

Übungsblatt 5: Nichtlineare Gleichungssysteme und Matrizeneigenwertprobleme (20 Punkte)

22. November 2017

Abgabe bis **30.11.2017** 23:55 Uhr

Hinweis: Benutzen Sie numpy wie in den einzelnen Aufgaben angegeben.

Aufgabe 1: Hauptkomponentenanalyse (PCA) (20 Punkte)

Bob, ein Forscher in der Area 51, wird eines Tages damit beauftragt eine mysteriöse schwarze Box (msB) zu untersuchen. Er kann weder deren Ursprung einschätzen, noch ihre Funktionsweise erahnen. Gepackt vom wissenschaftlichen Ehrgeiz, macht er sich an die Arbeit.

Da es sechs Messeingänge in der msB gibt, beschließt Bob alle zu nutzen um Zeitreihen der Signale zu messen. Als er sich diese im Nachhinein anschaut, stellt er mit Verwunderung fest, dass alle Signale identisch sind. Er vermutet aber auch, dass diese jeweils für unterschiedliche Koordinaten stehen. In seiner Ratlosigkeit wendet er sich an seine Kollegin Alice. Diese teilt seine Verwunderung und rät ihm eine Hauptkomponentenanalyse, auch PCA genannt, durchzuführen.

Voller neuem Tatendrang wendet sich Bob an seine studentische Hilfskraft (Sie) und erklärt: „Mittels einer PCA können umfangreiche Datensätze strukturiert, vereinfacht und veranschaulicht werden. Dazu wird der Raum, welcher vom Datensatz aufgespannt wird, transformiert bis einzelne Dimensionen möglichst wenig korrelieren. Anschließend kann bei Bedarf die Dimensionalität des Datensatzes reduziert werden. Eine Dimension wird in der Regel dann entfernt, wenn diese sehr wenig Informationen beinhaltet, ihre Varianz also niedrig ist.“

Während er sich schon seinem nächsten Projekt zuwendet, teilt er Ihnen mit, dass er bis Donnerstag eine effiziente Implementierung einer PCA haben möchte. Des Weiteren sollen Sie ihm bis zu besagtem Tag die Ergebnisse der PCA liefern. Zu Ihrem Glück finden Sie noch eine Anleitung:

- (a) Sichtung des Datensatzes (1 Punkt)
- (i) Laden Sie den Datensatz D von `data.txt` mit `np.loadtxt` ein.
 - (ii) Verwenden Sie `shape` um sich die Dimensionalität des Datensatzes auszugeben. Die Dimensionen sollten $N \times M$ entsprechen, wobei N die Anzahl der Datenpunkte und M die Anzahl der Dimensionen darstellt. Geben sie die Anzahl der Datenpunkte und Dimensionen in Ihrem Datensatz an.
 - (iii) Plotten Sie die Zeitreihe für jede Dimension in einem jeweils separaten Plot.
- (b) Vorbereitung des Datensatzes (1 Punkt)
- (i) Mean Free: Berechnen Sie jeweils den Mittelwert einer Dimension. Ziehen Sie diesen anschließend von der gehörigen Dimension ab.
 - (ii) Transponieren Sie den Datensatz D . Bedenken Sie, dass sie eine erneute Variablenzuweisung benötigen damit D transponiert bleibt.
- (c) Berechnung Kovarianz Matrix (1 Punkt)
- (i) Berechnen Sie die Kovarianz Matrix K Ihres Datensatzes. (Hinweis: Es sollte eine 2×2 Matrix sein)
- (d) Eigenwerte und Eigenvektoren mit der Potenz Methode (4 Punkte)
- (i) Implementieren Sie die Potenz Methode zur Berechnung von Eigenwerten und Eigenvektoren. Lassen Sie des Weiteren die Anzahl der Iterationsschritte ausgeben.
 - (ii) Berechnen Sie die Eigenwerte Λ und Eigenvektoren V von K mit der von Ihnen implementierten Potenz Methode.
 - (iii) Stellen Sie sicher, dass alle Eigenwerte von Λ positiv sind. Ist dies nicht der Fall, nehmen Sie den Betrag der Eigenwerte.
 - (iv) Sortieren Sie alle Eigenwerte und die damit verbundenen Eigenvektoren in Absteigender Reihenfolge, sofern notwendig.
 - (v) Transponieren Sie V_{sortiert} .
 - (vi) Multiplizieren Sie die V_{sortiert}^T mit Ihrem Datensatz um $D_{PCA_{\text{power}}}$ zu erhalten.
- (e) Eigenwerte und Eigenvektoren mit der QR-Zerlegung (4 Punkte)
- (i) Implementieren Sie die QR-Zerlegung zur Berechnung von Eigenwerten und Eigenvektoren. Lassen Sie des Weiteren die Anzahl der Iterationsschritte ausgeben.
 - (ii) Berechnen Sie die Eigenwerte Λ und Eigenvektoren V von K mit der von Ihnen implementierten QR-Zerlegung.
 - (iii) Stellen Sie sicher, dass alle Eigenwerte von Λ positiv sind. Ist dies nicht der Fall nehmen Sie den Betrag der Eigenwerte.
 - (iv) Sortieren Sie alle Eigenwerte und die damit verbundenen Eigenvektoren in Absteigender Reihenfolge, sofern notwendig.
 - (v) Transponieren Sie V_{sortiert} .
 - (vi) Multiplizieren Sie die V_{sortiert}^T mit Ihrem Datensatz um $D_{PCA_{QR}}$ zu erhalten.

(f) Reduktion der Dimensionalität (3 Punkte)

- (i) Plotten Sie die Zeitreihe für jede Dimension von $D_{PCA_{power}}$ und $D_{PCA_{QR}}$. Vergleichen Sie jeweils diese mit den Plots von Aufgabe aiii. Was fällt Ihnen auf? (Hinweis: Plotten Sie alle restlichen Dimensionen eines D_{PCA} jeweils in einen Plot)
- (ii) Berechnen Sie die Varianz σ^2 jeder Dimension von $D_{PCA_{power}}$ und $D_{PCA_{QR}}$. Normieren Sie jeweils σ^2 . Was fällt Ihnen auf?
- (iii) Reduzieren Sie die Dimensionalität von $D_{PCA_{QR}}$ indem Sie alle Dimensionen mit $\sigma^2 < 10^{-6}$ verwerfen ($D_{PCA_{QR-red}}$). (Hinweis: Gemeint ist eine Matrix, welche nur Dimensionen enthält, die dem Kriterium entsprechen) Wie viele Dimensionen bleiben übrig? Wie verhält es sich mit $D_{PCA_{power}}$?
- (iv) Aus Ihren Beobachtungen von Aufgabe fi, fii und fiii: Erklären Sie ihre Beobachtungen und umreißen Sie das Prinzip hinter der PCA.

(g) Algorithmen zur Berechnung von Eigenwerten und -vektoren (3 Punkte)

- (i) Vergleichen Sie beide Algorithmen angesichts ihrer Laufzeit und Eigenwerte und Eigenvektoren? Fallen Ihnen Besonderheiten auf?
- (ii) Erklären Sie Ihre Beobachtungen, indem Sie die Funktionsweise beider Algorithmen zur Hilfe nehmen.

(h) Bericht an Bob (3 Punkte)

- (i) Basierend auf Ihren Beobachtungen in Aufgabe fi: Welches Modell beschreibt den Prozess im Inneren der msB am Besten? Erklären Sie Ihre Antwort.
- (ii) Basierend auf Ihren Beobachtungen: Welcher Algorithmus zur Bestimmung der Eigenwerte und Eigenvektoren ist effizienter?
- (iii) Speichern Sie die $D_{PCA_{red}}$ des effizientesten Algorithmus in einer *txt* Datei mit *np.savetxt*.

Hinweise:

- Matrizen transponieren ist mit numpy erlaubt.
- Zur Multiplikation von Matrizen dürfen Sie numpy verwenden.
- Der Betrag kann mit Hilfe von numpy genommen werden.
- Selbstverständlich ist das Nutzen von numpy Array Besonderheiten erlaubt (Indexing, Slicing, etc.).