Machine Learning und Data Analysis Course

Sommersemester 2011

Maschine Learning Department
Software engineering und theoretical computer science
Fakulty IV, Technical University Berlin
Prof. Dr. Klaus-Robert Müller
Email: klaus-robert.mueller@tu-berlin.de

# Exercise Sheet 3: Clustering

**Deadline:** See course calendar.

For this problem set please hand in code as well as written solutions. The code and an electronic version of the written solutions should be submitted to PASS (see the link on the website).

## Aufgaben

### Teil 1: Implementation

#### Exercise 1 (2.5 points)

Implement K-means Clustering as a function

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

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 read out 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 script).

The optional paramater `prog_fun` should be a handle to a function (see the matlab function `feval`) which is called after every step, to inform the user with regard to the progress of the algorithm.

The signature is:-

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

where `mu` are the actual cluster centroids, `r` cluster memberships and `X` are the data.

#### Exercise 2 (0.5 points)

Write a visualisation function for K-means clustering (see the argument `prog_fun`) with the name `plot_kmeans_USPS` which displays the actual centroids in a figure as a $16x16$ figure (greyscale) and waits for keyboard input. The centroids should be marked with the individual cluster indices.

#### Exercise 3 (2 Points)

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

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

which given the columnds 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 memberships at every step - every line is thus a clustering solution.
- `kmloss` is a $k \times 1$ vector, which contains the loss function values at every step.
- `mergeidx` is a $(k-1) \times 2$ matrix, which contains the indices of the clusters together.

**Exercise 4 (1 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`. In the script there is an example for a dendrogram plot.

**Exercise 5 (3 Points)**

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

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

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: 0) |
| | `prog_fun` | Optional: Name or handle of the visualisations function (default: `[]`) |

The visualisation function `prog_fun` should be called after every step to inform the user as to the progress of the algorithm; the signature should be:

`prog_fun(X, mu, sigma)`

where `X` are the data and `mu sigma` the actual parameters of the estimated mixture models. If `init_kmeans` has the value 1, then the centerpoints, covariances and mixture coefficients should be initialised with the result of a K-means clustering.

After every step the function should return 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 doesn't change - that is, a local maximum has been reached.

**Exercise 6 (1 Point)**

Write a visualisation function for the 2 dimensional version of the EM -algorithm (see argument `prog_fun` ) with the name `plot_em2d` which plots the data as well as the the covariances `sigma` as ellipses and waits for keyboard input to continue. Tip: the eigenvectors of `Sigma` gives the primary axes and the squareroots of the eigenvalues returns the radii.

## Teil 2: Application

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

**Exercise 7 (4 Points)**

Analyse the `5gaussians` dataset with all methods für $k = 2, \dots, 10$ Cluster.

1. Do all 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?

**Exercise 8 (3 Points)**

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?

**Exercise 9 (3 Points)**

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 $16 \times 16$ image at every agglomarative step.