenden Plots werden auch als JPEG-Dateien mit dem Übungsblatt zur Verfügung gestellt.

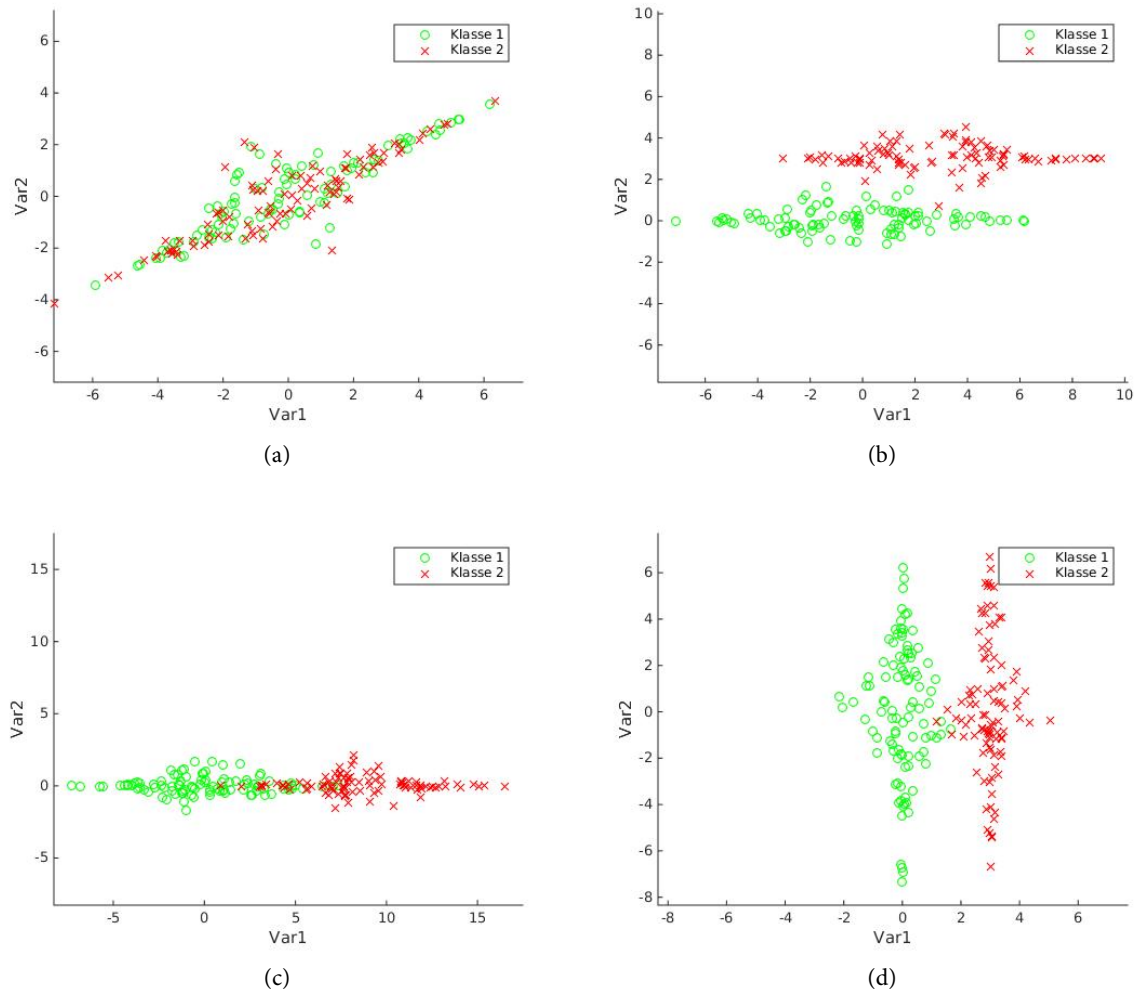

Abbildung 1: Plots

Aufgabe 3 (Principal Component Analysis – 2) (8 Punkte)

In dieser Aufgaben wird nochmals mit dem aus Übungsblatt 3 bekannten Datensatz gearbeitet. Diesmal soll allerdings nicht klassifiziert, sondern untersucht werden, welche *principal components* (PCs) den meisten Informationsgehalt liefern. Dazu sollen die folgenden Schritte befolgt werden:

a) Man lese die Trainingsdaten ein und ignoriere dabei die Klassen und behandle alle Datensätze als einen einzigen.

b) Man führe eine PCA durch. (2 Punkte)

c) Welche PCs enthalten zusammen mindestens 95 Prozent der Streuung? (2 Punkte)

d) Man projiziere die vierdimensionalen Daten in einen Raum, der mindestens 95 Prozent der Streuung enthält. (1 Punkt)

e) Das Ergebnis soll geeignet geplottet werden. (1 Punkt)

f) Ergibt die PCA in diesem Zusammenhang Sinn? Warum oder warum nicht? Die Frage kann sehr unterschiedlich beantwortet werden. (2 Punkte)

Hinweis: In Matlab kann für eine `n*m` Matrix `data` mit `n` Zeilen und `m` Spalten, wobei die Spalten Kenngrößen repräsentieren und die Zeilen Stichproben, über den Befehl `[pc ws scatter] = pca( data )` die PCA auf die Daten in der Matrix angewandt werden. Dabei liefert `pc` eine Matrix mit Spaltenvektoren, die die *principal components* repräsentieren. In `scatter` wird für jede *principal component* angegeben, wieviel Streuung der Datensatz auf die PC projiziert aufweist. Die Variable `ws` („working set“) liefert die Koordinaten der Punktmengen im projizierten Raum. Man beachte, dass zur PCA die Punktmenge im Ursprung zentriert wird. `ws` ist also *eine* Möglichkeit, die Dimensionsprojektion vorzunehmen. Es ist aber zu empfehlen, *nicht* mit der Variablen `ws` zu arbeiten: Wenn neue (Test-)Daten verwendet werden sollen, müssen diese sonst ebenfalls erst im Ursprung zentriert werden und danach reduziert werden. Eine andere Möglichkeit besteht demgegenüber darin, den ursprünglichen Datensatz auf die gewünschten PCs zu projizieren. Dann kann die Zentrierung um den Ursprung vermieden werden. Gearbeitet werden kann mit diesem Beispiel, das in Pseudocode angegeben ist:

```
data = [1 2; 3 4; 5 6];          // create some data
[pc ws sc] = pca( data );        // apply PCA
firstComp = pc( :, 1 );          // get first PC.
newData = firstComp' * data';    // project data onto first PC
newData = newData';              // rearrange to column order.
```

Aufgabe 4 (Linear Discriminant Analysis) (8 Punkte)

Das Ziel der PCA besteht vorrangig darin, eine Dimensionsreduktion bei höchstmöglicher Varianzerhaltung zu erreichen. Dies kann, wie in Aufgabe 2 behandelt, dazu führen, dass die Klassenseparierbarkeit verschlechtert wird. Hier setzt die *Diskriminanzanalyse* an: Gesucht ist eine Projektion in einen Raum, die die Unterscheidungsfähigkeit zwischen den Klassen optimiert. Eine solche Analyse soll in dieser Aufgabe durchgeführt werden. Mit dem Übungsblatt wird die Funktion

```
createPointCloud( numPoints, variance, stretchFactor, alpha )
```

zur Verfügung gestellt. Sie erstellt eine Punktmenge, die grundsätzlich entlang der x-Achse gestreckt ist. Die einzelnen Argumente sollen dabei hier kurz erläutert werden:

- `numPoints` bezeichnet die Anzahl der zu erstellenden Punkte,
- `variance` bezeichnet die Streuung in x-Richtung,
- `stretchFactor` beeinflusst die Streuung in y-Richtung,
- `alpha` gibt den Winkel (in Grad) an, um den die Punktwolke gedreht wird.

Folgende Punkte sind nun zu bearbeiten:

a) Es sollen zwei Punktmengen erstellt werden, wobei eine Separierbarkeit zwischen den beiden repräsentierten Klassen sichergestellt sei. Im Skript `aufgabe4.m` ist dies bereits umgesetzt. Zur *eigenen Orientierung* sind Variationen aber erlaubt.

b) Führt eine LDA durch. (5 Punkte)

c) Projiziert die Punktmengen in den optimalen Raum. (1 Punkt)

d) Plottet die originalen Datensätze und deren Projektionen und beurteilt bzw. begründet die Resultate. (2 Punkte)

Hinweise:

1. Achtet bei der Darstellung der Punktwolken darauf, dass die Achsenskalierung in x- und y-Richtung identisch ist.
2. Für die Lösung von Aufgabenteil b) wird folgendes Vorgehen vorgeschlagen:
   - Man berechne μ_1, μ_2 und μ.
   - Danach kann der *scatter-within* der Klassen 1 und 2 und der gesamte berechnet werden – in der Form `ScatterW = ScatterW1 + ScatterW2`.
   - Nun wird der *scatter-between* der Klassen ermittelt.
   - Berechnet die Projektionsmatrix: `proj = inv( ScatterW ) * ScatterB` und beachtet: Diese Projektion behält die Darstellung im zweidimensionalen Raum bei, alle Punkte befinden sich aber auf einer Linie.
   - Alternativ zum letzten Schritt kann die Projektionsmatrix `proj = inv( ScatterW ) * (mu1 - mu2)` bestimmt werden. Dann ist zu beachten, dass diese Projektion auf den eindimensionalen Raum erfolgt.
3. Bei Matrix- bzw. Vektoroperationen ist es sehr wichtig, sich über die Ausrichtung der Daten im Klaren zu sein. In der Vorlesung wird von Spaltenvektoren ausgegangen, oftmals wird jedoch mit Zeilenvektoren gearbeitet.