Praktikum Maschinelles Lernen und Datenanalyse

Sommersemester 2007

Fachgebiet Maschinelles Lernen
Institut für Softwaretechnik und
theoretische Informatik
Fakultät IV, Technische Universität Berlin
Prof. Dr. Klaus-Robert Müller
Email: krm@cs.tu-berlin.de

Übungsblatt 2: Unüberwachtes Lernen

Abgabeschluss: Montag, der 21.05.2007 um 10:00 Uhr.

Für dieses Aufgabenblatt sind sowohl Code als auch eine schriftliche Ausarbeitung abzugeben. Sendet Euren Code an `mikio@cs.tu-berlin.de` **und** `buenau@cs.tu-berlin.de` **mit Subject** "`ML-Praktikum Abgabe` *`Name`*". Gebt Eure Ausarbeitungen in unserem Sekretariat (FR 6-9) bei Frau Gerdes (Raum FR 6052) ab.

Bitte beachtet weiter die Coding-Richtlinien von Blatt 1

Aufgaben

Aufgabe 1: Hauptkomponentenanalyse (PCA) (10 Punkte)

1. Schreibe eine Funktion `PCA` mit der Signatur

   ```
   [Z, U, D] = 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. $D_1 \geq D_2 \ldots$).

2. Wir untersuchen das Verhalten von PCA auf einem Datensatz, der per Konstruktion einen "interessanten" m-dimensionalen Unterraum enthält.

   - Erzeuge einen 20-dimensionalen Datensatz mit 100 Datenpunkten, wobei

     $$X_i = Az_i + \sqrt{s}e_i.$$

     Hierbei ist A eine zufällige 20×3-Matrix, die am Anfang einmal gewählt wird, und z_i und e_i sind normalverteilte Vektoren (siehe die Funktion `randn`). Der Parameter s ist die Standardabweichung des Rauschens. (Verwende keine `for`-Schleife!)
   - Plotte die Hauptkomponentenwerte für verschiedene Rauschanteile s. Was fällt auf?
   - Berechne den Fehler zwischen der Rekonstruktion der X_i aus den ersten m Hauptkomponenten und dem "wahren" Punkt Az_i für verschiedene Rauschanteile s. Funktioniert die PCA immer zufriedenstellend?

3. Wende PCA auf die einzelnen Ziffern des `usps`-Datensatzes von Blatt 1 an.

   - Für den Originaldatensatz, plotte die Hauptkomonentenwerte, und die ersten 10 Hauptkomponentenrichtungen.
   - Verrausche die Bilder jetzt leicht, und versuche durch optimale Wahl von m die Bilder zu entrauschen. Plotte das jeweils beste Ergebnis pro Ziffer und das jeweils gewählte m.

Aufgabe 2: Isomap (20 Punkte)

1. Schreibe eine Funktion `Isomap` mit der Signatur

   ```
   Y = Isomap(X, m, n_rule, param)
   ```

   welche für gegebene d-dimensionale Daten $X \in \mathbb{R}^{d \times n}$ eine m-dimensionale Einbettung $Y \in R^{m \times n}$ 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.

2. Wende den Algorithmus auf die Datensätze an, die auf der Website angegeben sind. Finde dabei jeweils geeignete Paramter k oder ϵ und plotte die Einbettung in einem zweidimensionalen Koordinatensystem und nenne die Werte der Parameter. Vergleiche Dein Ergebnis mit der 'richtigen' Einbettung (falls gegeben), indem Du die Datenpunkte entsprechend einfärbst.

3. Was passiert, wenn k oder ϵ zu klein bzw. zu groß gewählt werden? Belege Deine Antwort durch geeignete Plots. Zur Untersuchung kann es hilfreich sein, spezielle eigene Datensätze (*toy datasets*) zu erzeugen.

4. In dieser Aufgabe soll der Einfluss von Rauschen anhand des zweidimensionalen `flatroll`-Datensatzes untersucht werden. Addiere dazu Normalverteiltes Rauschen mit Varianz in $[0.2, 1.8]$ auf beide Dimensionen. Was passiert bei welchen noise-level und warum? Belege Deine Antwort durch geeignete Plots. Dazu kann es hilfreich sein, den Graphen zu plotten, der dem verwendeten Distanzmaß (shortest paths) zugrundeliegt.

Aufgabe 3: LLE (20 Punkte)

1. Schreibe eine Funktion `Isomap` mit der Signatur

   ```
   Y = LLE(X, m, n_rule, param)
   ```

   welche für gegebene d-dimensionale Daten $X \in \mathbb{R}^{d \times n}$ eine m-dimensionale Einbettung $Y \in R^{m \times n}$ mit dem LLE 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.

2. Wende den Algorithmus auf die Datensaetze an, die auf der Website angegeben sind. Finde dabei jeweils geeignete Paramter k oder ϵ und plotte die Einbettung in einem zweidimensionalen Koordinatensystem und nenne die Werte der Parameter. Vergleiche Dein Ergebnis mit der 'richtigen' Einbettung (falls gegeben), indem Du die Datenpunkte entsprechend einfärbst.

3. Was passiert, wenn k oder ϵ zu klein bzw. zu groß gewählt werden? Belege Deine Antwort durch geeignete Plots. Zur Untersuchung kann es hilfreich sein, spezielle eigene Datensätze (*toy datasets*) zu erzeugen.

4. In dieser Aufgabe soll der Einfluss von Rauschen anhand des zweidimensionalen `flatroll`-Datensatzes untersucht werden. Addiere dazu Normalverteiltes Rauschen mit Varianz in $[0.2, 1.8]$ auf beide Dimensionen. Was passiert und warum? Belege Deine Antwort durch geeignete Plots. Dazu kann es hilfreich sein, den Nachbarschaftsgraphen zu plotten.

Zusatzaufgabe (20 Punkte)

1. Entwickle eine Heuristik zur Wahl von m bei PCA. Hierbei versucht man, einen Knick in der Folge der Hauptkomponentenwerte zu finden. Versuche eine möglichst robuste Methode hierzu zu entwickeln, und teste ihr Verhalten auf einigen verrauschten Datensätzen.

2. Entwickle eine Heuristik zur Wahl von k oder ϵ für Isomap oder LLE (wahlweise). Gib den verwendeten Code ab und begründe anhand von Datensätzen und Plots, warum die Methode funktioniert.

3. Entwickle eine Methode, um Isomap oder LLE (wahlweise) robuster gegenüber Rauschen zu machen. Zur Dokumentation Deiner Ergebnisse gib den verwendeten Code und eine Beschreibung der Methode sowie der Ergebnisse (mit aussagekräftigen Plots) ab.