

Übungsblatt 3: Clustering

Abgabeschluss: Montag, der 8.06.2009 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).

Aufgaben

Teil 1: Implementation

Aufgabe 1 (2.5 Punkte)

Implementiere den K-means Clustering Algorithmus als Funktion

```
[ mu, r ] = kmeans(X, k, max_iter, prog_fun)
```

welche fuer die Spalten der $d \times n$ Matrix X die $d \times k$ Matrix der k Cluster Zentren μ sowie den n -dimensionalen Vektor r der Zugehoerigkeiten berechnet. Dabei enthaelt der i -te Eintrag von r den Index des Clusters zu dem der i -te Datenpunkt gehoert. Der Algorithmus soll terminieren, wenn sich die Zugehoerigkeiten nicht mehr aendern, spaetestens jedoch nach `max_iter` Schritten (optionaler Parameter mit Standardwert 100). Die Funktion soll nach jeder Iteration folgende Informationen ausgeben.

- Die Nummer der Iteration.
- Die Anzahl der Cluster-Zugehoerigkeiten, die sich in diesem Schritt geaendert haben.
- Den Wert der Fehlerfunktion (siehe Skript).

Der optionale Parameter `prog_fun` ist ein handle auf eine Funktion (siehe Matlab Funktion `feval`), welche nach jedem Schritt aufgerufen wird, um den ueber den Fortschritt des Algorithmus zu informieren. Die Signatur ist

```
prog_fun(X, mu, r)
```

wobei μ die aktuellen Clusterzentren, r die aktuellen Zuweisungen und X die Daten sind.

Aufgabe 2 (0.5 Punkte)

Schreibe eine Visualisierungsfunktion fuer K-means Clustering (siehe Argument `prog_fun`) mit dem Namen `plot_kmeans_USPS` welche fuer eine beliebige Anzahl Cluster die aktuellen Zentren in einer figure als 16×16 Bilder (in graustufen) darstellt und auf Tastendruck wartet. Die Zentren sollen mit den jeweiligen Cluster-Indices beschriftet sein.

Aufgabe 3 (2 Punkte)

Implementiere stepwise optimal hierarchical agglomerative clustering mit K-means Kriterium als Funktion

```
[ R, kmloss, mergeidx ] = kmeans_agglo(X, r)
```

welche fuer die Spalten der $d \times n$ Matrix X mit initialer Clusterloesung beschrieben durch den $1 \times n$ Zugehoerigkeitsvektor r eine hierarchische Clusterloesung berechnet. Das Ergebnis soll in folgendem Format zurueckgegeben werden.

- R ist eine $(k - 1) \times n$ Matrix welche fuer jeden Schritt die Zugehoerigkeiten enthaelt, jede Zeile ist also eine Clusterloesung.
- `kmloss` ist ein $k \times 1$ Vektor, welcher den Wert der Kostenfunktion in jedem Schritt enthaelt.
- `mergeidx` ist eine $(k - 1) \times 2$ Matrix, welche in jeder Zeile die Indices der vereinigten Cluster enthaelt.

Aufgabe 4 (1 Punkt)

Implementiere eine Funktion, welche zu einer gegebenen hierarchischen Clusterloesung einen Dendrogram-Plot erstellt:

`agglo_dendro(kmloss, mergeidx)`

Die Parameter `kmloss` und `mergeidx` entsprechen den Ergebnissen von `kmeans_agglo`. Im Skript gibt es ein Beispiel fuer einen Dendrogram Plot.

Aufgabe 5 (3 Punkte)

Implementiere den EM-Algorithmus für Gaussche Mixturmodelle als Funktion

`[pi, mu, sigma] = em_mog(X, k, max_iter, init_kmeans, prog_fun)`

wobei die Parameter folgende Bedeutung haben.

Ausgabe	<code>pi</code> <code>mu</code> <code>sigma</code>	$1 \times k$ -Matrix der $\hat{\pi}_k$ $d \times k$ -Matrix der $\hat{\mu}_k$ (Mittelpunkte) Cell-array der Länge k der $d \times d$ Kovarianzmatrizen $\hat{\Sigma}_k$
Eingabe	<code>X</code> <code>k</code> <code>max_iter</code> <code>init_kmeans</code> <code>prog_fun</code>	$d \times n$ -Matrix der Datenpunkte Anzahl der Gaussverteilungen Optional: Maximale Anzahl der Iterationen (default: 100) Optional: Initialisierung durch K-Means Clusterloesung (default: 0) Optional: Name oder handle der Visualisierungsfunktion (default: [])

Die Visualisierungsfunktion `prog_fun` wird nach jedem Schritt aufgerufen, um über den Fortschritt zu informieren; die Signatur lautet

`prog_fun(X, mu, sigma)`

wobei `X` die Daten und `mu` und `sigma` die aktuellen Parameter des geschätzten Mixturemodels sind. Wenn `init_kmeans` den Wert 1 hat, dann werden Mittelpunkte, Kovarianzmatrizen und Mixturekoeffizienten mit dem Ergebnis einer K-Means Clusterloesung initialisiert.

Die Funktion soll nach jedem Schritt die Nummer der Iteration und die log likelihood pro Datenpunkt ausgeben. Der Algorithmus soll terminieren, wenn die maximale Anzahl an Iteration `max_iter` erreicht wurde oder sich die log likelihood nicht mehr ändert, also ein lokales Maximum erreicht wurde.

Aufgabe 6 (1 Punkt)

Schreibe für den zweidimensionalen Fall eine Visualisierungsfunktion fuer den EM-Algorithmus (siehe Argument `prog_fun`) mit dem Namen `plot_em2d` welche die Daten plotet sowie die Kovarianzmatrizen `sigma` durch Ellipsen darstellt und auf Tastendruck wartet. Tipp: Die Eigenzerlegung von `Sigma` liefert die Hauptachsen, und die Wurzeln aus den Eigenwerten die Radien.

Teil 2: Anwendung

Verdeutliche Deine Antworten zu den folgenden Fragen durch geeignete plots.

Aufgabe 7 (4 Punkte)

Analysiere den `5gaussians` Datensatz mit allen Methoden für $k = 2, \dots, 10$ Cluster:

1. Sind alle Methoden in der Lage, die 5 Cluster zuverlässig zu finden?
2. Welchen Einfluss hat die Initialisierung des EM-Algorithmus mit einer K-Means Loesung auf die Anzahl benoetigter Iteration und die Qualität der Lösung?
3. Wie sieht das Dendrogramm der hierarchischen Clusterlösung aus und kann man anhand des Dendrogramms eine Schätzung fuer eine geeignetes k abgeben?

Aufgabe 8 (3 Punkte)

Analysiere den `2gaussians` Datensatz mit k-means und dem EM Algorithmus:

1. Welcher Algorithmus funktioniert besser, und wieso?
2. Wie hängt die Lösung des EM-Algorithmus von der Initialisierung ab?

Aufgabe 9 (3 Punkte)

Wende EM und K-Means Clustering auf den USPS Datensatz mit $k = 10$ an.

1. Welcher Algorithmus liefert bessere Ergebnisse?
2. Erstelle ein Dendrogramm zu einer hierarchischen Clusterloesung und zusätzlich einen Plot, der zu jedem agglomerativen Schritt die Mittelpunkte der Clusterzentren als 16×16 Bilder zeigt.