Machine learning lab course (Praktikum)

Summer term 2013

Fachgebiet Maschinelles Lernen
Fakultt IV, Technische Universitt Berlin
Prof. Dr. Klaus-Robert Mller
Email: klaus-robert.mueller@tu-berlin.de

Problem set 2: Clustering, EM

Part 1: Implementation

Function stubs for these assignments have been provided in `ps2_implementation.py`.

Assignment 1 (10 point)

Implement K-means Clustering as a function

`mu, r = kmeans(X, k, max_iter=100)`

which, with respect to the columns of the $d \times n$ Matrix `X`, calculates the $d \times k$ Matrix for the `k` Cluster centroids `mu` as well as the n-dimensional vector `r` of cluster membership: the i-th entry of `r` should contain the index of the Clusters to which the i-th datapoint belongs.

The algorithm should terminate when the membership no longer changes or after `max_iter` (optional parameter with default value 100) no. of steps, depending on which comes first. The function should print the following information after each iteration:

- The number of iterations performed so far.
- The number of cluster memberships which changed in the preceding step.
- The loss function value (see guide).

Assignment 2 (10 point)

Implement stepwise optimal hierarchichal agglomerative clustering with the K-means criterion as a function.

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

which given the columns of the $d \times n$ Matrix `X` and the initial clustering solution given by the $1 \times n$ membership vector `r` calculate a hierarchical clustering solution. The result should be returned in the following format:

- `R` is a $(k-1) \times n$ matrix which contains the cluster membership *before* each agglomeration step. That is, the l-th row of `R` contains the (one-based) cluster indices for each data point where the total number of clusters is $k-l+1$. The first row of `R` is the initial clustering `r`, and the last row is a clustering with two clusters.
- `kmloss` is a $k \times 1$ vector, which contains the loss function value after each agglomeration step, where the first entry is the loss of the initial clustering `r`. The loss function is the sum of the Euclidean distances from each data point to its cluster centre.
- `mergeidx` is a $(k-1) \times 2$ matrix, which contains the indices of the two clusters that were merged at each step. That is, the l-the row of `mergeidx` contains the two indices that were unified in the l-th step. The index of the new (joint) cluster is the cluster index in the second column.

You should implement this yourself, do not use functions like `scipy.cluster.hierarchy.linkage`.

Assignment 3 (10 point)

Implement a function which given a hierarchical clustering sets up a dendrogram plot:

`agglo_dendro(kmloss, mergeidx)`

The parameters `kmloss` and `mergeidx` correspond to the the results of `kmeans_agglo`. See the handbook for an example dendrogram plot.
You may use the function `scipy.cluster.hierarchy.dendrogram`.

Assignment 4 (20 points)

Implement the EM algorithm for gaussian mixture models as a function:

```
pi, mu, sigma = em_mog(X, k, max_iter=100, init_kmeans=False)
```

where the parameters have the following definitions:

Output	`pi`	$1 \times k$-Matrix of $\hat{\pi}_k$
	`mu`	$d \times k$-Matrix of $\hat{\mu}_k$ (Center Points)
	`sigma`	Cell-array of length k of the $d \times d$ covariance matrices $\hat{\Sigma}_k$
Input	`X`	$d \times n$-Matrix of datapoints
	`k`	number of normally distributed components
	`max_iter`	Optional: maximal number of Iterations (default: 100)
	`init_kmeans`	Optional: Initialisation by means of K-Means Cluster solution (default: `False`)

After every step the function should print the number of the iteration and the log likelihood per datapoint. The algorithm should terminate when the maximal number of iterations `max_iter` has been reached or the log likelihood does not change; i.e. when a local maximum has been reached.

Assignment 5 (10 point)

Write a function that visualizes the EM solution for two-dimensional data:

```
plot_em_solution(X, mu, sigma)
```

The figure should show:

- the data as a scatter plot;
- the mean vectors as red crosses; and
- the covariance matrices as ellipses (centered at the mean).

Part 2: Application

Please clarify your answers to the following questions with suitable plots.

Assignment 6 (20 points)

Analyse the `5gaussians` dataset with both methods for $k = 2, \ldots, 10$ cluster.

1. Do both methods find the 5 clusters reliably?
2. What role does the initialisation of the EM algorithm with a K-means solution play in the number of necessary iterations and the quality of the solution?
3. What does the dendrogramm of the hierarchical clustering look like and is it possible to pick a suitable value of k from the dendrogramm?

Assignment 7 (10 point)

Analyse the `2gaussians` dataset with k-means and the EM-algorithm.

1. Which algorithm works better and why?
2. How does the solution of the EM-algorithm depend on the intialisation?

Assignment 8 (10 point)

Use Em and K-means clustering on the USPS dataset with $k = 10$.

1. Which algorithm delivers better results?
2. Set up a Dendrogramm to the hierarchical clustering solution and also a plot which displays the cluster centroids as a 16×16 image at every agglomerative step.