1. Beispiel der Übung Medizinische Bildverarbeitung, v1.0

UE 183.630-2015S

20.4.2015

Fragen bitte an:
Markus Holzer, `markus.holzer@meduniwien.ac.at`
Markus Krenn, `markus.krenn@meduniwien.ac.at`

1 Genereller Ablauf

Das erste Beispiel dient einerseits dem Vertrautwerden mit Matlab und andererseits dazu, das Verständis der PCA (Principal Component Analysis) zu vertiefen.

1.1 Abgabemodalitäten

- Eine gemeinsame Abgabe pro Team, inkl Angabe der Teammitglieder
- Abgabe per Mail
- Deadline für die Abgabe ist der 17. Mai 24h

Inhalt der Abgabe:

- Abzugeben ist der Lauffähige Code als `.zip`.
- Es muss ein `RUN.m` File enhalten sein, das ohne Userinteraktion alle Resultate berechnet und die Figures plottet.
- Der Code muss dokumentiert sein (inkl. Angabe, wo welches Unterbeispiel gerechnet wird).
- Weiters ist ein PDF zu attachen, in dem die einzelnen Punkte ausgearbeitet sind (ca. 3-5 Seiten).

- Die Ausarbeitung sollte knapp und präzise sein, bitte keine Code-Teile oder allgemeine Erklärungen inkludieren.
- Die Abgabe kann in Deutsch oder Englisch ausgearbeitet werden.
- Fragen zu Umfang / Inhalt und Feedback zum bereits erstellten Code/Report in den Übungseinheiten oder per Email

2 Angabe

Ziel des Beispiels ist die Implementierung einer PCA (Principal Component Analysis) und die Untersuchung ihrer grundlegenden Eigenschaften. Teile des Codes werden zur Verfügung gestellt (Daten, Plotfunktionen, etc) um die Konzentration aufs Wesentliche zu erleichtern.

2.1 Vorhandene Hilfsfunktionen

`plot2DPCA.m` und `plot3DPCA.m` können verwendet werden, um Datenpunkte, Eigenvektoren, Eigenwerte, Ellipsen/Ellipsoide und rekontruierte Daten zu plotten. `plotDEMO.m` zeigt die Möglichkeiten, `open plot2DPCA.m` zeigt die Dokumentation und Verwendung.

Evtl benötigte Matlab Funktionen sind unter anderem: `clear`, `close all`, `load`, `plot` (dazu als Parameter `,'.'` für Punkte), `axis equal`, `bar`, `imagesc`, `repmat`, `mean`, `var`, `std`, `sort`, `mvnrnd` und `eig`. Informationen zu den Matlab-Funktionen mit `doc Funktionsname`.

2.2 Fragestellung

In Klammer jeweils die erreichbare Punktezahl, insgesamt 30 Punkte.

Für alle Punkte wird angenommen, dass die n Datenpunkte d Dimensionen haben und in einer $d \times n$-Matrix $\mathbf{D}$ vorliegen. D.h. $2 \times n$ für 2D und $3 \times n$ für 3D.

1. **Kovarianzmatrix**

   (a) Schreiben Sie eine Funktion `ourCov.m`, die die Kovarianzmatrix $\mathbf{C}$ für $\mathbf{D}$ berechnet. Die Verwendung der Matlab Funktion `cov` ist zur Berechnung nicht erlaubt, aber kann zum Ergebnis-Vergleich verwendet werden (`cov` erwartet eine $n \times d$ Matrix). **(2 Punkte)**

   (b) Berechnen Sie $\mathbf{C}$ für die Daten in `daten.mat`. Zeigen Sie die Daten mit `plot` in separaten Figures und stellen Sie die Skalierung auf `axis equal`. Interpretieren Sie die unterschiedlichen $\mathbf{C}$ zwischen den Datensets! Welche Informationen stehen an welcher Stelle von $\mathbf{C}$? **(2 Punkte)**

2. **PCA** – Schreiben Sie eine Funktion `pca.m`, die die PCA für **D** berechnet. Die Berechnung soll unabhängig von der Dimension der Daten sein. Ergebnis sind die absteigend sortierten Eigenwerte und die nach absteigenden Eigenwerten sortierten, normierten Eigenvektoren. Sie können für die Berechnung der Eigenvektoren/-werte die Matlab Funktion `eig` verwenden. **(2 Punkte)**

   (a) Plotten Sie mit `plot2DPCA.m` Ihre Ergebnisse für die Daten aus `daten.mat`. **(1 Punkt)**

   (b) Was geben die Eigenvektoren an? Wo sieht man das im Plot? **(1.5 Punkt)**

   (c) Was geben die Eigenwerte an? Wo sieht man das im Plot? In welcher Relation stehen sie zur Gesamtvarianz? **(1.5 Punkte)**

   (d) Welchen Einfluss hat ein fehlender Mittelwertabzug (bei **D**) auf die Berechnung? **(1 Punkt)**

3. **Unterraum-Projektion**

   (a) Berechnen Sie die PCA für `data3`. Projiziieren Sie die Daten in `data3` auf den Hauptvektor (Plot). Welche Dimension haben Ihre Daten jetzt? Rekonstruieren Sie die Projektion und plotten Sie das Ergebnis mittels `plot2DPCA.m`. Beschreiben Sie den Effekt von Projektion und Rekonstruktion auf die Datenpunkte. Wie groß ist der Durchschnittliche Fehler zwischen Rekonstruktion und Originaldaten? **(3 Punkte)**

   (b) Machen Sie die selbe Untersuchung, nur mit dem Nebenvektor. Welche Eigenvektoren werden Sie verwenden, um eine Datenmatrix mit möglichst wenig Fehler mit möglichst wenig Eigenvektoren (in diesem Fall 1) darzustellen? **(1 Punkt)**

4. **Untersuchungen in 3D**

   (a) Berechnen Sie die PCA und plotten Sie Daten und Eigenvektoren für die Daten in `daten3d.mat`. Beschreiben Sie die Relation von Kovarianzmatrix (Varianzen), Eigenwerten und -vektoren und den Ellipsoiden der Standardabweichungen. **(2 Punkt)**

   (b) Projiziieren Sie auf den Unterraum, der durch die ersten beiden Eigenvektoren aufgespannt wird. Welche Dimension haben Ihre Daten? Rekonstruieren Sie die Punkte im Originalraum und plotten Sie das Ergebnis. Welche Information ist verloren gegangen? **(1 Punkt)**

5. **Shape Modell**

(a) Berechnen Sie die PCA der Shape Daten in `shape.mat` – die Matrix `aligned` hat die Dimensionen `nPunkte x nDimensionen x nShapes`. Schreiben Sie eine Funktion `generateShape`, die zu einem Parametervektor `b` mit einer Länge entsprechend der Zahl der Eigenvektoren neue Shapes generieren kann.**(4 Punkt)**

(b) Schreiben Sie eine Funktion `plotShape`, die die Shapes in blau darstellt und plotten und interpretieren Sie die Einzelnen Modes (d.h. `b` ist 0 bis auf einen Wert) im Bereich von $\pm 3\lambda$, wobei λ die Standardabweichung des entprechenden Modes bezeichnet. Die Funktion soll gleichzeitig auch das mean shape (d.h. b gleich dem Nullvektor) in rot darstellen. Beschreiben und interpretieren Sie. **(4 Punkt)**

(c) Setzen Sie nun `b=randn(1,nEigenvectors).*stddeviations`. Beschränken Sie nun wie in den 2D und 3D Beispielen die Zahl der Eigenvektoren, dementsprechend die Länge von `b`, plotten Sie die resultierenden Shapes und interpretieren Sie. Beschränken Sie so, dass das Shape Modell 100%, 95%, 90% und 80% der Gesamtvarianz beinhaltet.**(4 Punkt)**