ADVANCED BIOSTATISTICS ABSTAT17

EM Algorithm

Lisete Sousa

Department of Statistics and Operations Research, CEAUL
Faculty of Sciences of Lisbon University

IGC, April 10th - 13th, 2017

Contents

Introduction

MLE - Allele frequencies from genotype frequencies

EM Algorithm - Allele frequencies from phenotype frequencies

EM Algorithm - Differentially expressed genes

ESTIMATION OF ALLELE FREQUENCIES

… From Genotype Frequencies

- Maximum likelihood
- Example – Blood type
- Example in `R`

… From Phenotype Frequencies

- Maximum likelihood
- Complicated system to solve…
- EM algorithm
- Example – Blood type
- Example in `R`

IDENTIFICATION OF DIFFERENTIALLY EXPRESSED GENES

- EM algorithm for mixture of distributions
- Example 1 – Normal-Uniform – package `nudge`
- Example 2 – Gamma-Gamma-Gamma

Introduction

The Expectation-Maximization (EM) algorithm is a parameter estimation method which falls into the general framework of maximum likelihood estimation, and is applied in cases where part of the data can be considered to be incomplete, or “hidden”.

It is essentially an iterative optimization algorithm which, at least under certain conditions, will converge to parameter values at a local maximum of the likelihood function.

The EM algorithm has become a popular tool in statistical estimation problems involving incomplete data, or in problems which can be posed in a similar form, such as mixture estimation or censored data.

Maximum likelihood estimation of allele frequencies from genotype frequencies

Consider the AB0 blood group system (3 alleles). Genotype frequencies, under Hardy-Weinberg equilibrium, are as follows:

AA	A0	BB	B0	AB	00
p^2	$2pr$	q^2	$2qr$	$2pq$	r^2

NOTE: Here, allele frequencies p_A, p_B and p_0 are p, q and r, respectively.

A set of n individuals contains:

- X_{AA} - Number of individuals with genotype AA;
- X_{A0} - Number of individuals with genotype A0;
- X_{BB} - Number of individuals with genotype BB;
- X_{B0} - Number of individuals with genotype B0;
- X_{AB} - Number of individuals with genotype AB;
- X_{00} - Number of individuals with genotype 00.

Thus,

$$(X_{AA}, X_{A0}, X_{BB}, X_{B0}, X_{AB}, X_{00}) \frown \text{Multinomial}(n; p^2, 2pr, q^2, 2qr, 2pq, r^2)$$

The corresponding likelihood function is

$$L(p, q, r | n_{AA}, n_{A0}, n_{BB}, n_{B0}, n_{AB}, n_{00}) =$$

$$\begin{aligned} = & \quad \frac{n!}{n_{AA}!n_{A0}!n_{BB}!n_{B0}!n_{AB}!n_{00}!} \times \\ & \quad \times (p^2)^{n_{AA}}(2pr)^{n_{A0}}(q^2)^{n_{BB}}(2qr)^{n_{B0}}(2pq)^{n_{AB}}(r^2)^{n_{00}} \propto \\ \propto & \quad p^{2n_{AA}+n_{A0}+n_{AB}} q^{2n_{BB}+n_{B0}+n_{AB}} r^{n_{A0}+n_{B0}+2n_{00}} \end{aligned}$$

Note: $n = nAA + nA0 + nBB + nB0 + nAB + n00$

By Lagrange multiplier, we obtain the maximum likelihood estimates:

- $\hat{p} = \frac{1}{2n}(2n_{AA} + n_{A0} + n_{AB}) = \left(n_{AA} + \frac{1}{2}n_{A0} + \frac{1}{2}n_{AB}\right)/n$
- $\hat{q} = \frac{1}{2n}(2n_{BB} + n_{B0} + n_{AB}) = \left(n_{BB} + \frac{1}{2}n_{B0} + \frac{1}{2}n_{AB}\right)/n$
- $\hat{r} = \frac{1}{2n}(2n_{00} + n_{A0} + n_{B0}) = \left(n_{00} + \frac{1}{2}n_{A0} + \frac{1}{2}n_{B0}\right)/n$

Example 1: For the data

AA	AB	A0	BB	B0	00
12	35	21	15	30	9

the maximum likelihood estimates are: $\hat{p} = 0.328$, $\hat{q} = 0.389$ and $\hat{r} = 0.283$.

How to do this in R?

Read the data:

```
> nAA<-12
> nAB<-35
> nA0<-21
> nBB<-15
> nB0<-30
> n00<-9
> n<-nAA+nAB+nA0+nBB+nB0+n00
```

Calculate de estimates:

```
> p<-(nAA+nA0/2+nAB/2)/n
> q<-(nBB+nB0/2+nAB/2)/n
> r<-(n00+nA0/2+nB0/2)/n
> p;q;r
[1] 0.3278689
[1] 0.3893443
[1] 0.2827869
```

EM algorithm for estimating the allele frequencies from phenotype frequencies

Situation: Grouped Data

The estimation is relatively easy if the genotype frequencies are known. However, this is not usual. For AB0 blood group system, for example, what we frequently know is the blood type.

Example 1 (cont.):

The previous data would then be:

A	B	AB	0
33	45	35	9

The variables representing the number of individuals with each genotype and phenotype, can be organized according to the following scheme:

$$\begin{tikzcd}
 & X_A \arrow[d] & & & X_B \arrow[d] & & X_{AB} \arrow[d] & X_0 \arrow[d] \\
 & p_A \arrow[dl] \arrow[dr] & & & p_B \arrow[dl] \arrow[dr] & & p_{AB} \arrow[d] & p_0 \arrow[d] \\
X_{AA} \arrow[d] & & X_{A0} \arrow[d] & X_{BB} \arrow[d] & & X_{B0} \arrow[d] & X_{AB} \arrow[d] & X_{00} \arrow[d] \\
p_{AA} \arrow[d, equal] & & p_{A0} \arrow[d, equal] & p_{BB} \arrow[d, equal] & & p_{B0} \arrow[d, equal] & p_{AB} \arrow[d, equal] & p_0 \arrow[d, equal] \\
p^2 & & 2pr & q^2 & & 2qr & 2pq & r^2
\end{tikzcd}$$

How to estimate p, q and r in this case?

Consider the model:

A	B	AB	0
$p^2 + 2pr$	$q^2 + 2qr$	$2pq$	r^2

Maximum likelihood estimation:
A complicated equation system.

EM algorithm:
Useful when information is missing. In this particular case, there is no information about n_{AA}, n_{A0}, n_{BB} and n_{B0}.

Step 1: Choose initial values for p, q and r. Let them be represented by $p_{(0)}$, $q_{(0)}$ and $r_{(0)}$.

Step 2 (E): Calculate an approximate value for the missing terms, using the previous initial values.

- $n^*_{AA} = n_A \frac{p^2_{(0)}}{p^2_{(0)}+2p_{(0)}r_{(0)}} = n_A \frac{p_{(0)}}{p_{(0)}+2r_{(0)}}$

  Note that $X_A \frown \text{Bi}(n, p^2+2pr)$ and $X_{AA}|X_A \frown \text{Bi}(n_A, \frac{p^2}{p^2+2pr})$.

- $n^*_{A0} = n_A \frac{2r_{(0)}}{p_{(0)}+2r_{(0)}}$
- $n^*_{BB} = n_B \frac{q_{(0)}}{q_{(0)}+2r_{(0)}}$
- $n^*_{B0} = n_B \frac{2r_{(0)}}{q_{(0)}+2r_{(0)}}$

Step 3 (M): Calculate the first iterate for p, q and r, from maximum likelihood estimators. Here, missing information is substituted by the approximate values obtained in the previous step.

- $p_{(1)} = \frac{1}{2n}(2n^*_{AA} + n^*_{A0} + n_{AB})$
- $q_{(1)} = \frac{1}{2n}(2n^*_{BB} + n^*_{B0} + n_{AB})$
- $r_{(1)} = \frac{1}{2n}(2n_{00} + n^*_{B0} + n^*_{A0})$

Step 4: Repeat step 2, substituting $p_{(0)}$, $q_{(0)}$ and $r_{(0)}$ by $p_{(1)}$, $q_{(1)}$ and $r_{(1)}$.

Step 5: Repeat the procedure until convergence.

Initial values:

1. Start with 0.3(3) for $p_{(0)}$, $q_{(0)}$ and $r_{(0)}$.
2. Bernstein's suggestion:

$$p_{(0)} = 1 - \sqrt{(n_0 + n_B)/n}$$
$$q_{(0)} = 1 - \sqrt{(n_0 + n_A)/n}$$
$$r_{(0)} = \sqrt{n_0/n}$$

Example 1 (cont.):

EM algorithm estimates are $\hat{p} = 0.331$, $\hat{q} = 0.409$, $\hat{r} = 0.259$.

How to do this in R?

Read the data:

```
> nA<-33
> nB<-45
> nAB<-35
> n0<-9
```

Initialize the parameters (Bernstein's suggestion) – Step 1:

```
> p<-1-sqrt((n0+nB)/n)
> q<-1-sqrt((n0+nA)/n)
> r<-sqrt(n0/n)
> p;q;r
```

```
[1] 0.3347009
[1] 0.4132613
[1] 0.2716072
```

Calculate approximate values for n_{AA}, n_{A0}, n_{BB}, n_{B0} – Step 2 (E):

```
> exp<-function(p,q,r){
+ nAA<-nA*p/(p+2*r)
+ nA0<-nA*2*r/(p+2*r)
+ nBB<-nB*q/(q+2*r)
+ nB0<-nB*2*r/(q+2*r)
+ c(nAA,nA0,nBB,nB0)
+ }
```

Visualize n^*_{AA}, n^*_{A0}, n^*_{BB}, n^*_{B0}, obtained in the first iteration:

```
> exp(p,q,r)
[1] 12.58109 20.41891 19.44300 25.55700
```

NOTE that the first two values sum to 33 (n_A) and the last two sum to 45 (n_B).

Update p, q and r – Step 3 (M):

```
> param<-function(nAA,nA0,nBB,nB0){
+ p<-(2*nAA+nA0+nAB)/(2*n)
+ q<-(2*nBB+nB0+nAB)/(2*n)
+ r<-(2*n0+nA0+nB0)/(2*n)
+ c(p,q,r)
+ }
```

Visualize $\hat{p}$, $\hat{q}$ and $\hat{r}$, obtained in the first iteration:

```
> param(nAA,nA0,nBB,nB0)
[1] 0.3278689 0.3893443 0.2827869
```

Iterative procedure – Steps 4 and 5:

```
> i<-0; er<-1
> while(sum(er>=0.00001)>0)
+ { e<-exp(p,q,r)                              # Step E
+   par<-param(e[1],e[2],e[3],e[4])            # Step M
+   er<-abs(c(p,q,r)-c(par[1],par[2],par[3]))  # STOP criteria
+   i<-i+1
+   p<-par[1]; q<-par[2]; r<-par[3]
+   cat(i,p,q,r,"\n") # print the 3 values for each iteration.
+ }
```

```
1 0.3302504 0.4075533 0.2621964
2 0.3309501 0.4085211 0.2605288
3 0.3312228 0.4089185 0.2598587
4 0.3313321 0.4090797 0.2595882
5 0.3313762 0.4091449 0.2594789
6 0.331394 0.4091713 0.2594346
7 0.3314012 0.409182 0.2594167
8 0.3314042 0.4091863 0.2594095
```

Print the final solution in a friendly way:

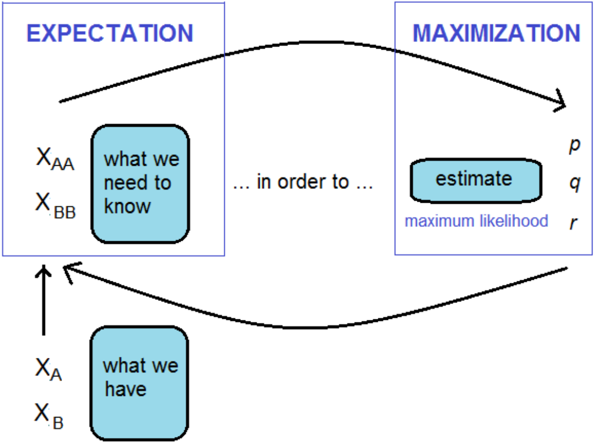
```
> cat("\n Obtained solution after ",i," iterations:\n
+ p^ =",p,
+ "\n q^ =",q,
+ "\n r^ =",r,"\n")

Obtained solution after  8  iterations:

p^ = 0.3314042
q^ = 0.4091863
r^ = 0.2594095
```

In example 1, the estimates of the allele frequencies obtained from both methods are very similar:

	MLE (known genotypes)	EM (unknown genotypes)
$\hat{p}$	0.328	0.331
$\hat{q}$	0.389	0.409
$\hat{r}$	0.283	0.259

Final Remarks

For the situation in which we have grouped data, the EM algorithm may be summarized according to the following scheme:

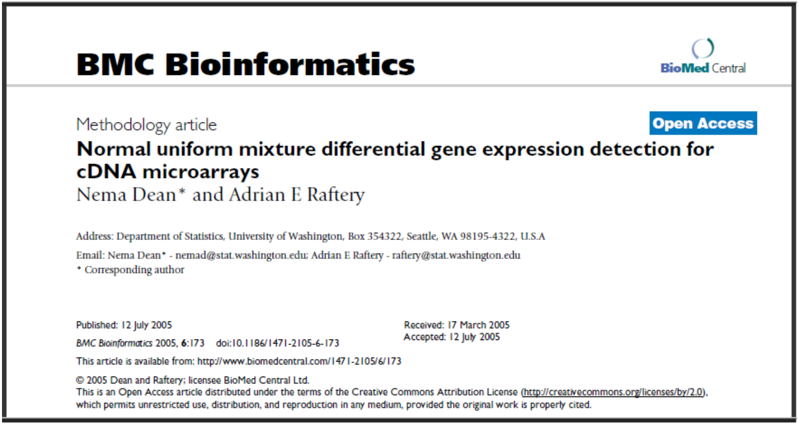


EM algorithm on the identification of differentially expressed genes

Situation: Mixture of distributions

EM algorithm can be applied to other situations, which require more probabilistic calculations. That is the case when we have a mixture of distributions. As these calculations require a more advanced level of theoretical knowledge, we will just show the main idea through some publications and examples:

Normal – Uniform Distributions



BMC Bioinformatics

BioMed Central

Methodology article

Open Access

Normal uniform mixture differential gene expression detection for cDNA microarrays

Nema Dean* and Adrian E Raftery

Address: Department of Statistics, University of Washington, Box 354322, Seattle, WA 98195-4322, U.S.A

Email: Nema Dean* - nemad@stat.washington.edu; Adrian E Raftery - raftery@stat.washington.edu

* Corresponding author

Published: 12 July 2005

Received: 17 March 2005
Accepted: 12 July 2005

BMC Bioinformatics 2005, **6**:173 doi:10.1186/1471-2105-6-173

This article is available from: http://www.biomedcentral.com/1471-2105/6/173

- Dean and Raftery (2005), consider a Normal Uniform mixture model (NUDGE - Normal Uniform Differential Gene Expression) for the normalized log intensities.
- The method uses EM algorithm to estimate membership probabilities for each gene and classify it accordingly.
- The mixture gives posterior probabilities of differential expression which do not need to be adjusted for multiple testing.
- Typically a threshold for probability of being expressed of 0.5 is used for classification of the genes into the two groups (prob.≥ 0.5 for DE genes; prob.< 0.5 for non DE genes).

Theoretical Aspects

- Consider that the only available data are the (averaged) log ratio intensities for n genes - $\{x_1, \ldots, x_n\}$ - this is, the group to which each gene belongs to is not known.
- Conditional on the group, the gene expression level follows either a Gaussian (non DE genes) or an Uniform (DE genes). Hence,

$$f(x_i) = \pi f_{N(\mu,\sigma^2)}(x_i) + (1-\pi) f_{U(a,b)}(x_i) \ , i = 1, \ldots, n.$$

- Because it is not known in advance to which group each gene belongs to, this is viewed as a missing data problem and the Expectation-Maximization (EM) algorithm is used to estimate the model parameters.

- The maximum likelihood estimates of a and b are $\hat{a} = \min\{x_i : i = 1, \ldots, n\}$, and $\hat{b} = \max\{x_i : i = 1, \ldots, n\}$; these do not change during the algorithm.

In iteration j:

- Expectation step: for each data point x_i, $i = 1, \ldots, n$, the membership value for the group of DE genes, is given by

  $$\hat{y}_i^{(j)} = \frac{(1 - \hat{\pi}^{(j-1)}) f_{U(\hat{a},\hat{b})}(x_i)}{\pi^{(j-1)} f_{N(\mu^{(j-1)},(\sigma^{(j-1)})^2)}(x_i) + (1 - \pi^{(j-1)}) f_{U(\hat{a},\hat{b})}(x_i)},$$

  $i = 1, \ldots, n.$

  To start the algorithm initial values for the membership values are needed. The starting value for the label y_i is

  $$\hat{y}_i^{(0)} = \begin{cases} 1, & \text{if } \left|\frac{x_i - \bar{x}}{s_x}\right| > 2 \\ 0, & \text{otherwise} \end{cases},$$

- Maximization step: with expectation values in hand for group membership, recompute plug-in estimates for distribution parameters:

$$\hat{\pi}^{(j)} = \frac{\sum_{i=1}^{n}(1-\hat{y}_i^{(j)})}{n}, \qquad \hat{\mu}^{(j)} = \frac{\sum_{i=1}^{n}(1-\hat{y}_i^{(j)})x_i}{\sum_{i=1}^{n}(1-\hat{y}_i^{(j)})},$$

$$(\hat{\sigma}^{(j)})^2 = \frac{\sum_{i=1}^{n}(1-\hat{y}_i^{(j)})(x_i-\hat{\mu}^{(j)})^2}{\sum_{i=1}^{n}(1-\hat{y}_i^{(j)})}.$$

Advantages and Disadvantages

The main advantages of NUDGE are:

- Very fast;
- Gives probabilities of differential expression, which can also be used to rank the genes in terms of how likely they are to be differentially expressed;
- Can be applied to either single-slide or replicated experiments;

Two disadvantages are pointed out:

- Does not analyze Affymetrix microarrays;
- The output presents a list of DE genes, not specifying which of them are up- and down-regulated.

Example of Application

The main function is `nudge1`. For more information see `nudge` vignette (Dean, 2006).

- `nudge1(logratio,logintensity,dye.swap=FALSE,...)`

  Normalizes data, fits a normal uniform mixture and estimates the probabilities of differential expression for the case of two samples being compared directly.

  - logratio - *matrix or vector of* $\log_2$ *ratios of intensity expressions in 2 samples; rows correspond to genes and columns to replicates.*
  - logintensity - *matrix or vector of total* $\log_2$ *intensities for the 2 samples.*
  - dye.swap - *indicates whether or not the data results from a dye swap.*

In case of comparing both samples with a reference sample, you should use function nudge2.

nudge1 function detects if there is only a single replicate or multiple replicates.

Swirl Zebrafish Dataset

Consider the Swirl Zebrafish Dataset (lr), available through package limma (Smyth et al., 2016).

- Swirl zebrafish data is a direct **two-color design**, and is available in limma.
- The main goal of the Swirl experiment is to identify genes with altered expression in the **swirl mutant** compared to **wild-type** (wt) zebrafish.

Download the Data

The data files are available at `http://bioinf.wehi.edu.au/limma/`, as a zipped file.

For details see the script for downloading the data at `Script Examples_07 Expectation Maximization.R`.

Apply `nudge` package

```
> result <- nudge1(logratio=lr,logintensity=li,dye.swap=T)
> names(result)
```

```
[1] "pdiff" "lRnorm" "mu" "sigma" "mixprob"
[8] "a" "b" "loglike" "iter"
```

```
> result$mu
[1] -0.3486176
> result$sigma
[1] 0.6102006
> result$mixprob
[1] 0.9848056
> result$a
[1] -3.283844
> result$b
[1] 4.873135
> result$iter
[1] 17
```

- `pdiff` is a vector with the estimated posterior probabilities of being in the group of differentially expressed genes;
- `lRnorm` is a vector with the normalized log ratios;
- `mixprob` The prior/mixing probability of a gene being in the group of genes that are not differentially expressed;
- `loglike` is the log likelihood for the fitted mixture model;
- `iter` is the number of iterations run by the EM algorithm until either convergence or iteration limit is reached.

It is possible to look, for example, at the top 20 genes (names or numbers) with the highest probability of differential expression,

```
> s <- sort(result$pdiff, decreasing = T, index.return = T)
> attach(RG)
> rownames(lr) <- genes$Name
> cbind(rownames(lr)[s$ix[1:20]], round(s$x[1:20], 2))
```

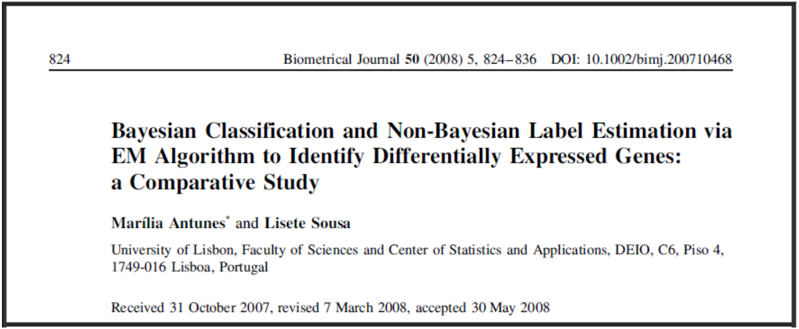
```
       [,1]     [,2]
 [1,]  "18-F10" "1"
 [2,]  "BMP2"   "1"
 [3,]  "BMP2"   "1"
 [4,]  "Dlx3"   "1"
 [5,]  "Dlx3"   "1"
...
 [16,] "27-E17" "0.98"
 [17,] "18-G13" "0.98"
 [18,] "vent"   "0.98"
 [19,] "vent"   "0.98"
 [20,] "vent"   "0.98"
```

- The top 20 genes correspond to a posterior probability of being DE higher than 0.98.
- There is also the possibility of looking at the number (and "names") of genes with a probability of differential expression being greater than a given threshold (usually 0.5).

```
> thresh <- 0.5
> sum(result$pdiff >= thresh)
[1] 75
```

And so on...

Gamma – Gamma – Gamma Distributions



824 Biometrical Journal **50** (2008) 5, 824–836 DOI: 10.1002/bimj.200710468

Bayesian Classification and Non-Bayesian Label Estimation via EM Algorithm to Identify Differentially Expressed Genes: a Comparative Study

Marília Antunes* **and Lisete Sousa**

University of Lisbon, Faculty of Sciences and Center of Statistics and Applications, DEIO, C6, Piso 4,
1749-016 Lisboa, Portugal

Received 31 October 2007, revised 7 March 2008, accepted 30 May 2008

- Antunes and Sousa (2008) consider a simple mixture model at the data level, as in Dean and Raftery (2005), with the difference that they consider that each gene is either down-regulated, non-DE or up-regulated and three Gamma distributions are assumed for the intensity measures.

- In addition to the EM algorithm, the authors apply a second method for gene classification, which is based on an hierarchical Bayesian model.

Acknowledgements:

Antónia Turkman (DEIO–FCUL), for allowing the use of some material produced by us in previous courses.

Bibliography:

Antunes, M. and Sousa, L. (2008). Bayesian classification and non-Bayesian label estimation via EM algorithm to identify differentially expressed genes: a comparative study. *Biometrical Journal* 50(5), 824â-836.

Dean, N. (2006). The Normal Uniform Differential Gene Expression: (nudge) Detection Package. Bioconductor.

Dean, N. and Raftery, A.E. (2005). Normal uniform mixture differential gene expression detection for cDNA microarrays. *BMC Bioinformatics* 6, 173.

McLachlan, G. and Krishnan, T. (2008). *The EM Algorithm and Extensions*, 2nd Edition, Wiley.

Smyth, G.K., Ritchie, M., Thorne, N., Wettenhall, J. and Shi, W. (2016). User's Guide. limma: Linear Models for Microarray Data. Bioconductor.