The Potential of Recent Developments in Nonparametric Mixture Distributions

Dankmar Böhning

Free University Berlin / Humboldt University at Berlin

email: boehning@zedat.fu-berlin.de

natural genesis of *mixture distributions*

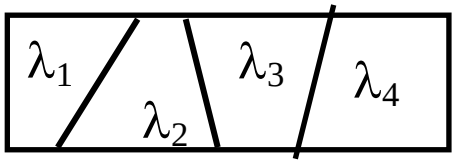
through **unobserved** heterogeneity:

homogeneity

one-parametric density $f(x,\lambda)$

λ parameter of the population,
x in sample space

heterogeneity



density in subpopulation j: $f(x,\lambda_j)$

latent variable Z describing population membership

joint density f(x,z) with

$$f(x,z) = f(x|z)f(z) = f(x,\lambda_z)p_z$$

marginal density:

$$f(x,P) = f(x,\lambda_1)p_1 + f(x,\lambda_2)p_2 + \ldots + f(x,\lambda_k)p_k$$

$P = (\lambda_1 \ldots \lambda_k; p_1 \ldots p_k)$ is *mixing distribution*

P needs to be *estimated* :

log-likelihood

$l(P)$ = **Error!= Error!**{ **Error!**}

Example 1 (popular)

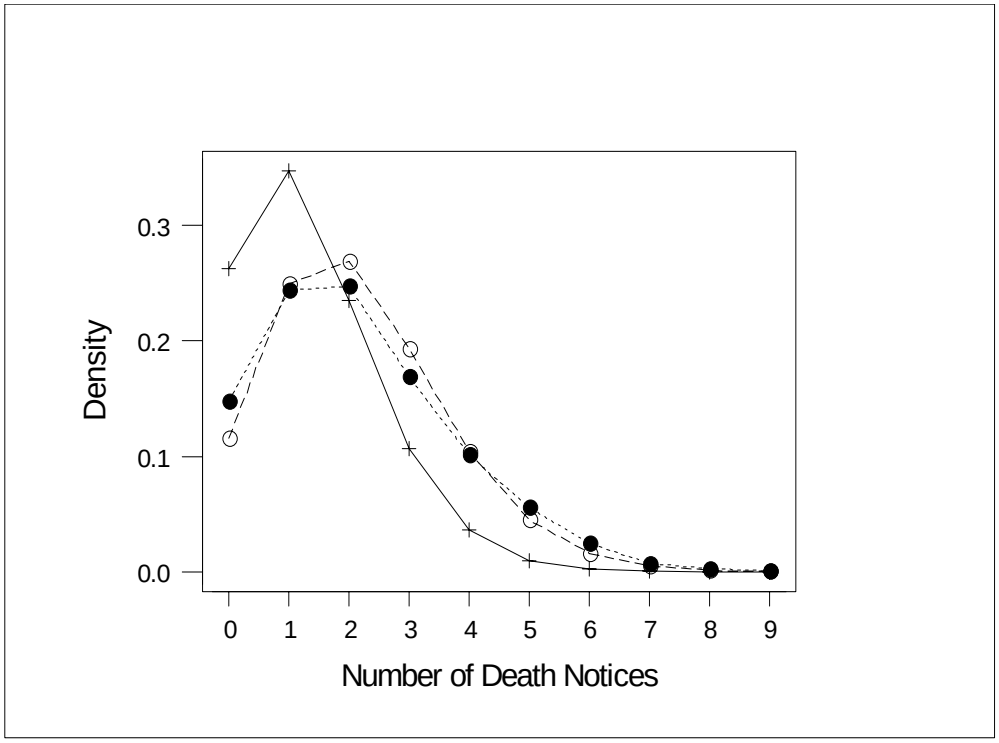
number of death notices in the *Times* newspaper, 1910-1912, for women aged 80 years and over,

Number of notices x_i	0	1	2	3	4	5	6	7	8	9
Frequency	162	267	271	185	111	61	27	8	3	1

it was suggested
(Hasselblad 1969, Titterington, Smith, and Makov 1985)

k = 2 components
with parameter estimates

$\lambda_1 = 1.2561$ $\lambda_2 = 2.6634$
$p_1 = 0.3599$ $p_2 = 0.6401$



Empirical density (solid circle), estimated Poisson density (+), and estimated nonparametric Poisson mixture density (circle)

interpretation

different pattern of death for winter (z=1) and summer (z=2)

[discrete Error!
or …. **continuous Error!** mixing ? **]**

Consider

$$\Gamma = \{ (f(x_1,\lambda), \ldots, f(x_n,\lambda))^T \mid \lambda \in \Lambda \}$$

and conv $(\Gamma) = \{ \Sigma_j \, p_j \mathbf{f}_j \mid \mathbf{f}_j \in \Gamma \}$

Then … (Th. of Carathéodory):

every point in conv (Γ) can be represented by a **discrete convex sum** with at most n points !

Global Maximization and Gradient Function

log-likelihood *l concave* functional on the set of *all* discrete probability distributions Ω (number of components k is **not** fixed)!

major tool: *directional derivative*

$\Phi(P,Q) = \lim_{\alpha \to 0}$ **Error!**

= **Error!**= **Error!**- n

in *particular,* for one-point mass at λ: Q_λ

$\Phi(P,Q_\lambda)$ = **Error!**- n = **Error!**- n

$\Phi(P,Q_\lambda)$ = **Error!**- n

gradient function:

d(λ,P) : = **Error!Error!**

Example (Poisson)**:**

$f(x,\lambda) = Po(x,\lambda) = e^{-\lambda}\lambda^x/x!$

then d(λ,P) =

Error!Error!= Error! Error!

mixture maximum likelihood theorem
(Lindsay 83a,b *Ann. Statist.*; Böhning 82 *Ann. Statist.*)

a) $P;\hat{}$ is NPMLE

$\Leftrightarrow \Phi(P;\hat{},Q_\lambda) \leq 0$ for all λ

$\Leftrightarrow d(\lambda,P;\hat{}) \leq 1$ for all λ

- $d(\lambda,P;\hat{}) = 1$

for all support points λ of $P;\hat{} =$
$\left(\lambda;\hat{}_1 \ldots \lambda;\hat{}_{k;\hat{}}; p;\hat{}_1 \ldots p;\hat{}_{k;\hat{}}\right)$

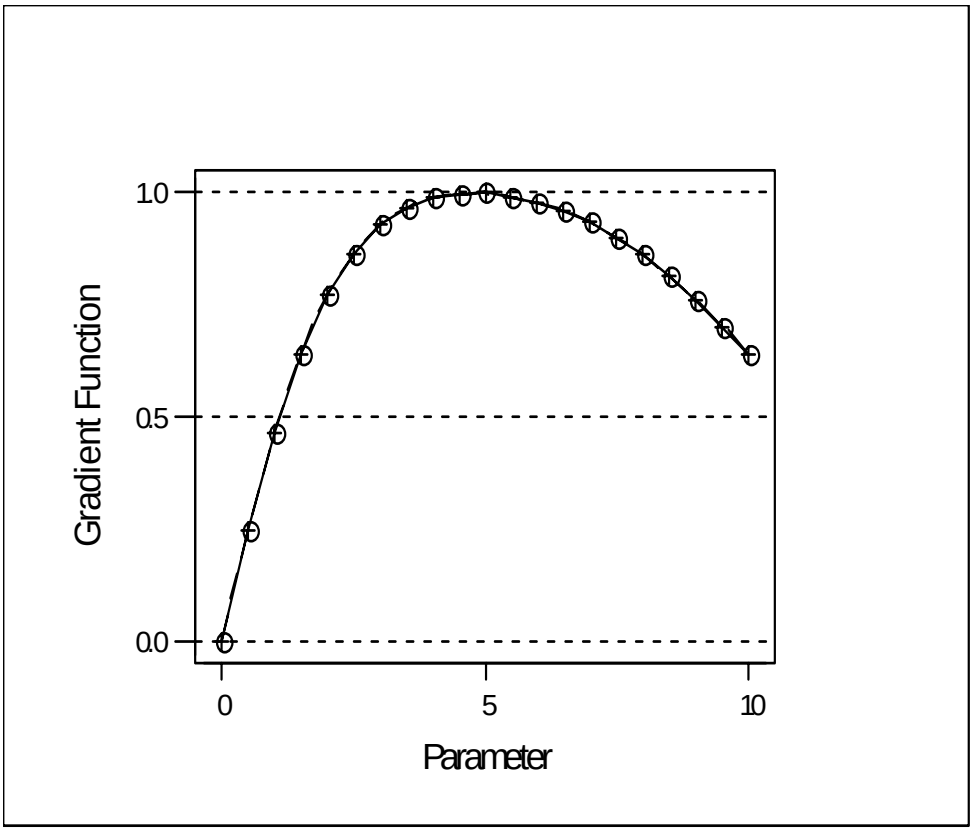
what is it good for ? (I)

simulated data set of size n=100 from a homogeneous Poisson distribution with λ= 5:

x_i	1	2	3	4	5	6	7	8	9	10
frequency	2	10	17	20	19	12	10	4	4	2

$\bar{x}; = 4.78$

gradient function $d(\lambda, \bar{x};)$



conclusion: $\bar{x};$ = NPMLE

(no need for further algorithmic iteration)

what is it good for ? (II)

Example 2 (popular)

Simar (1976) pioneer in NPMLE for mixtures of Poisson distributions

Accident data of Thyrion (1960) used by Simar (1974)

x_i	0	1	2	3	4	5	6	7
frequency	7840	1317	239	42	14	4	4	1

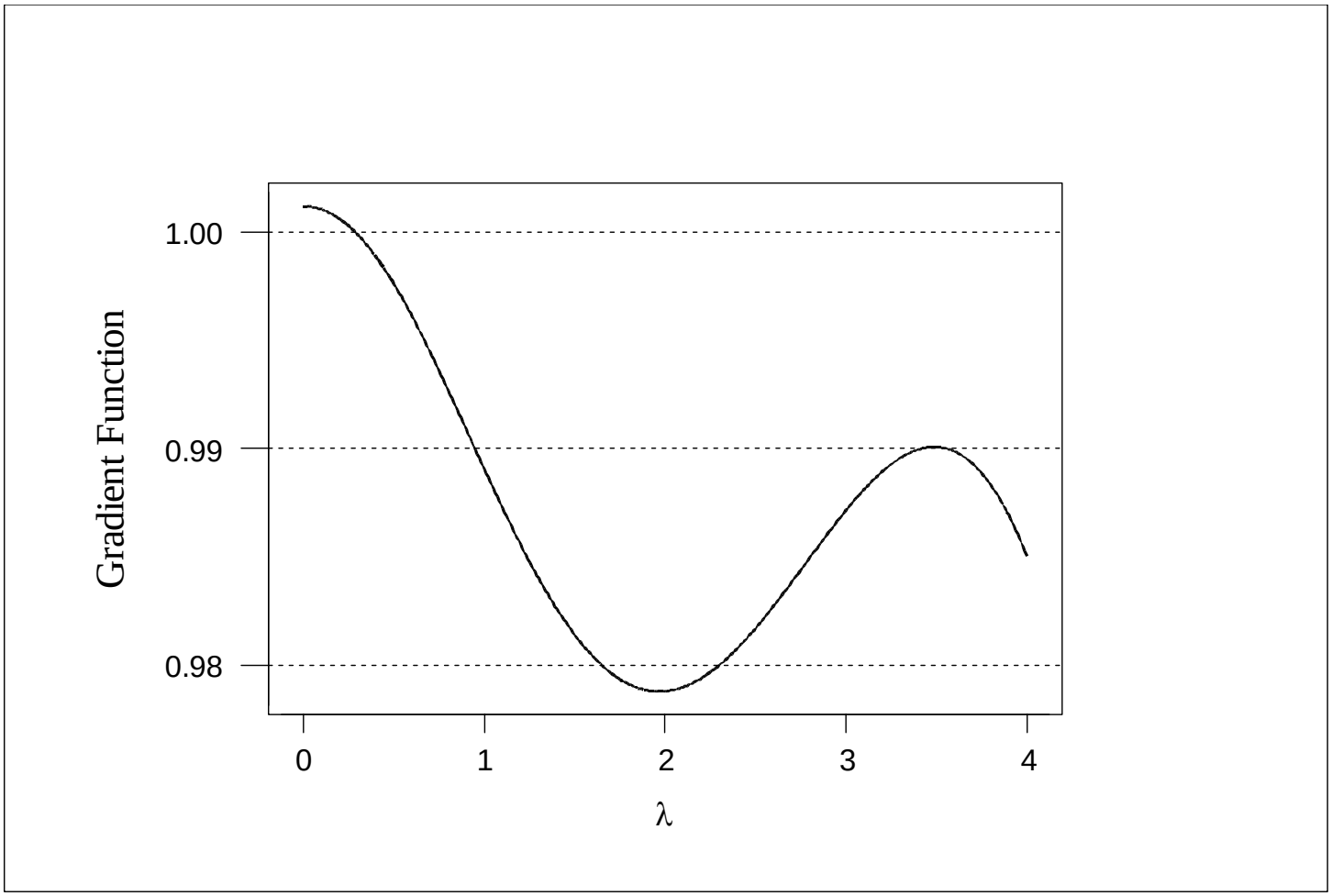
NPMLE given by Simar is

$$P;^{\wedge} =$$

Error!

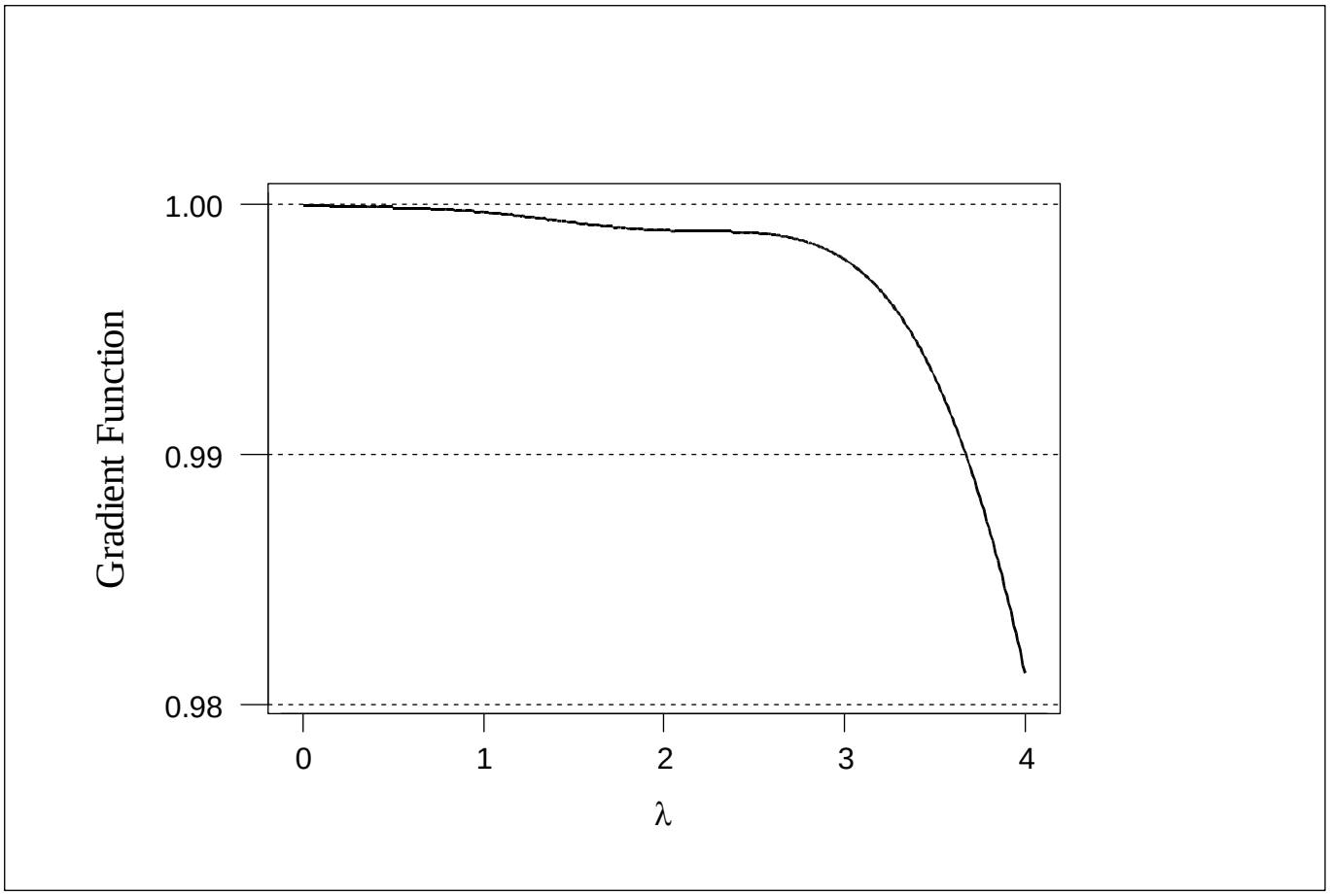
this NPMLE occurs in the lit. frequently including Carlin and Louis (1996, p. 74, Table 3.2)

Gradient function for the accident data of Simar (1976) and the estimator of P given by Simar



Gradient function for the accident data of Simar (1976) and the nonparametric maximum likelihood estimator of P given as

$$\hat{P}; = \begin{pmatrix} 0. & 0.3356 & 2.5454 \\ ; & 0.4184 & 0.5730 & 0.0087 \end{pmatrix}$$



globally converging algorithms

VDM

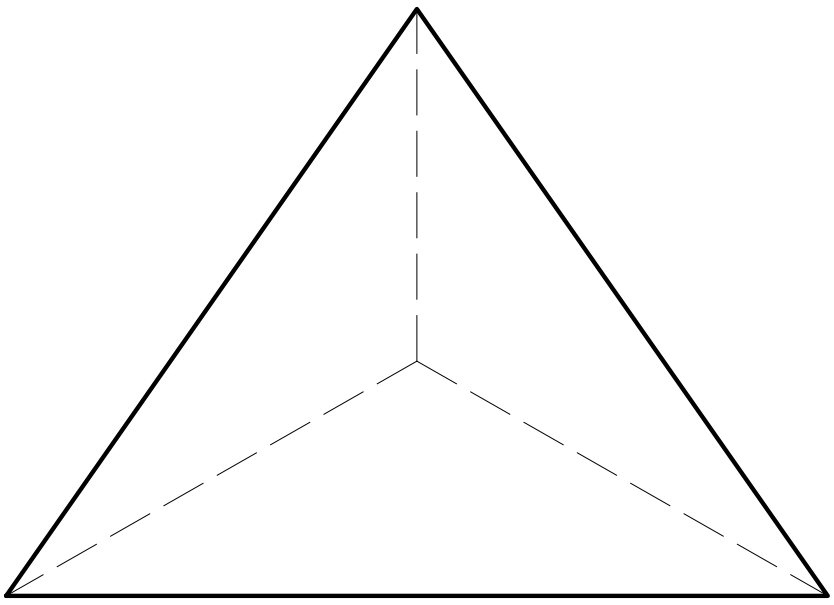
vertex direction

$$(1-\alpha)P + \alpha Q_{\lambda}$$

step 1. choose λ_{max} to maximize $d(P,\lambda)$ in λ

step 2. choose α_{max} to maximize

$$l((1-\alpha)P + \alpha Q_{\lambda_{max}})$$



illustration

let

$$P = (\lambda_1\ \lambda_2\ \lambda_3;\ p_1\ p_2\ p_3)$$

and $\lambda_{max} = \lambda_4$, then

$$(1-\alpha_{max})\ P + \alpha_{max}\ Q_{\lambda_{max}} =$$
$$(\lambda_1\ \ \lambda_2\ \lambda_3\ \ \ \lambda_4;\ \ p'_1\ p'_2\ p'_3\ \ \alpha_{max})$$

with $p'_j = (1-\alpha_{max})\ p_j$ for $j = 1,2,3.$

disadvantage of VDM

- slow in convergence
- tendency to generate clusters of components

VEM

bad vertex direction

$$P + \alpha P(\lambda_{min}) \{Q_{\lambda_{max}} - Q_{\lambda_{min}}\}$$

step 1. choose λ_{max} to maximize $d(P,\lambda)$ in λ

step 2. choose λ_{min} to minimize $d(P,\lambda)$ in support of P

step 3. choose α_{max} to maximize

$$l\,(P + \alpha P(\lambda_{min}) \{Q_{\lambda_{max}} - Q_{\lambda_{min}}\})$$

in α

illustration

let

$$P = (\lambda_1\ \lambda_2\ \lambda_3;\ p_1\ p_2\ p_3)$$

and $\lambda_{min} = \lambda_2$, $\lambda_{max} = \lambda_4$, then

$$P + \alpha_{max}\, P(\lambda_{min})\, \{Q_{\lambda_{max}} - Q_{\lambda_{min}}\} =$$

Error!

if $\alpha_{max} = 1$, then

$$P + P(\lambda_{min})\, \{Q_{\lambda_{max}} - Q_{\lambda_{min}}\} =$$

$$(\lambda_1\ \ \lambda_3\ \ \lambda_4; p_1\ \ p_3\ \ p_2)$$

the component λ_2 is **exchanged** with λ_4
(therefore the name vertex-exchange method)

advantages

- VEM is converging much better than VDM
- VEM can discard “bad” components easily

EM algorithm for fixed number of components k

the complete likelihood in the mixture model

Error!Error! $f(x_i,\lambda_j)$**Error!**p_j**Error!**

where z_{ij} are n unobserved realisations of k component-indicators $Z_{i1}, \ldots, Z_{ik}$

leading to the complete log-likelihood

$$l_{com}(P) = \Sigma_i \Sigma_j z_{ij} \log(p_j) + \Sigma_i \Sigma_j z_{ij} \log f(x_i,\lambda_j)$$

E-step

$$E(Z_{ij} \mid P, x) = \textbf{Error!} =: e_{ij}$$

M-step

$$E(l_{com}(P)) = \Sigma_i \Sigma_j e_{ij} \log(p_j) + \Sigma_i \Sigma_j e_{ij} \log f(x_i,\lambda_j)$$

maximization leads to **new iterates**:

$$p_j^{(new)} = \Sigma_i e_{ij} / n$$

$\lambda_j^{(new)}$ = depends on form of $f(x,\lambda)$

[often = $\Sigma_i e_{ij}x_i / \Sigma_i e_{ij}$]

the problem of multiple maxima

Seidel et al. (2000): null-distribution of LRS depends on choice of initial value for EM algorithm

an illustration

sample of size 100 from exponential

$$f(x,\lambda) = 1/\lambda \exp(-x/\lambda) \quad (\lambda=1)$$

this test data set is available from my web-site:

www.medizin.fu-berlin.de/sozmed/bo1.html

k=2 (two components)

set	**initial values**		**EM iterate**		$l(P_{EM})$
	mean	weight	mean	weight	
1	1	0.5	0.7296	0.5749	-73.3814
	2	0.5	0.8154	0.4251	
2	0.5	0.5	0.7590	0.4781	-73.3555
	1.0	0.5	0.7726	0.5219	
extreme values					
3	0.001	0.5	0.0019	0.0235	-71.0982
	3.700	0.5	0.7845	0.9765	
quartiles					
4	0.180	0.5	0.0239	0.0939	-69.0262
	1.280	0.5	0.8430	0.9061	
5	0.5	0.5	0.7552	0.5091	-73.3566
	1.5	0.5	0.7774	0.4909	

a globally convergent algorithm

idea: use gradient function for improvement

step 0. choose and fix number of components k;
choose arbitrary starting value
$P = (\lambda_1 \lambda_2 \ldots \lambda_k; p_1 p_2 \ldots p_k)$
for EM algorithm

step 1. use EM algorithm to iterate P_{EM}

step 2. determine λ_{max} to maximize $d(P_{EM},\lambda)$ in λ;
determine λ_{min} to minimize $d(P_{EM},\lambda)$ in **support** of P_{EM}

step 3. (Exchange λ_{max} with λ_{min}):

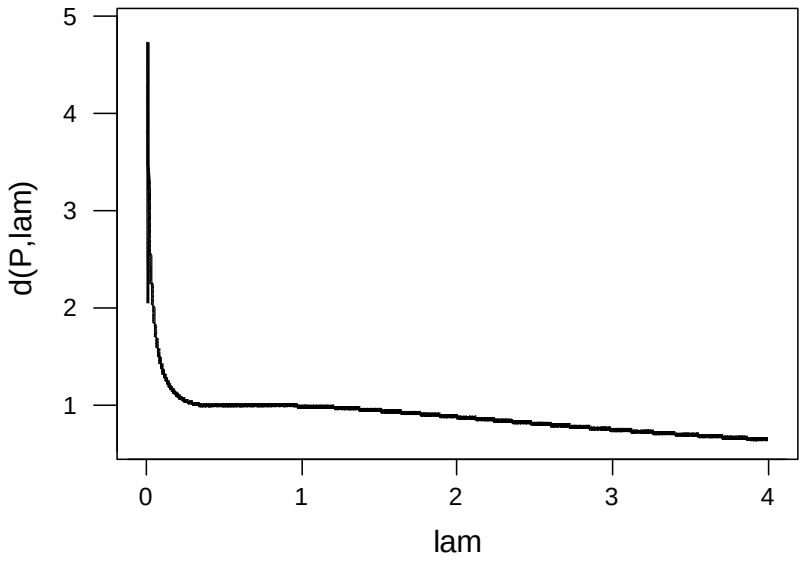
$$P = P_{EM} + P_{EM}(\lambda_{min}) \{Q_{\lambda_{max}} - Q_{\lambda_{min}}\}$$

and go to step 1.

how does this work in practice

k=2, set 2

iteration	**initial values**		**EM iterate**		$l(P_{EM})$
	λ_j	p_j	λ_j^{EM}	p_j^{EM}	
1	0.5	0.5	0.7590	0.4781	-73.3555
	1.0	0.5	0.7726	0.5219	
	λ_{max} = 0.002		λ_{min}=0.7590		
2	0.0020	0.4781	0.0239	0.0939	-69.0263
	0.7726	0.5219	0.8430	0.9061	
	λ_{max} = 0.002		λ_{min} = 0.0239		
3	*identical to step 2 $\Rightarrow$ stop!*				



k=3 (three components)

set	**initial values**		**EM iterate**		$l(P_{EM})$
	mean	weight	mean	weight	
1	1	1/3	0.7620	0.4046	-73.3550
	2	1/3	0.7685	0.2944	
	3	1/3	0.7693	0.3044	
extreme values					
2	0.001	1/3	0.0019	0.0235	-71.0983
	0.570	1/3	0.7831	0.5724	
	3.700	1/3	0.7863	0.4042	
quartiles					
3	0.180	1/3	0.0239	0.0939	-69.0262
	0.570	1/3	0.8430	0.3740	
	1.280	1/3	0.8430	0.5321	

results using global algorithm

k=3, set 1

iterat.	**initial values**		**EM iterate**		$l(P_{EM})$
	λ_j	p_j	λ_j^{EM}	p_j^{EM}	
1	1	1/3	0.7620	0.4046	-73.3550
	2	1/3	0.7685	0.2944	
	3	1/3	0.7693	0.3044	
	λ_{max} = 0.002		λ_{min}=0.7620		
2	0.0020	0.4046	0.0239	0.0939	-69.0263
	0.7685	0.2944	0.8430	0.9061	
	0.7693	0.3044	-	-	
	k is reduced to k = 2				

what to do?

dimension adjustment of algorithm (to keep number of components to be k)

step 1. use EM algorithm to iterate P_{EM}

step 1.1. if #components = k, go to step 2.

else (#components of P_{EM} = k-1), determine λ_{max} to maximize $d(P_{EM},\lambda)$ in λ and set

$$P = (1-\alpha_{max})P_{EM} + \alpha_{max} Q_{\lambda_{max}}$$

and go to step 1.

step 2. determine λ_{max} to maximize $d(P_{EM},\lambda)$ in λ;
determine λ_{min} to minimize $d(P_{EM},\lambda)$ in **support** of P_{EM}

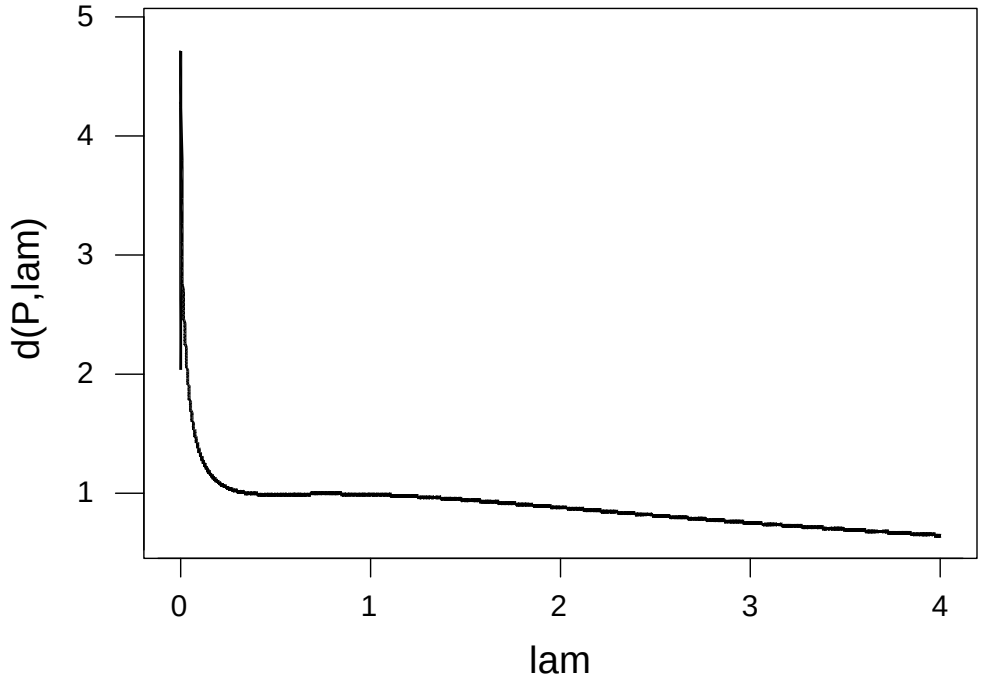
step 3. (Exchange λ_{max} with λ_{min}):

$$P = P_{EM} + P_{EM}(\lambda_{min}) \{Q_{\lambda_{max}} - Q_{\lambda_{min}}\}$$

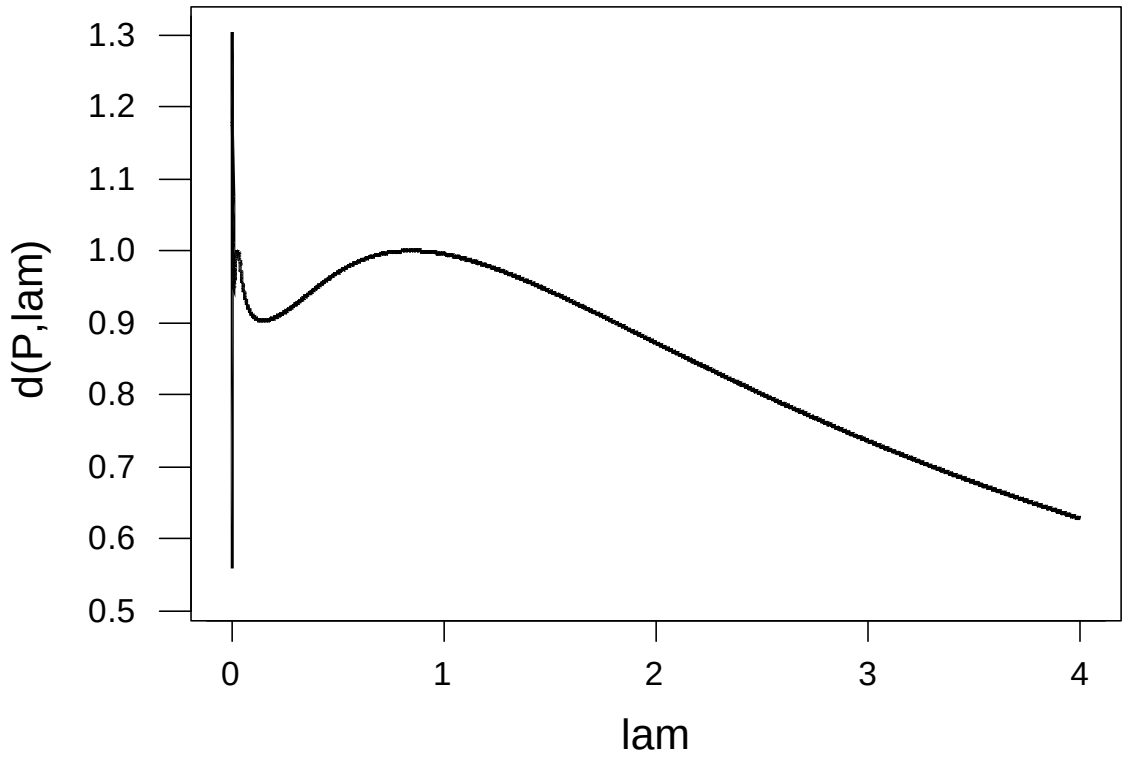
and go to step 1.

k=3, set 1

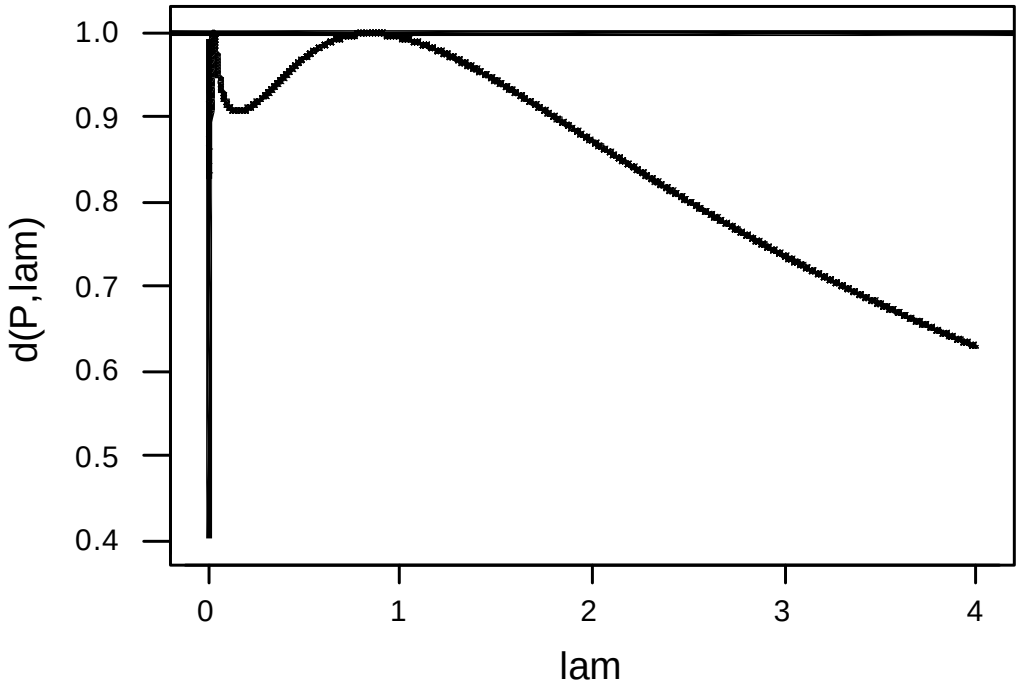
iterat.	**initial values**		**EM iterate**		$l(P_{EM})$
	λ_j	p_j	λ_j^{EM}	p_j^{EM}	
1	1	1/3	0.7620	0.4046	-73.3550
	2	1/3	0.7685	0.2944	
	3	1/3	0.7693	0.3044	
	λ_{max} = 0.002		λ_{min}=0.7620		
	vertex exchange step				
2	0.0020	0.4046	0.0239	0.0939	-69.0263
	0.7685	0.2944	0.8430	0.9061	
	0.7693	0.3044	-	-	
	k is reduced to k = 2				
	λ_{max} = 0.0020		α_{max} = 0.0051		
	vertex direction step				
3	0.0239	0.0934	0.0271	0.0825	-68.8691
	0.8430	0.9061	0.8419	0.9073	
	0.0020	0.0051	0.0017	0.0102	



step 1 (Set1)



step 2 (Set1)



step 3 (Set1)

since for $\hat{P} = (0.0020\ 0.0239\ 0.8430; 0.0051\ 0.0934\ 0.9015)$,
$d(\lambda, \hat{P}) \leq 1$ for all $\lambda \Rightarrow \hat{P}$ is NPMLE

some idea to choose α_{max}

consider

$$\varphi(\alpha) = l((1-\alpha)P + \alpha\, Q_\lambda)$$
$$= \Sigma_x \log \{ (1-\alpha)\, f(x,P) + \alpha\, f(x,\lambda) \}$$

then derivatives are of simple structure:

$$\varphi'(\alpha) \,|_{\alpha=0} = \Sigma_x \textbf{ Error!} = \Sigma_x\, g(x,\lambda,P)$$

$$\varphi''(\alpha) \,|_{\alpha=0} = -\, \Sigma_x \textbf{ Error!}$$

$$= -\, \Sigma_x\, g(x,\lambda,P)^2 \le 0$$

suggestion:

$$\alpha_{max} = \text{Newton-Raphson-Correction}$$

$$= \Sigma_x\, g(x,\lambda,P) \;/\; \Sigma_x\, g(x,\lambda,P)^2$$

note: $\alpha_{max} > 0$, if $d(\lambda,P) > 1$