April 11, 2014

1. Prior in `mclust`?

See Fraley and Raftery (2007). They talked about imposing conjugate priors to the parameters $\boldsymbol{\mu}_k$, Σ_k and τ_k. And in EM algorithm, instead of using MLE's, they use the posterior modes in the M-step.

Note 1: They only gave prior distributions for mean and variance, but not for the proportion. I'm assuming that the proportion is estimated the same way as in Fraley and Raftery (2002).

Note 2: The reason behind using a Bayesian approach is that EM often fails to converge when no constraints are added to the covariance matrices and/or the number of components is large.

Note 3: In the paper, they mentioned two ways of initializing the EM algorithm: either by obtaining an initial classification through hierarchical clustering (so EM starts with M-step), or by obtaining initial parameters from a simpler model (for example, spherical contour of cov matrices) (in this case, EM starts with E-step).

2. Celeux and Govaert (1995) paper

In this paper, the authors talked about spectral decomposition of covariance matrix $\Sigma_k = \lambda_k D_k A_k D_k^T$. Variations on the parameters λ_k, A_k and D_k lead to 14 different models. For each model, the authors described in detail how to obtain MLE for the covariance matrix Σ_k. Some have closed-form solutions, while others have to be computed using iterative procedure.

Observation: Estimation of Σ_k depends heavily on the constraints on parameters.

3. Cholesky decomposition?

The Cholesky decomposition that appears in `mclust` code is not mentioned in Celeux and Govaert (1995). I suspect that it is only used to speed up the optimization process.

According to `mclust` document, the upper triangular Cholesky factor for "EEE" and "VVV" are included as part of the output of `me` or `mstep`, or input of `estep`.

April 16, 2014

Our current goal is to try reproducing the optimization done in the M-step from the package `mclust`. I will start by writing the objective function in R, and then use some kind of optimization routine to find the optimal solution.

Based on Celeux and Govaert, the objective function, in general, is the log-likelihood of complete data:

$$F(\theta|\boldsymbol{x}_1,...,\boldsymbol{x}_n,\boldsymbol{c}) = \sum_{k=1}^{K}\sum_{i=1}^{n} c_{ik}\log[p_k\Phi(\boldsymbol{x}_i|\mu_k,\Sigma_k)]$$

Substituting in the normal PDF, and after some algebra, the objective function becomes:

$$F_0(\Sigma_k) = \sum_{k=1}^{K}\sum_{i=1}^{n} c_{ik}(\boldsymbol{x}_i-\hat{\mu}_k)^T\Sigma_k^{-1}(\boldsymbol{x}_i-\hat{\mu}_k) - \sum_{k=1}^{K}\sum_{i=1}^{n} c_{ik}\log|\Sigma_k|$$

where $\hat{\mu}_k$ can be easily estimated, and $c_{ik} = I(\boldsymbol{x}_i \in \text{Cluster } k)$ is given for all i and k.

Here we assume that the data are 2-dimensional ($d = 2$), the number of components in Gaussian mixture is 4 ($K = 4$) and the covariance matrix for component k is parametrized as follows:

$$\Sigma_k = \lambda_k D_k A D_k^T$$

This models assumes that the K clusters have the same shape (A), but potentially different volumes (λ_k) and orientations (D_k). The total number of parameters to estimate, apart from mean and mixing proportion parameters, is $K\beta-(K-1)(d-1)$, where $\beta = d(d+1)/2$.

According to Celeux and Govaert (1995), this model does not have a closed-form solution and optimization needs to be done by iteratively.

April 23, 2014

We wrote R code for different parts of EM algorithm, assuming that the covariance structure is "EEE" (same shape, orientation and volume for all clusters). But ran into problems when we try running it with a simulated data set. This data set runs perfectly okay with `me` function from package `mclust`.

The most recent error message is: `Error in solve.default():system is computationally singular`.

A traceback at the program shows the following:

```
Browse[1]> traceback()
6: solve.default(mat)
5: solve(mat) at objective function_EEE.R#44
4: fn(par, ...)
3: (function (par)
   fn(par, ...))(c(0.438203727931431, 1.64666457028043, 2.74573947601847,
   10.3178307855187))
2: optim(par = initial.cov, obj.fun, z = z, data = data) at M-step_EEE.R#14
1: MstepEEE(z, data, ini.cov) at #3
```

and the message remains the same no matter what initial value for covariance matrix we use. It seems like that the singularity occurs in the function `optim()`, but we're not sure at this point whether it is this function doing something that causes this or it comes from our own function.

Update: It seems the problem comes from the fact that we have not told R that the matrix should be positive definite. It's still unclear where or how to add these constraints in the code.

Update: Can use elements Cholesky decomposition factor as input, so that the product is positive definite.

Input for objective matrix will be the elements of the lower triangular matrix, and the function `upperTriangle` from GDATA package can be used to construct a lower triangular matrix.

Update: Code has been modified, but need more tuning of the constraints passed to `optim` function in M-step.

April 25, 2014

After the code works, try to think of working examples:

1. Iris data for measurement error-free case. Compare with `mclust`.

2. After we add in the measurement error in the code, we can simulate data with errors.

May 1, 2014

Need to modify code so that the ME function I wrote has the same input/output structure as that in MCLUST package.

I/O of function `meEEE` include:

Input:

1. data matrix (data)
2. initial membership matrix (z)

Output:

1. model name ("EEE")
2. sample size (n)
3. number of coordinates (p)
4. number of clusters (G)
5. final membership estimates (zhat)
6. parameter estimates (muhat, sigmahat, phat)
7. log-likelihood
8. information on iteration

To do:

Modify code to VVV case. Use closed form for covariance matrix.

May 5, 2014

To do:

1. Include optimization routine in the VVV case.
2. Look at Di's RNA-Seq data. Compare results with MCLUST.
3. Closed-form solution for VVV after we add estimation error?