

Übungsblatt 2: Unüberwachtes Lernen

Abgabeschluss: Montag, der 19.05.2008 um 10:00 Uhr.

Für dieses Aufgabenblatt sind sowohl Code als auch eine schriftliche Ausarbeitung abzugeben. Der Code und eine elektronische Version der Ausarbeitung (als PDF) muss über PASS abgegeben werden (siehe link auf der Website). Zusätzlich ist ein Ausdruck der Ausarbeitung abzugeben, entweder in der Veranstaltung oder über unser Sekretariat.

Aufgaben

Teil 1: Implementation

Aufgabe 1 (5 Punkte)

Schreibe eine Funktion `PCA` mit der Signatur

$$[Z, U, D] = \text{PCA}(X, m)$$

die eine $d \times n$ matrix X und die Anzahl der zu verwendenden Komponenten m erhält, und daraus die Hauptkomponenten und die entsprechend projizierten Datenpunkt in der $m \times n$ matrix Z berechnet.

U und D sollen die Hauptkomponenten enthalten: U ist eine $d \times d$ -Matrix, die die Hauptkomponentenrichtungen enthält, und D ein $1 \times d$ -Vektor, der die Hauptkomponentenwerte enthält, beides absteigend sortiert (d.h. $D_1 \geq D_2 \dots$).

Aufgabe 2 (10 Punkte)

Schreibe eine Funktion `Isomap` mit der Signatur

$$Y = \text{Isomap}(X, m, n_rule, param)$$

welche für gegebene d -dimensionale Daten $X \in \mathbb{R}^{d \times n}$ eine m -dimensionale Einbettung $Y \in \mathbb{R}^{m \times n}$ mit dem Isomap Algorithmus berechnet. Der Parameter `n_rule` bestimmt die Methode ('knn' oder 'eps-ball'), mit der der Graph auf den Daten konstruiert wird, `param` ist der dazugehörige Parameter (k bzw. ϵ). Die Funktion soll robust gegen fehlerhafte Parameter sein, verwende die Matlab-Funktion `error`, um den Benutzer über Fehler zu informieren. Du solltest insbesondere testen, ob der Graph zusammenhängend ist.

Aufgabe 3 (10 Punkte)

Schreibe eine Funktion `LLE` mit der Signatur

$$Y = \text{LLE}(X, m, n_rule, param)$$

welche den LLE-Algorithmus implementiert wobei die Parameter dieselbe Bedeutung wie in der Funktion `Isomap` haben.

Teil 2: Anwendung

Aufgabe 4 (8 Punkte)

Wende PCA auf die einzelnen Ziffern des `usps`-Datensatzes von Blatt 1 an. Schreiben Sie hierfür ein Skript (bzw. eine Funktion, die keine Argumente hat, um die Einschränkung in Matlab zu umgehen, daß in Skripten keine lokalen Funktionen definiert werden können), welches wie folgt vorgeht:

1. Lade den `usps`-Datensatz.

2. Für jede Ziffer:
 - (a) Extrahiere die \mathbf{x} -Daten für die entsprechende Ziffer.
 - (b) Berechne die PCA für diese Daten.
 - (c) Plote die Hauptkomponentenwerte und die ersten 5 Hauptkomponentenrichtungen als Bilder.
3. Verrausche nun die Bilder durch Addieren von Gaußschem Rauschen (Matlab-Funktion `randn`). Bestimme die Varianz selbst geeignet, so daß die Bilder relativ stark verrauscht sind.
4. Für jede Ziffer:
 - (a) Extrahiere die \mathbf{x} -Daten für die entsprechende Ziffer.
 - (b) Berechne die PCA für diese Daten, wobei m durch Ausprobieren so gewählt sein soll, daß die entrauschte Rekonstruktion möglichst gut ist.
 - (c) Plote für fünf Beispiele jeweils das verrauschte Bild und die entrauschte Rekonstruktion.

Hinweis: verwende `subplot`, um die Bilder in einem einzigen Ausgabefenster anzuordnen.

Aufgabe 5 (8 Punkte)

Wende LLE und Isomap auf die Datensätze an, die auf der Website angegeben sind. Schreibe wieder ein Skript, das wie folgt vorgeht:

1. Für alle Datensätze:
 - (a) Lade den entsprechenden Datensatz.
 - (b) Wende Isomap auf den Datensatz an, wobei die Parameter geeignet zu wählen sind (in Abhängigkeit vom Datensatz)
 - (c) Plote die Einbettung in einem zweidimensionalen Koordinatensystem. Zeige den Namen des Datensatzes, die Methode und den verwendeten Parameter im Title des Plots an (mit `title`).
 - (d) Wiederhole dasselbe für LLE.

Aufgabe 6 (9 Punkte)

In dieser Aufgabe soll der Einfluss von Rauschen auf LLE und Isomap anhand des zweidimensionalen `flatroll`-Datensatzes untersucht werden. Schreibe ein Skript, das wie folgt vorgeht:

1. Lade den Datensatz.
2. Addiere Rauschen mit den Varianzen 0,2 und 1,8 zu dem Datensatz und speichere das Ergebnis jeweils in eigenen Variablen.
3. Wende LLE und Isomap auf die Datensätze an, jeweils für die Varianten `knn` und `eps-ball`. Finde durch Ausprobieren jeweils einen eindeutig zu kleinen, einen möglichst guten und einen eindeutig zu großen Parameterwert.
4. Erzeuge für LLE und Isomap in jeder der beiden Varianten einen Plot, der das Ergebnis und den Graphen der kürzesten Wege für jeden der drei Parameterwerte plottet. (Also insgesamt vier Plots mit je sechs Unterplots).