EM and MM Algorithms

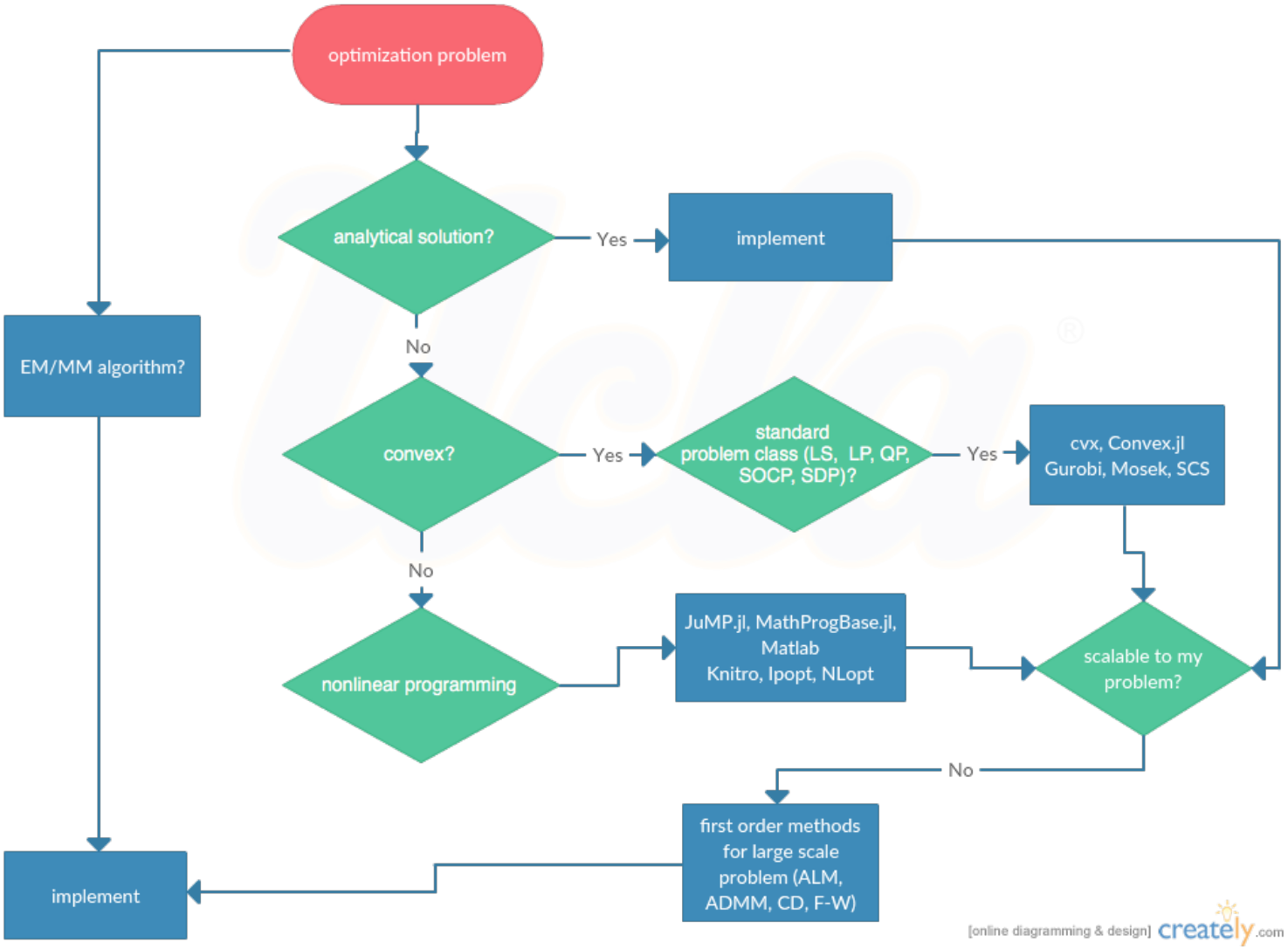
Hua Zhou

Department of Biostatistics
University of California, Los Angeles

Aug 10, 2016
SAMSI OPT Summer School

Slides and Jupyter Notebook for the lab session are available at:
https://github.com/Hua-Zhou/Public-Talks

My flowchart for solving optimization problems



Outline

Problem setup

We consider continuous optimization in this lecture

$$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & \text{constraints on } \boldsymbol{x}, \end{aligned}$$

where both objective function f and constraint set are continuous.

I'll switch back and forth between maximization and minimization during lecture (sorry!)

- Applied mathematicians talk about minimization exclusively
- Statisticians talk about maximization due to maximum likelihood estimation

They are fundamentally same problem: minimizing $f(\boldsymbol{x})$ is equivalent to maximizing $-f(\boldsymbol{x})$

Newton's method

Consider maximizing log-likelihood $L(\theta)$, $\theta \in \mathbf{\Theta} \subset \mathbb{R}^p$.

- Newton's method was originally developed for finding roots of nonlinear equations $f(\boldsymbol{x}) = \mathbf{0}$
- Newton's method (aka *Newton-Raphson method*) is considered the gold standard for its fast (quadratic) convergence

$$\frac{\|\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^*\|}{\|\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^*\|^2} \to \text{constant}$$

- Idea: iterative quadratic approximation

- Notations. $\nabla L(\theta)$: gradient vector, $d^2 L(\theta)$: Hessian matrix
- Statistical jargon. $\nabla L(\theta)$: score vector, $-d^2 L(\theta)$: observed information matrix, $\mathrm{E}[-d^2 L(\theta)]$: expected (Fisher) information matrix
- Taylor expansion around the current iterate $\theta^{(t)}$

$$L(\theta) \approx L(\theta^{(t)}) + \nabla L(\theta^{(t)})^T(\theta - \theta^{(t)}) + \frac{1}{2}(\theta - \theta^{(t)})^T d^2 L(\theta^{(t)})(\theta - \theta^{(t)})$$

and then maximize the quadratic approximation

- To maximize the quadratic function, we equate its gradient to zero

$$\nabla L(\theta^{(t)}) + [d^2 L(\theta^{(t)})](\theta - \theta^{(t)}) = \mathbf{0}_p,$$

which suggests the next iterate

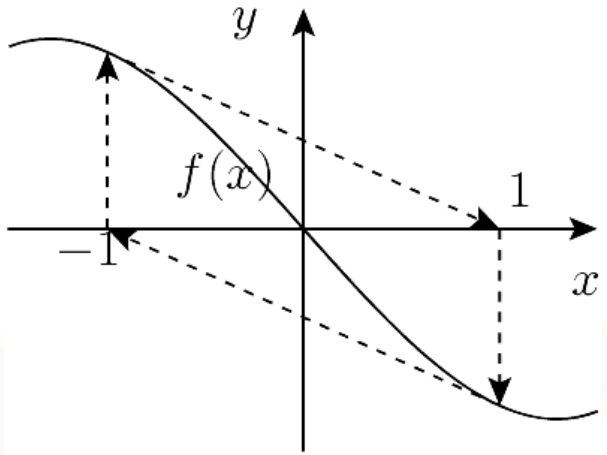
$$\begin{aligned} \theta^{(t+1)} &= \theta^{(t)} - [d^2 L(\theta^{(t)})]^{-1} \nabla L(\theta^{(t)}) \\ &= \theta^{(t)} + [-d^2 L(\theta^{(t)})]^{-1} \nabla L(\theta^{(t)}) \end{aligned}$$

Issues with Newton's method and remedies

- Need to derive, evaluate, and "invert" the observed information matrix. In statistical problems, often evaluating Hessian costs $O(np^2)$ flops and inverting it costs $O(p^3)$ flops

  Remedies:

  1. automatic (analytical) differentiation
  2. numerical differentiation (works for small problems)
  3. quasi-Newton methods (BFGS, ℓ-BFGS probably the most popular black-box nonlinear optimizer)
  4. exploit structures whenever possible to reduce the cost of evaluating and inverting the Hessian



- Stability: Naïve Newton's iterate is not guaranteed to be an ascent algorithm. It's equally happy to head uphill or downhill

  Remedies:
  1. approximate $-d^2 L(\boldsymbol{\theta}^{(t)})$ by a positive definite $\boldsymbol{A}$ (if it's not), and
  2. line search (backtracking)

  By first-order Taylor expansion,

$$\begin{aligned} & L(\boldsymbol{\theta}^{(t)} + s\Delta\boldsymbol{\theta}^{(t)}) - L(\boldsymbol{\theta}^{(t)}) \\ = \; & s\nabla L(\boldsymbol{\theta}^{(t)})^T \Delta\boldsymbol{\theta}^{(t)} + o(s) \\ = \; & s\nabla L(\boldsymbol{\theta}^{(t)})^T \boldsymbol{A}^{-(t)} \nabla L(\boldsymbol{\theta}^{(t)}) + o(s). \end{aligned}$$

  For s sufficiently small, right hand side is strictly positive

In summary, Newton scheme iterates according to

$$\boxed{\theta^{(t+1)} = \theta^{(t)} + s[\boldsymbol{A}^{(t)}]^{-1}\nabla L(\theta^{(t)}) = \theta^{(t)} + s\Delta\theta^{(t)}}$$

where $\boldsymbol{A}^{(t)}$ is a pd approximation of $-d^2L(\theta^{(t)})$ and s is a step length

- Line search strategy: step-halving ($s = 1, 1/2, \ldots$), golden section search, cubic interpolation, Amijo rule, ...
  Newton direction $\Delta\theta^{(t)}$ only need to be calculated once. Cost of line search mainly lies in objective function evaluation
- How to approximate $-d^2 L(\theta)$? More of an art than science. Often requires problem specific analysis.
- Taking $\boldsymbol{A}^{(t)} = \boldsymbol{I}$ leads to the method of steepest ascent, aka gradient ascent

Fisher's scoring

Fisher's scoring method replaces $-d^2 L(\theta)$ by the expected (Fisher) information matrix

$$\boldsymbol{I}(\theta) = \mathrm{E}[-d^2 L(\theta)] = \mathrm{E}[\nabla L(\theta) \nabla L(\theta)^T] \succeq \mathbf{0}_{p \times p},$$

which is psd under exchangeability of expectation and differentiation. Therefore the Fisher's scoring algorithm iterates according to

$$\boxed{\theta^{(t+1)} = \theta^{(t)} + s[\boldsymbol{I}(\theta^{(t)})]^{-1} \nabla L(\theta^{(t)})}$$

Constrained optimization

Main strategies for dealing with constrained optimization, e.g.,

$$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & c_i(\boldsymbol{x}) \geq 0, i = 1, \ldots, m \end{aligned}$$

- Interior point method (barrier method)

$$\text{minimize} \quad f(\boldsymbol{x}) - \mu \sum_{i=1}^{m} \log(c_i(\boldsymbol{x}))$$

  and push μ to 0
- Sequential quadratic programming (SQP)

$$\begin{aligned} \text{minimize} \quad & \text{quadratic approximation of } f \\ \text{subject to} \quad & c_i(\boldsymbol{x}) \geq 0 \end{aligned}$$

- Active set method

Interior point method is the most successful for large problems

Outline

EM algorithm

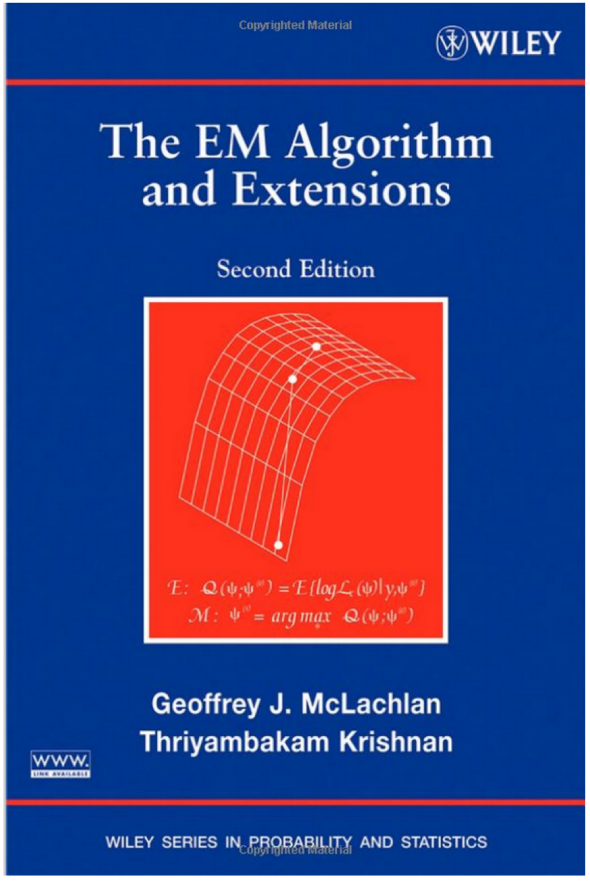
- Which are the most cited statistical papers?

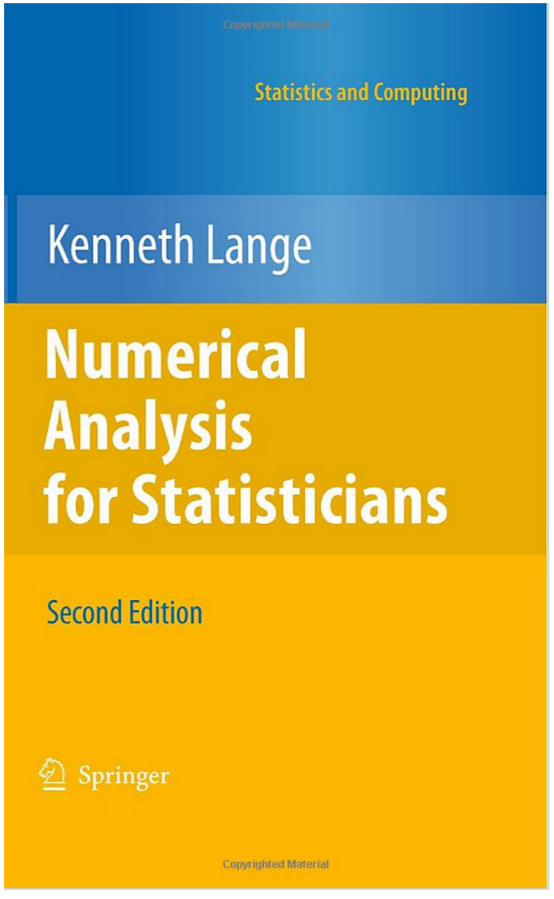
Paper	Citations	Per Year
Kaplan-Meier (Kaplan and Meier, 1958)	46886	808
EM (Dempster et al., 1977)	44050	1129
Cox model (Cox, 1972)	40920	930
Metropolis (Metropolis et al., 1953)	31284	497
FDR (Benjamini and Hochberg, 1995)	30975	1450
Unit root test (Dickey and Fuller, 1979)	18259	493
Lasso (Tibshirani, 1996)	15306	765
bootstrap (Efron, 1979)	12992	351
FFT (Cooley and Tukey, 1965)	11319	222
Gibbs sampler (Gelfand and Smith, 1990)	6531	251

Citation counts from Google Scholar on Feb 17, 2016.

- EM is one of the most influential statistical ideas, finding applications in various branches of science

General reference books on EM





- Notations
  - $\boldsymbol{Y}$: observed data
  - $\boldsymbol{Z}$: missing data
  - $\boldsymbol{X} = (\boldsymbol{Y}, \boldsymbol{Z})$: complete data
- Goal: maximize the log-likelihood of the observed data $\ln g(\boldsymbol{y}|\boldsymbol{\theta})$ (optimization!)
- Idea: choose $\boldsymbol{Z}$ such that MLE for the complete data is trivial
- Let $f(\boldsymbol{x}|\boldsymbol{\theta}) = f(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{\theta})$ be the density of complete data
- Each iteration of EM contains two steps
  - E step: calculate the conditional expectation

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathbf{E}_{\boldsymbol{Z}|\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{\theta}^{(t)}}[\ln f(\boldsymbol{Y}, \boldsymbol{Z}|\boldsymbol{\theta}) \mid \boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{\theta}^{(t)}]$$

  - M step: maximize $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ to generate the next iterate

$$\boldsymbol{\theta}^{(t+1)} = \operatorname{argmax}_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$$

- Fundamental inequality of EM

$$\begin{aligned}
& Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}) - \ln g(\boldsymbol{y}|\boldsymbol{\theta}) \\
= \;& \mathbf{E}[\ln f(\boldsymbol{Y}, \boldsymbol{Z}|\boldsymbol{\theta})|\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{\theta}^{(t)}] - \ln g(\boldsymbol{y}|\boldsymbol{\theta}) \\
= \;& \mathbf{E}\left\{\ln\left[\frac{f(\boldsymbol{Y}, \boldsymbol{Z} \mid \boldsymbol{\theta})}{g(\boldsymbol{Y} \mid \boldsymbol{\theta})}\right] \mid \boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{\theta}^{(t)}\right\} \\
\text{(inform. ineq.)} \quad \leq \;& \mathbf{E}\left\{\ln\left[\frac{f(\boldsymbol{Y}, \boldsymbol{Z} \mid \boldsymbol{\theta}^{(t)})}{g(\boldsymbol{Y} \mid \boldsymbol{\theta}^{(t)})}\right] \mid \boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{\theta}^{(t)}\right\} \\
= \;& Q(\boldsymbol{\theta}^{(t)} \mid \boldsymbol{\theta}^{(t)}) - \ln g(\boldsymbol{y}|\boldsymbol{\theta}^{(t)})
\end{aligned}$$

Rearranging shows

$$\ln g(\boldsymbol{y} \mid \boldsymbol{\theta}) \geq Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}) - Q(\boldsymbol{\theta}^{(t)} \mid \boldsymbol{\theta}^{(t)}) + \ln g(\boldsymbol{y} \mid \boldsymbol{\theta}^{(t)})$$

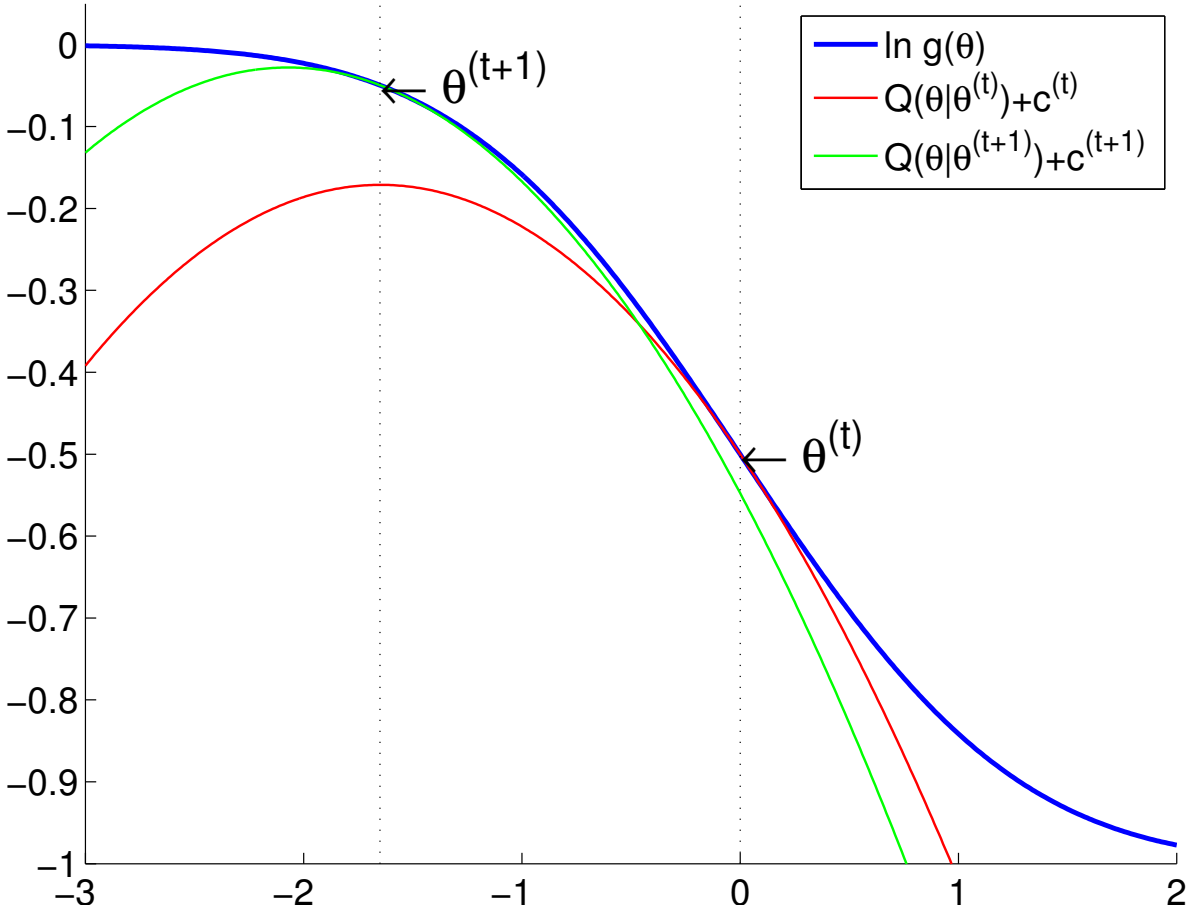
for all $\boldsymbol{\theta}$. Equality is achieved at $\boldsymbol{\theta} = \boldsymbol{\theta}^{(t)}$

- Ascent property of EM

$$\begin{aligned} \ln g(\boldsymbol{y} \mid \boldsymbol{\theta}^{(t+1)}) &\geq Q(\boldsymbol{\theta}^{(t+1)} \mid \boldsymbol{\theta}^{(t)}) - Q(\boldsymbol{\theta}^{(t)} \mid \boldsymbol{\theta}^{(t)}) + \ln g(\boldsymbol{y} \mid \boldsymbol{\theta}^{(t)}) \\ &\geq \ln g(\boldsymbol{y} \mid \boldsymbol{\theta}^{(t)}) \end{aligned}$$

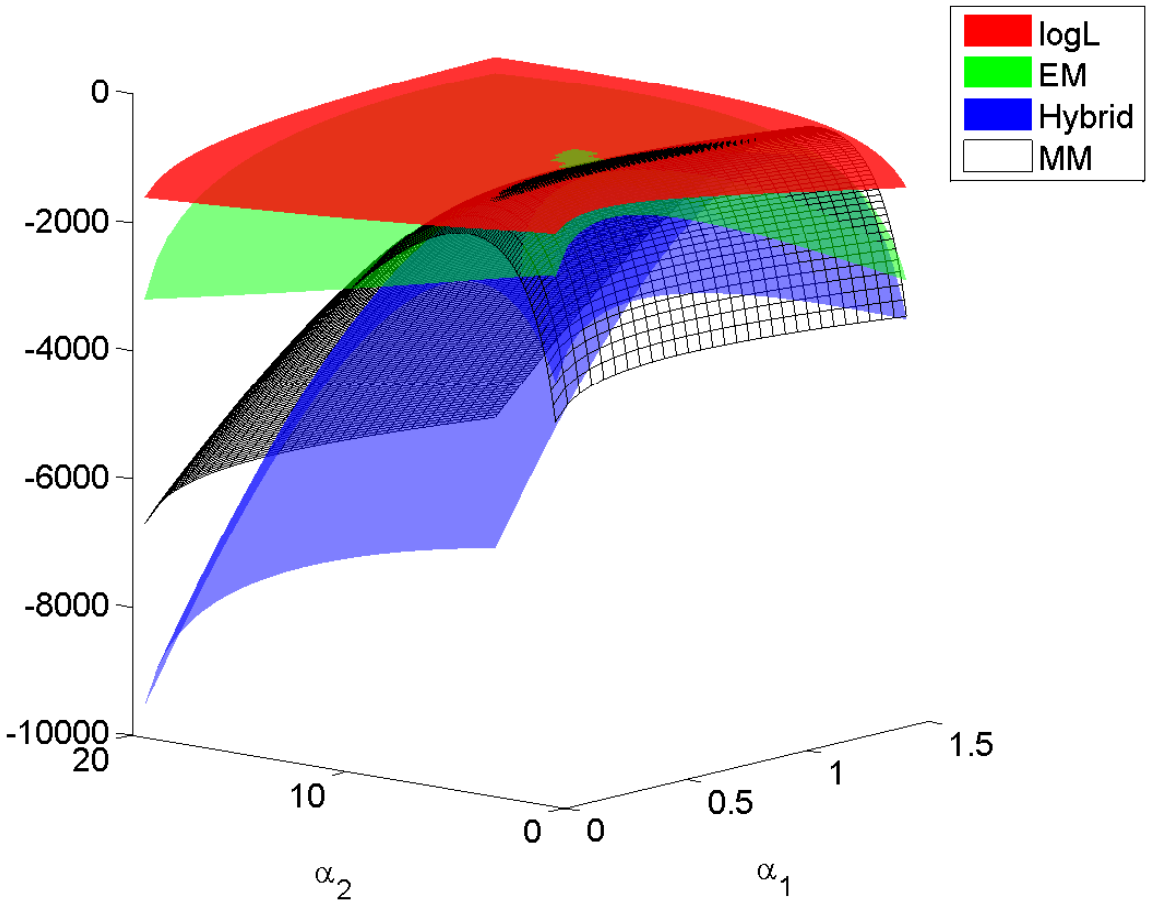
- Obviously we only need $Q(\boldsymbol{\theta}^{(t+1)} \mid \boldsymbol{\theta}^{(t)}) - Q(\boldsymbol{\theta}^{(t)} \mid \boldsymbol{\theta}^{(t)}) \geq 0$ for this ascent property to hold (Generalized EM)

A picture to keep in mind



One more ...

2D function

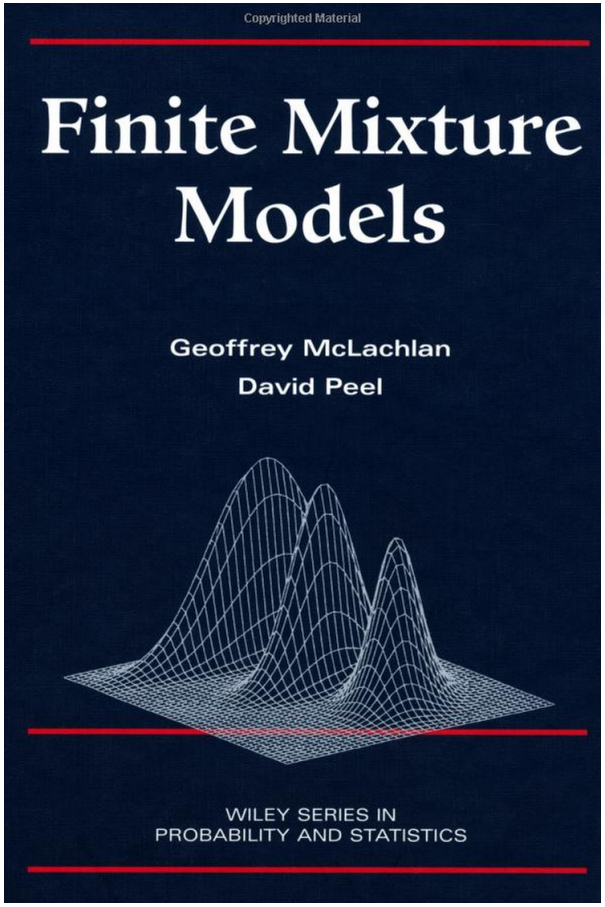


Remarks on EM

- Work magically (at least to applied mathematicians). No need for gradient and Hessian
- Concavity of the observed log-likelihood function (objective function) is not required
- Under regularity conditions, EM converges to a stationary point of $\ln g(\boldsymbol{y}|\boldsymbol{\theta})$. For a non-concave objective function, it can be a local mode or even a saddle point
- Original idea also appeared in the HMM paper (Baum-Welch algorithm) (Baum et al., 1970)
- Tend to separate variables – parallel computing

A canonical EM example – finite mixture model

A canonical example for EM



- Each data point $\boldsymbol{y}_i \in \mathbb{R}^d$ comes from candidate distribution h_j with probability π_j, $\sum_{j=1}^{k} \pi_j = 1$
- Mixture density $h(\boldsymbol{y}) = \sum_{j=1}^{k} \pi_j h_j(\boldsymbol{y} \mid \boldsymbol{\mu}_j, \boldsymbol{\Omega}_j)$, $\boldsymbol{y} \in \mathbb{R}^d$, where

$$h_j(\boldsymbol{y} \mid \boldsymbol{\mu}_j, \boldsymbol{\Omega}_j) = \left(\frac{1}{2\pi}\right)^{d/2} |\det(\boldsymbol{\Omega}_j)|^{-1/2} e^{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu}_j)^{\mathsf{T}} \boldsymbol{\Omega}_j^{-1} (\boldsymbol{y}-\boldsymbol{\mu}_j)}$$

  are $N(\boldsymbol{\mu}_j, \boldsymbol{\Omega}_j)$
- Given data $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$, want to estimate

$$\boldsymbol{\theta} = (\pi_1, \ldots, \pi_k, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k, \boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_k).$$

  Observed (incomplete) data log-likelihood is

$$\ln g(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n | \boldsymbol{\theta}) = \sum_{i=1}^{n} \ln h(\boldsymbol{y}_i) = \sum_{i=1}^{n} \ln \sum_{j=1}^{k} \pi_j h_j(\boldsymbol{y}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Omega}_j)$$

EM for fitting mixture model

- Missing data: let $z_{ij} = I\{\boldsymbol{y}_i \text{ comes from group } j\}$
- Complete data likelihood is

$$f(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{\theta}) = \prod_{i=1}^{n} \prod_{j=1}^{k} [\pi_j h_j(\boldsymbol{y}_i|\boldsymbol{\mu}_j, \boldsymbol{\Omega}_j)]^{z_{ij}}$$

and thus complete log-likelihood is

$$\ln f(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{j=1}^{k} z_{ij} [\ln \pi_j + \ln h_j(\boldsymbol{y}_i|\boldsymbol{\mu}_j, \boldsymbol{\Omega}_j)]$$

E step

- Conditional expectation

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathbf{E}\left\{\sum_{i=1}^{n}\sum_{j=1}^{k} z_{ij}[\ln \pi_j + \ln h_j(\boldsymbol{y}_i|\boldsymbol{\mu}_j, \boldsymbol{\Omega}_j) \mid \boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{\pi}^{(t)}, \boldsymbol{\mu}_1^{(t)}, \ldots, \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Omega}_1^{(t)}, \ldots, \boldsymbol{\Omega}_k^{(t)}]\right\}$$

- By Bayes rule, we have

$$\begin{aligned} w_{ij}^{(t)} &:= \mathbf{E}[z_{ij} \mid \boldsymbol{y}, \boldsymbol{\pi}^{(t)}, \boldsymbol{\mu}_1^{(t)}, \ldots, \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Omega}_1^{(t)}, \ldots, \boldsymbol{\Omega}_k^{(t)}] \\ &= \frac{\pi_j^{(t)} h_j(\boldsymbol{y}_i|\boldsymbol{\mu}_j^{(t)}, \boldsymbol{\Omega}_j^{(t)})}{\sum_{j'=1}^{k} \pi_{j'}^{(t)} h_{j'}(\boldsymbol{y}_i|\boldsymbol{\mu}_{j'}^{(t)}, \boldsymbol{\Omega}_{j'}^{(t)})} \end{aligned}$$

- Q function becomes

$$\sum_{i=1}^{n}\sum_{j=1}^{k} w_{ij}^{(t)} \ln \pi_j + \sum_{i=1}^{n}\sum_{j=1}^{k} w_{ij}^{(t)} \left[-\frac{1}{2}\ln \det \boldsymbol{\Omega}_j - \frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{\mu}_j)^{\mathsf{T}} \boldsymbol{\Omega}_j^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu}_j)\right]$$

M step

- Maximizer of the Q function gives the next iterate

$$\begin{aligned}
\pi_j^{(t+1)} &= \frac{\sum_{i=1}^n w_{ij}^{(t)}}{n} \\
\boldsymbol{\mu}_j^{(t+1)} &= \frac{\sum_{i=1}^n w_{ij}^{(t)} \boldsymbol{y}_i}{\sum_{i=1}^n w_{ij}^{(t)}} \\
\boldsymbol{\Omega}_j^{(t+1)} &= \frac{\sum_{i=1}^n w_{ij}^{(t)} (\boldsymbol{y}_i - \boldsymbol{\mu}_j^{(t+1)})(\boldsymbol{y}_i - \boldsymbol{\mu}_j^{(t+1)})^{\mathsf{T}}}{\sum_i w_{ij}^{(t)}}
\end{aligned}$$

- Why? Multinomial MLE and weighted multivariate normal MLE

Remarks

- No gradient and Hessian calculation at all
- Simplex constraint on π and psd constraint on $\mathbf{\Omega}_j$ are satisfied by EM iterates
- The "membership variables" $w_{ij}^{(t)}$ can be computed in parallel. Suchard et al. (2010) observe $>$ 100 speed up when fitting massive mixture model (huge n) on GPU
- It is a common theme EM/MM type algorithms are amenable to massively parallel computing
- Similar EM applies to other component distributions, as far as its MLE is easy to compute

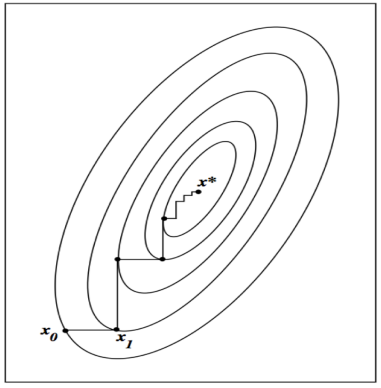
Other classical EM algorithms

- Hidden Markov model (Baum-Welch, or forward-backward algorithm)
- Factor analysis
- Variance component model
- Hyper-parameter estimation in empirical Bayes procedure. MAP (maximum *a posteriori* estimation)
- Missing data
- Grouped/censorized/truncated model
- Robust regression (t-distribution and t-regression)
- ...

ECM

Expectation Conditional Maximization (Meng and Rubin, 1993)

- In some problems the M step is difficult (no analytic solution)
- Conditional maximization is easy (block ascent)
  - partition parameter vector into blocks $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_B)$
  - alternate update $\boldsymbol{\theta}_b$, $b = 1, \ldots, B$



- Ascent property still holds. Why?
- ECM may converge slower than EM (more iterations) but the total computer time may be shorter due to ease of the CM step

ECME

ECM Either (Liu and Rubin, 1994)

- Each CM step maximizes either the Q function or the original incomplete observed log-likelihood
- Ascent property still holds. Why?
- Faster convergence than ECM

AECM

Alternating ECM (Meng and van Dyk, 1997)

- The specification of the complete-data is allowed to be different on each CM-step
- Ascent property still holds. Why?

PX-EM and efficient data augmentation

- Parameter-EXpanded EM (Liu et al., 1998; Liu and Wu, 1999)
- Efficient data augmentation (Meng and van Dyk, 1997)
- Idea: Speed up the convergence of EM algorithm by efficiently augmenting the observed data (introducing a working parameter in the specification of complete data)

Example: *t* distribution

- $\boldsymbol{W} \in \mathbb{R}^p$ is a multivariate t-distribution $t_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu)$ if $\boldsymbol{W} \sim \text{Normal}(\boldsymbol{\mu}, \boldsymbol{\Sigma}/u)$ and $u \sim \text{Gamma}(\frac{\nu}{2}, \frac{\nu}{2})$
- Widely used for robust modeling
- Gamma(α, β) has density

$$f(u \mid \alpha, \beta) = \frac{\beta^\alpha u^{\alpha-1}}{\Gamma(\alpha)} e^{-\beta\mu}, \quad u \geq 0,$$

with mean $\mathbf{E}(U) = \alpha/\beta$ and $\mathbf{E}(\ln U) = \psi(\alpha) - \ln \beta$

- Density of $\boldsymbol{W}$ is

$$f_p(\boldsymbol{w} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) = \frac{\Gamma(\frac{\nu+p}{2})|\boldsymbol{\Sigma}|^{-1/2}}{(\pi\nu)^{p/2}\Gamma(\nu/2)[1 + \delta(\boldsymbol{w}, \boldsymbol{\mu}; \boldsymbol{\Sigma})/\nu]^{(\nu+p)/2}},$$

where $\delta(\boldsymbol{w}, \boldsymbol{\mu}; \boldsymbol{\Sigma}) = (\boldsymbol{w} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{w} - \boldsymbol{\mu})$ is the Mahalanobis squared distance between $\boldsymbol{w}$ and $\boldsymbol{\mu}$

- Given iid data $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n$, the log-likelihood is

$$\begin{aligned} L(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) &= -\frac{np}{2} \ln(\pi\nu) + n\left[\ln \Gamma\left(\frac{\nu+p}{2}\right) - \ln \Gamma\left(\frac{\nu}{2}\right)\right] \\ &\quad -\frac{n}{2} \ln |\boldsymbol{\Sigma}| + \frac{n}{2}(\nu+p) \ln \nu \\ &\quad -\frac{1}{2}(\nu+p) \sum_{j=1}^{n} \ln[\nu + \delta(\boldsymbol{w}_j, \boldsymbol{\mu}; \boldsymbol{\Sigma})] \end{aligned}$$

- How to compute MLE $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}, \hat{\nu})$?

Multivariate t MLE – EM

- $\boldsymbol{W}_j | u_j$ independent $N(\boldsymbol{\mu}, \boldsymbol{\Sigma}/u_j)$ and U_j iid gamma$(\frac{\nu}{2}, \frac{\nu}{2})$
- Missing data: $\boldsymbol{z} = (u_1, \ldots, u_n)^T$
- Log-likelihood of complete data is

$$\begin{aligned} L_c(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) \quad = \quad & -\frac{1}{2} np \ln(2\pi) - \frac{n}{2} \ln |\boldsymbol{\Sigma}| \\ & -\frac{1}{2} \sum_{j=1}^{n} u_j (\boldsymbol{w}_j - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{w}_j - \boldsymbol{\mu}) \\ & -n \ln \Gamma\left(\frac{\nu}{2}\right) + \frac{n\nu}{2} \ln\left(\frac{\nu}{2}\right) \\ & +\frac{\nu}{2} \sum_{j=1}^{n} (\ln u_j - u_j) - \sum_{j=1}^{n} \ln u_j \end{aligned}$$

E step

- Since gamma is conjugate prior for U, conditional distribution of U given $\boldsymbol{W} = \boldsymbol{w}$ is gamma$((\nu + p)/2, (\nu + \delta(\boldsymbol{w}, \boldsymbol{\mu}; \boldsymbol{\Sigma}))/2)$. Thus

$$\begin{aligned}
\mathbf{E}(U_j \mid \boldsymbol{w}_j, \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)}, \nu^{(t)}) &= \frac{\boldsymbol{\nu}^{(t)} + p}{\nu^{(t)} + \delta(\boldsymbol{w}_j, \boldsymbol{\mu}^{(t)}; \boldsymbol{\Sigma}^{(t)})} =: u_j^{(t)} \\
\mathbf{E}(\ln U_j \mid \boldsymbol{w}_j, \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)}, \nu^{(t)}) &= \ln u_j^{(t)} + \left[\psi\left(\frac{\nu^{(t)} + p}{2} \right) - \ln\left(\frac{\nu^{(t)} + p}{2} \right) \right]
\end{aligned}$$

- Overall Q function (up to an additive constant) takes the form

$$\begin{aligned}
&-\frac{n}{2} \ln |\boldsymbol{\Sigma}| - \frac{1}{2} \sum_{j=1}^{n} u_j^{(t)} (\boldsymbol{w}_j - \boldsymbol{\mu})^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{w}_j - \boldsymbol{\mu}) \\
&-n \ln \Gamma\left(\frac{\nu}{2} \right) + \frac{n\nu}{2} \ln\left(\frac{\nu}{2} \right) \\
&+\frac{n\nu}{2} \left[\frac{1}{n} \sum_{j=1}^{n} (\ln u_j^{(t)} - u_j^{(t)}) + \psi\left(\frac{\nu^{(t)} + p}{2} \right) - \ln\left(\frac{\nu^{(t)} + p}{2} \right) \right]
\end{aligned}$$

M step

- Maximization over $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is simply a weighted multivariate normal problem

$$\begin{aligned}
\boldsymbol{\mu}^{(t+1)} &= \frac{\sum_{j=1}^{n} u_j^{(t)} \boldsymbol{w}_j}{\sum_{j=1}^{n} u_j^{(t)}} \\
\boldsymbol{\Sigma}^{(t+1)} &= \frac{1}{n} \sum_{j=1}^{n} u_j^{(t)} (\boldsymbol{w}_j - \boldsymbol{\mu}^{(t+1)})(\boldsymbol{w}_j - \boldsymbol{\mu}^{(t+1)})^{\mathsf{T}}
\end{aligned}$$

  Notice down-weighting of outliers is obvious in the update

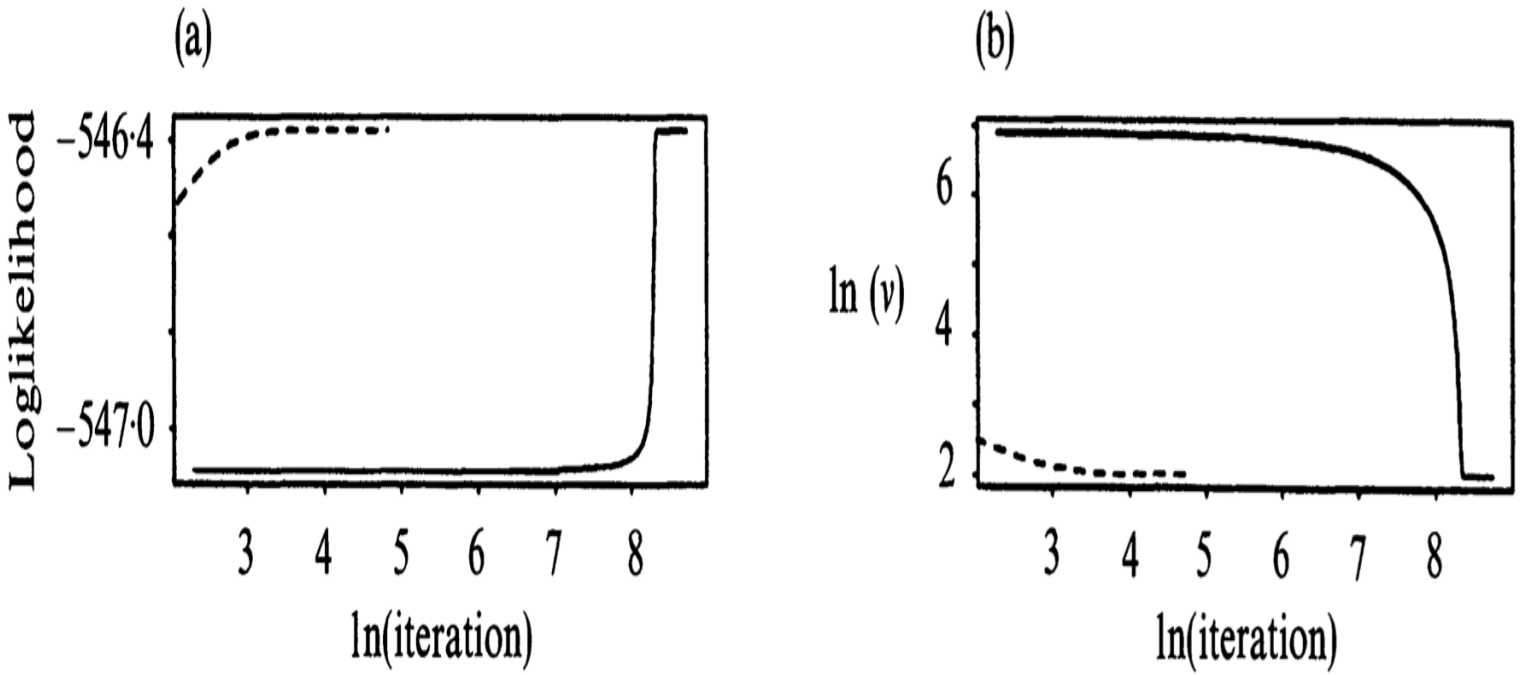
- Maximization over ν is a univariate problem – equating derivative to 0 and find the root

Multivariate t MLE – ECM and ECME

Partition parameter $(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu)$ into two blocks $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and ν

- ECM = EM for this example. Why?
- ECME: In the second CM step, maximize over ν in terms of the original log-likelihood function instead of the Q function. They have similar difficulty since both are univariate optimization problems!

An example from (Liu and Rubin, 1994). $n = 79$, $p = 2$, with missing entries. EM=ECM: solid line. ECME: dashed line.



Multivariate t MLE – efficient data augmentation

Assume ν known

- Write $\boldsymbol{W}_j = \boldsymbol{\mu} + \boldsymbol{C}_j / U_j^{1/2}$, where $\boldsymbol{C}_j$ is $N(\boldsymbol{0}, \boldsymbol{\Sigma})$ independent of U_j which is Gamma$(\frac{\nu}{2}, \frac{\nu}{2})$
- $\boldsymbol{W}_j = \boldsymbol{\mu} + |\boldsymbol{\Sigma}|^{-a/2} \boldsymbol{C}_j / U_j^{1/2}(a)$, where $U_j(a) = |\boldsymbol{\Sigma}|^{-a} U_j$
- Then the complete data is $(\boldsymbol{w}, \boldsymbol{z}(a))$, a the working parameter
- $a = 0$ corresponds to the vanilla EM
- Meng and van Dyk (1997) recommend using $a_{\text{opt}} = 1/(\nu + p)$ to maximize the convergence rate
- Exercise: work out the EM update for this special case
- The only change to the vanilla EM is to replace the denominator n in the update of $\boldsymbol{\Sigma}$ by $\sum_{j=1}^{n} u_j^{(t)}$
- PX-EM (Liu et al., 1998) leads to the same update

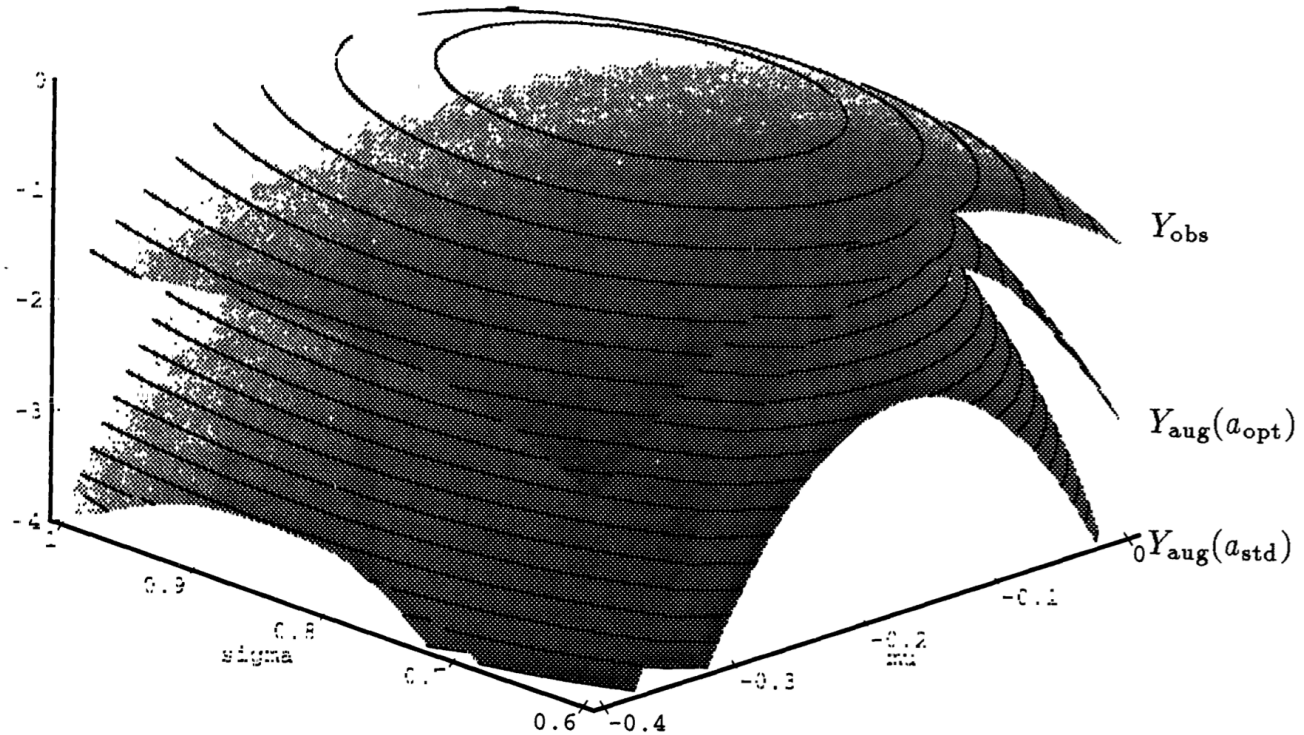


Fig. 5. Comparing the log-likelihoods: the plot shows $L(\theta|Y_{obs})$, as well as $E[L(\theta|Y_{aug})|Y_{obs}, \theta^*]$ for both the standard and the optimal augmentations (each adjusted by their maximum value for comparison); notice that the optimal augmentation results in a flatter log-likelihood that better approximates $L(\theta|Y_{obs})$

Assume ν unknown

- Version 1: $a = a_{\text{opt}}$ in both updating of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and ν
- Version 2: $a = a_{\text{opt}}$ for updating $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and taking the observed data as complete data for updating ν

Conclusion in (Meng and van Dyk, 1997): Version 1 is $8 \sim 12$ faster than EM=ECM or ECME. Version 2 is only slightly more efficient than Version 1

MCEM

Monte Carlo EM (Wei and Tanner, 1990)

- Hard to calculate Q function? Simulate!

$$Q(\theta \mid \theta^{(t)}) \approx \frac{1}{m} \sum_{j=1}^{m} \ln f(\boldsymbol{y}, \boldsymbol{z}_j \mid \theta),$$

  where $\boldsymbol{z}_j$ are iid from conditional distribution of missing data given $\boldsymbol{y}$ and previous iterate $\theta^{(t)}$
- Ascent property may be lost due to Monte Carlo errors
- Example: capture-recapture model (Robert and Casella, 2004, p184), generalized linear mixed model (Booth and Hobert, 1999)

SEM and SAEM

Stochastic EM (SEM) (Celeux and Diebolt, 1985)

- Same as MCEM with $m = 1$. A single draw of missing data $\boldsymbol{z}$ from the conditional distribution
- $\theta^{(t)}$ forms a Markov chain which converges to a stationary distribution. No definite relation between this stationary distribution and the MLE
- In some specific cases, it can be shown that the stationary distribution concentrates around the MLE with a variance inversely proportional to the sample size
- Advantage: can escape the attraction to inferior mode/saddle point in some mixture model problems

Simulated Annealing EM (SAEM) (Celeux and Diebolt, 1989)

- Increase m with the iterations, ending up with an EM algorithm

DA

Data Augmentation (DA) algorithm (Tanner and Wong, 1987)

- Aim for sampling from $p(\theta|\boldsymbol{y})$ instead of maximization
- Idea: incomplete data posterior density is complicated, but the complete-data posterior density is relatively easy to sample
- Data augmentation algorithm
  - draw $\boldsymbol{z}^{(t+1)}$ conditional on $(\boldsymbol{\theta}^{(t)}, \boldsymbol{y})$
  - draw $\boldsymbol{\theta}^{(t+1)}$ conditional on $(\boldsymbol{z}^{(t+1)}, \boldsymbol{y})$
- A special case of Gibbs sampler
- $\boldsymbol{\theta}^{(t)}$ converges to the distribution $p(\theta|\boldsymbol{y})$ under general conditions
- Ergodic mean converges to the posterior mean $\mathbf{E}(\theta|\boldsymbol{y})$, which may perform better than MLE in finite sample

EM as a maximization-maximization procedure

Neal and Hinton (1999)

- Consider the objective function

$$F(\boldsymbol{\theta}, q) = \mathbf{E}_q[\ln f(\boldsymbol{Z}, \boldsymbol{y} \mid \boldsymbol{\theta})] + H(q)$$

over $\boldsymbol{\Theta} \times \mathcal{Q}$, where $\mathcal{Q}$ is the set of *all* conditional pdfs of the missing data $\{q(\boldsymbol{z}) = p(\boldsymbol{z} \mid \boldsymbol{y}, \boldsymbol{\theta}), \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ and $H(q) = -\mathbf{E}_q \ln q$ is the entropy of q

- EM is essentially performing coordinate ascent for maximizing F
  - E step: At current iterate $\boldsymbol{\theta}^{(t)}$,

$$F(\boldsymbol{\theta}^{(t)}, q) = \mathbf{E}_q[\ln p(\boldsymbol{Z} \mid \boldsymbol{y}, \boldsymbol{\theta}^{(t)})] - \mathbf{E}_q \ln q + \ln g(\boldsymbol{y} \mid \boldsymbol{\theta}^{(t)}).$$

The maximizing conditional pdf is given by $q = \ln f(\boldsymbol{Z} \mid \boldsymbol{y}, \boldsymbol{\theta}^{(t)})$

  - M step: Substitute $q = \ln f(\boldsymbol{Z} \mid \boldsymbol{y}, \boldsymbol{\theta}^{(t)})$ into F and maximize over θ

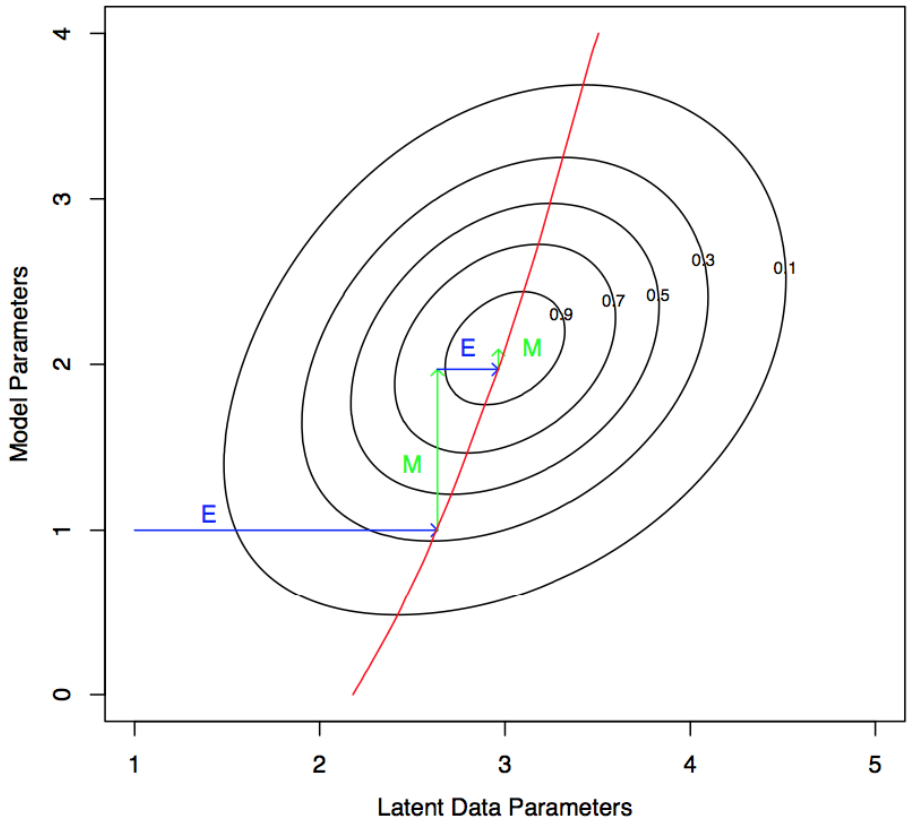


FIGURE 8.7. *Maximization–maximization view of the EM algorithm. Shown are the contours of the (augmented) observed data log-likelihood* $F(\theta', \tilde{P})$. *The E step is equivalent to maximizing the log-likelihood over the parameters of the latent data distribution. The M step maximizes it over the parameters of the log-likelihood. The red curve corresponds to the observed data log-likelihood, a* profile *obtained by maximizing* $F(\theta', \tilde{P})$ *for each value of* θ'.

Hastie et al. (2009, Section 8.5.3)

Incremental EM

- Assume iid observations. Then the objective function is

$$F(\theta, q) = \sum_i \left[\mathbf{E}_{q_i} \ln p(\boldsymbol{Z}_i, \boldsymbol{y}_i \mid \theta) + H(q_i)\right],$$

  where we search for q under factored form $q(\boldsymbol{z}) = \prod_i q_i(\boldsymbol{z})$
- Maximizing F over q is equivalent to maximizing the contribution of each data with respect to q_i
- Update θ by visiting data items sequentially rather than from a global E step
- Finite mixture example: for each data point i
  - evaluate membership variables w_{ij}, $j = 1, \ldots, k$
  - update parameter values $(\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\mu}_j^{(t+1)}, \boldsymbol{\Sigma}_j^{(t+1)})$ (only need to keep sufficient statistics!)

Faster convergence than vanilla EM in some examples

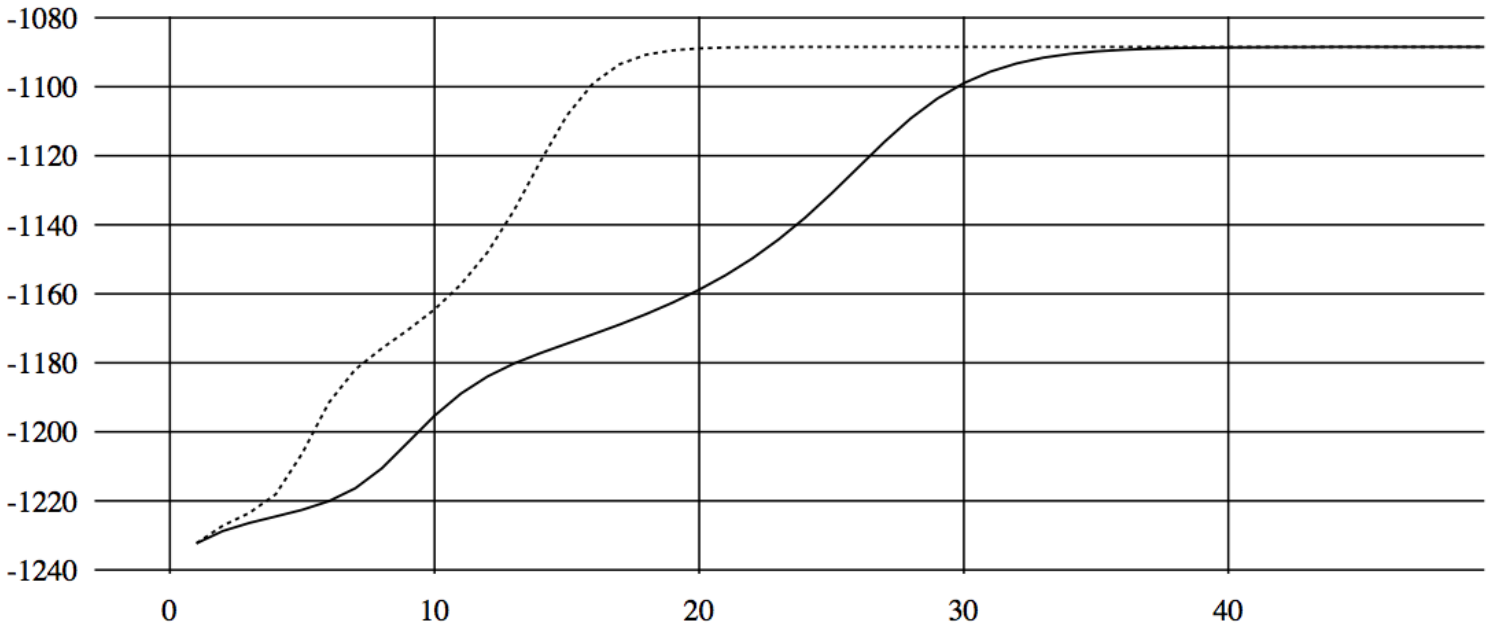
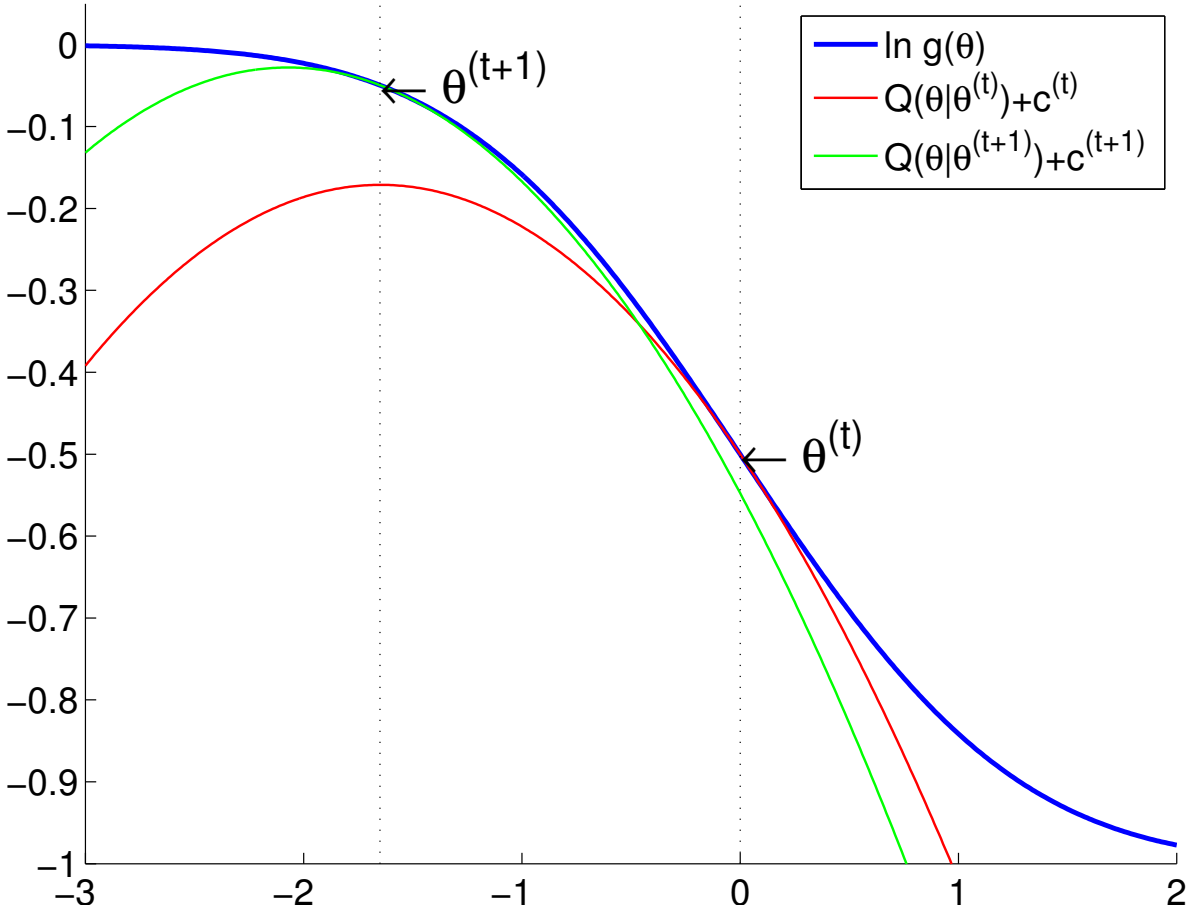


Figure 1. Comparison of convergence rates for the standard EM algorithm (solid line) and the incremental algorithm (dotted line). The log likelihood is shown on the vertical axis, the number of passes of the algorithm on the horizontal axis.

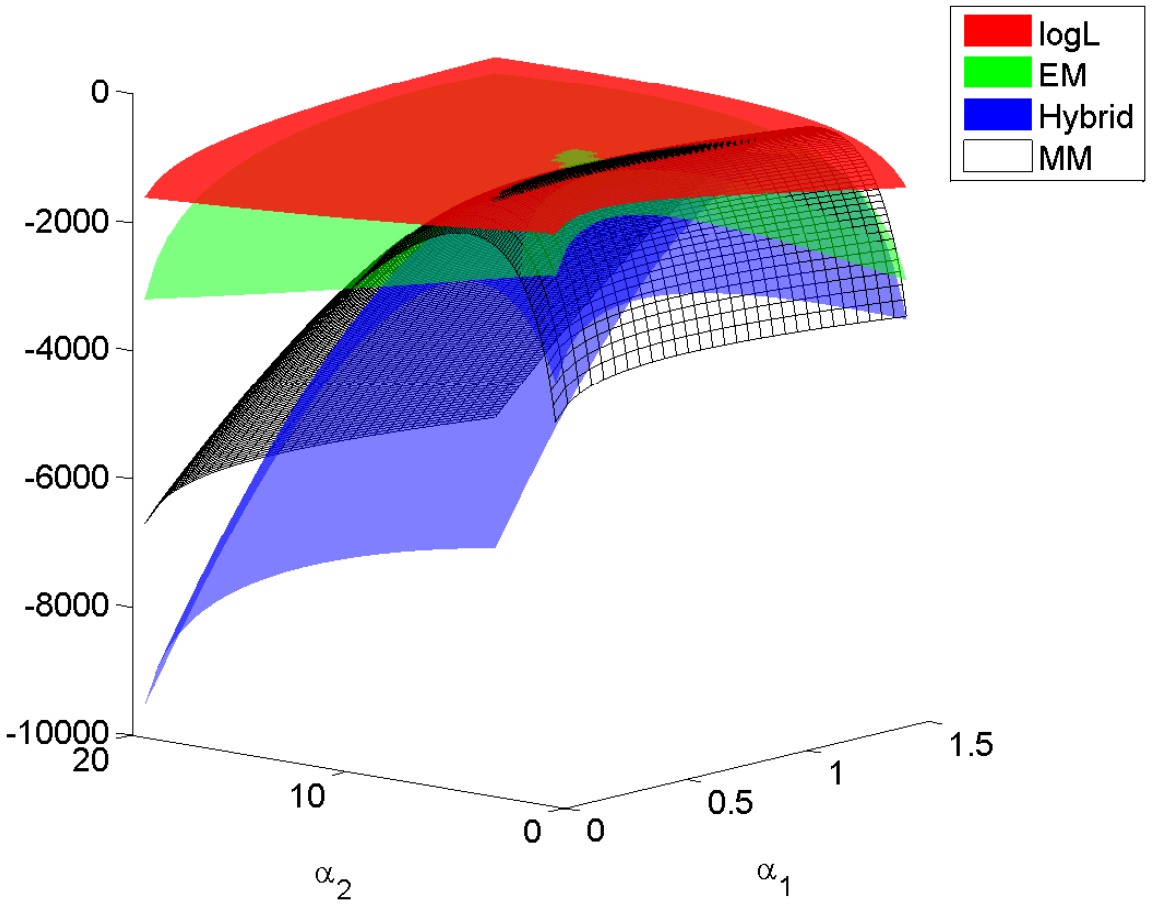
Outline

Our friend ...



Our other friend ...

2D function



EM as a minorization-maximization (MM) algorithm

Refer to the picture in previous slide

- The Q function constitutes a minorizing function of the objective function up to an additive constant

$$\begin{aligned} L(\theta) &\geq Q(\theta|\theta^{(t)}) + c^{(t)} \quad \text{for all } \theta \\ L(\theta^{(t)}) &= Q(\theta^{(t)}|\theta^{(t)}) + c^{(t)} \end{aligned}$$

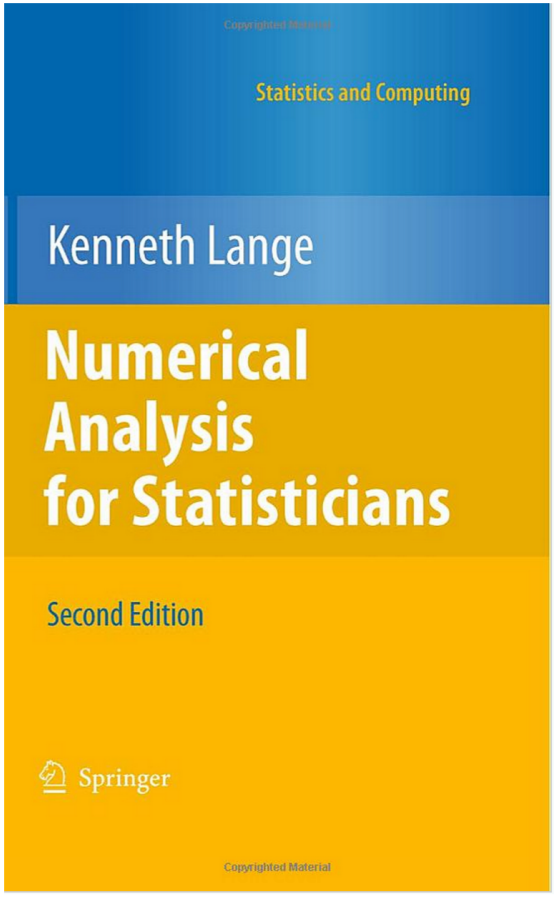
- Maximizing the Q function generates an ascent iterate $\theta^{(t+1)}$

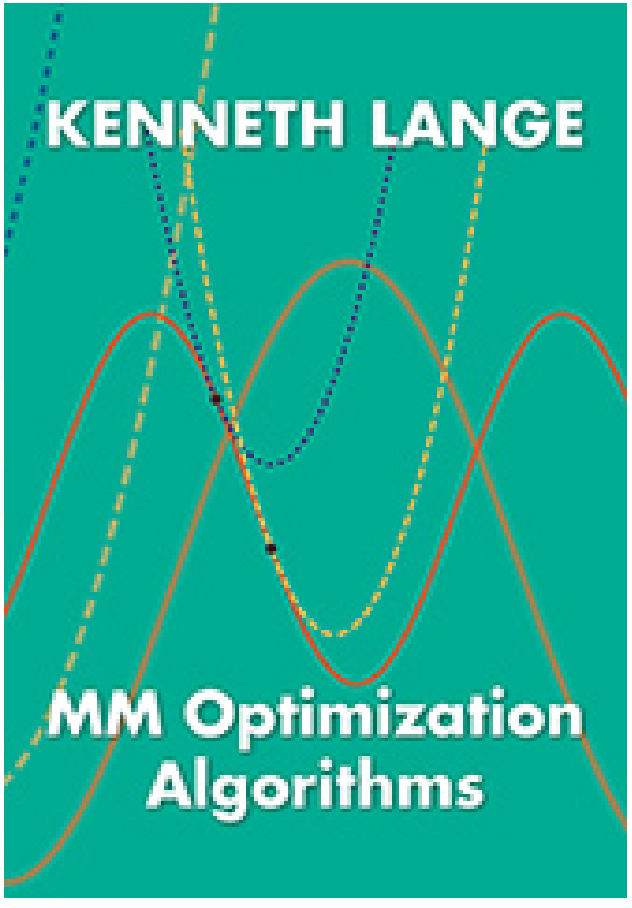
Questions:

- Is there any other way to produce such surrogate function?
- Can we flip the picture and apply same principle to minimization problem?

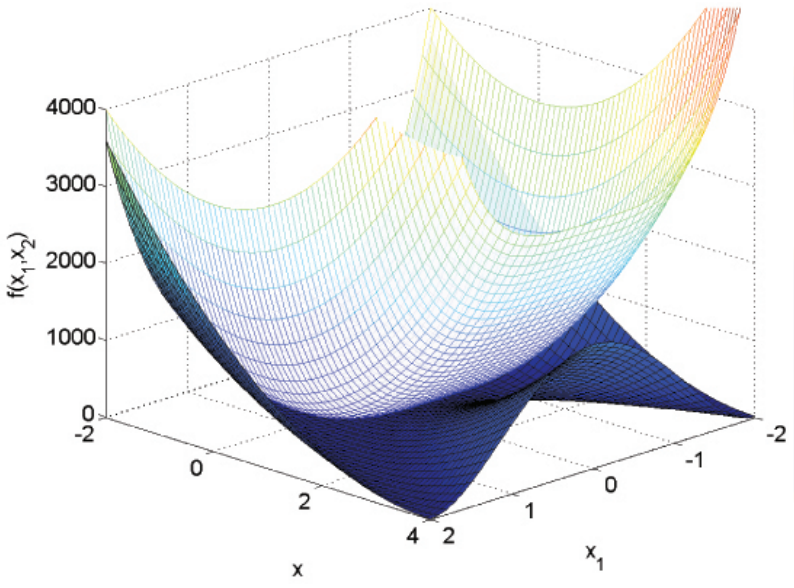
This generalization leads to a powerful tool – MM principle (Lange et al., 2000)

General reference book on MM





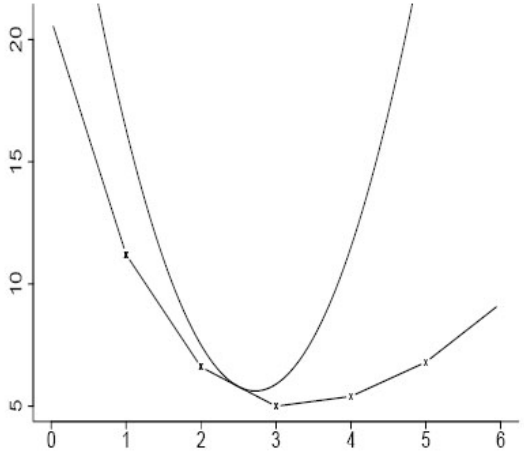
A powerful tool - MM algorithm



- A prescription for constructing optimization algorithms
- **M**ajorize/**M**inimize or **M**inorize/**M**aximize
- Creating a surrogate function that minorizes/majorizes the objective function
- Optimizing the surrogate function drives the objective function uphill or downhill as needed

Famous special cases in statistics: EM algorithm (Dempster et al., 1977), multidimensional scaling (de Leeuw and Heiser, 1977)

Majorization and definition of the algorithm



- A function $g(\theta|\theta^n)$ is said to majorize the function $f(\theta)$ at θ^n provided

$$f(\theta^n) = g(\theta^n|\theta^n)$$
$$f(\theta) \leq g(\theta|\theta^n) \quad \text{for all } \theta.$$

- So we choose a "nice/separable" majorizing function $g(\theta|\theta^n)$ and minimize it. This produces the next point θ^{n+1} in the algorithm.
- The descent property follows from

$$f(\theta^{n+1}) \leq g(\theta^{n+1}|\theta^n) \leq g(\theta^n|\theta^n) = f(\theta^n).$$

This makes MM algorithms very stable.

Rationale for the MM principle

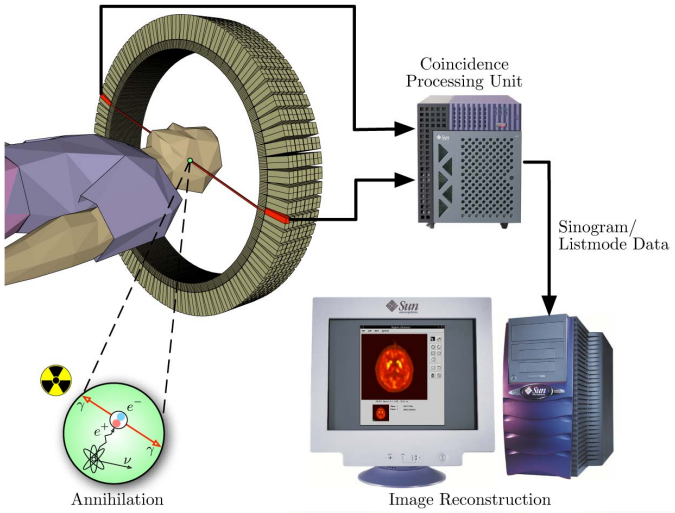
- Free the derivation from missing data structure
- Avoid matrix inversion
- Linearize an optimization problem
- Deal gracefully with certain equality and inequality constraints
- Turn a non-differentiable problem into a smooth problem
- Separate the parameters of a problem (perfect for massive, fine-scale parallelization)

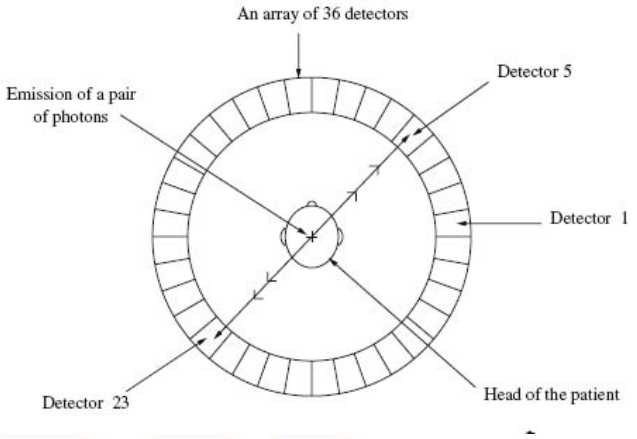
Generic methods of majorization and minorization

- Jensen's inequality – EM algorithms
- The Cauchy-Schwartz inequality - multidimensional scaling
- Supporting hyperplane property of a convex function
- Arithmetic-geometric mean inequality
- Quadratic upper bound principle - Böhning and Lindsay
- ...

A great source of inequalities: *The Cauchy-Schwarz Master Class* by Michael Steele.

Positron Emission Tomography (PET)





- Data: tube readings $\boldsymbol{y} = (y_1, \ldots, y_d)$
- Estimate: photon emission intensities (pixels) $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)$
- Poisson Model: $Y_i \sim \text{Poisson}(\sum_{j=1}^{p} c_{ij}\lambda_j)$ where c_{ij} = cond. prob. that a photon emitted by j-th pixel is detected by i-th tube

- Log-likelihood

$$L(\boldsymbol{\lambda}|\boldsymbol{y}) = \sum_i \left[y_i \ln \left(\sum_j c_{ij}\lambda_j \right) - \sum_j c_{ij}\lambda_j \right] + \text{const.}$$

  Essentially a Poisson regression with constraint $\lambda_j \geq 0$
- Regularized log-likelihood for smoother image:

$$L(\boldsymbol{\lambda}|\boldsymbol{y}) - \frac{\mu}{2} \sum_{\{j,k\} \in \mathcal{N}} (\lambda_j - \lambda_k)^2$$

- EM algorithm does not apply now: no likelihood model for the regularization term

Minorization step

- By concavity of the $\ln s$ function

$$\ln\left(\sum_j c_{ij}\lambda_j\right) \geq \sum_j \frac{c_{ij}\lambda_j^{(t)}}{\sum_j c_{ij'}\lambda_{j'}^{(t)}} \ln\left(\frac{\sum_{j'} c_{ij'}\lambda_{j'}^{(t)}}{c_{ij}\lambda_j^{(t)}} c_{ij}\lambda_j\right) = \sum_j \frac{c_{ij}\lambda_j^{(t)}}{\sum_j c_{ij'}\lambda_{j'}^{(t)}} \ln\lambda_j + c^{(t)}$$

- By concavity of the $-s^2$ function

$$-(\lambda_j - \lambda_k)^2 \geq -\frac{1}{2}(2\lambda_j - \lambda_j^{(t)} - \lambda_k^{(t)})^2 - \frac{1}{2}(2\lambda_k - \lambda_j^{(t)} - \lambda_k^{(t)})^2$$

- Combining minorizing terms gives an overall surrogate function

$$\begin{aligned} g(\boldsymbol{\lambda}|\boldsymbol{\lambda}^{(t)}) &= \sum_i y_i \sum_j \frac{c_{ij}\lambda_j^{(t)}}{\sum_{j'} c_{ij'}\lambda_{j'}^{(t)}} \ln\lambda_j - \sum_i \sum_j c_{ij}\lambda_j \\ &\quad -\frac{\mu}{4} \sum_{\{j,k\}\in\mathcal{N}} [(2\lambda_j - \lambda_j^{(t)} - \lambda_k^{(t)})^2 + (2\lambda_k - \lambda_j^{(t)} - \lambda_k^{(t)})^2] \end{aligned}$$

Maximization step

- $g(\boldsymbol{\lambda}|\boldsymbol{\lambda}^{(t)})$ is trivial to maximize because λ_j are separated!
- Solving for the root of

$$\begin{aligned} & \frac{\partial}{\partial \lambda_j} g(\boldsymbol{\lambda}|\boldsymbol{\lambda}^{(t)}) \\ = & \sum_i y_i \frac{c_{ij}\lambda_j^{(t)}}{\sum_{j'} c_{ij}\lambda_{j'}^{(t)}} \lambda_j^{-1} - \sum_i c_{ij} - \mu \sum_{k \in \mathcal{N}_j} (2\lambda_j - \lambda_j^{(t)} - \lambda_k^{(t)}) \\ = & \ 0 \end{aligned}$$

gives $\lambda_j^{(t+1)}$

- MM algorithm for PET:

  Initialize: $\lambda_j^{(0)} = 1$
  repeat
    $z_{ij}^{(t)} = (y_i c_{ij} \lambda_j^{(t)}) / (\sum_k c_{ik} \lambda_k^{(t)})$
    for $j = 1$ to p **do**
      $a = -2\mu|\mathcal{N}_j|$, $b = \mu(|\mathcal{N}_j|\lambda_j^{(t)} + \sum_{k \in \mathcal{N}_j} \lambda_k^{(t)}) - 1$, $c = \sum_i z_{ij}^{(t)}$
      $\lambda_j^{(t+1)} = (-b - \sqrt{b^2 - 4ac})/(2a)$
    end for
  until convergence occurs

- Parameter constraints $\lambda_j \geq 0$ are satisfied when start from positive initial values
- The loop for updating pixels can be carried out independently – massive parallelism

Multivariate t MLE (revisited)

Problem

- Given iid data $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n$, the log-likelihood is

$$\begin{aligned} f(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) \quad = \quad & -\frac{np}{2} \ln(\pi\nu) + n[\ln \Gamma(\frac{\nu+p}{2}) - \ln \Gamma(\frac{\nu}{2})] \\ & -\frac{n}{2} \ln |\boldsymbol{\Sigma}| + \frac{n}{2}(\nu+p) \ln \nu \\ & -\frac{1}{2}(\nu+p) \sum_{j=1}^{n} \ln[\nu + \delta(\boldsymbol{w}_j, \boldsymbol{\mu}; \boldsymbol{\Sigma})], \end{aligned}$$

  where $\delta(\boldsymbol{w}_j, \boldsymbol{\mu}; \boldsymbol{\Sigma}) = (\boldsymbol{w}_j - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{w}_j - \boldsymbol{\mu})$

- Want to find MLE $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}, \hat{\nu})$

Minorization step

- Apply the supporting hyperplane inequality

$$\begin{aligned} & -\ln[\nu + \delta(\boldsymbol{w}_j, \boldsymbol{\mu}; \boldsymbol{\Sigma})] \\ \geq & -\frac{\nu + \delta(\boldsymbol{w}_j, \boldsymbol{\mu}; \boldsymbol{\Sigma})}{\nu^{(t)} + \delta(\boldsymbol{w}_j, \boldsymbol{\mu}^{(t)}; \boldsymbol{\Sigma}^{(t)})} + 1 - \ln[\nu^{(t)} + \delta(\boldsymbol{w}_j, \boldsymbol{\mu}^{(t)}; \boldsymbol{\Sigma}^{(t)})] \end{aligned}$$

- Summing over observations gives an overall minorizing function

$$\begin{aligned} & g(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu | \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)}, \nu^{(t)}) \\ = & -\frac{np}{2}\ln(\pi\nu) + n[\ln\Gamma(\frac{\nu+p}{2}) - \ln\Gamma(\frac{\nu}{2})] \\ & -\frac{n}{2}\ln|\boldsymbol{\Sigma}| + \frac{n}{2}(\nu+p)\ln\nu \\ & -\frac{\nu+p}{2}\sum_{j=1}^{n}\frac{\nu + \delta(\boldsymbol{w}_j, \boldsymbol{\mu}; \boldsymbol{\Sigma})}{\nu^{(t)} + \delta(\boldsymbol{w}_j, \boldsymbol{\mu}^{(t)}; \boldsymbol{\Sigma}^{(t)})} \\ & +\frac{\nu+p}{2}\left\{n - \sum_{j=1}^{n}\ln[\nu^{(t)} + \delta(\boldsymbol{w}_j, \boldsymbol{\mu}^{(t)}; \boldsymbol{\Sigma}^{(t)})]\right\} \end{aligned}$$

Maximization step

Block ascent for maximizing the minorizing function

- Given $\nu^{(t)}$, update $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ by

$$\max_{\boldsymbol{\mu},\boldsymbol{\Sigma}} \quad -\frac{n}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{j=1}^{n}\frac{\nu^{(t)}+p}{\nu^{(t)}+\delta(\boldsymbol{w}_j,\boldsymbol{\mu}^{(t)};\boldsymbol{\Sigma}^{(t)})}\delta(\boldsymbol{w}_j,\boldsymbol{\mu};\boldsymbol{\Sigma})$$

  A weighted multivariate normal problem – same update as in the vanilla EM(=ECM)

- Given $(\boldsymbol{\mu}^{(t+1)}, \boldsymbol{\Sigma}^{(t+1)})$, update of ν is easy since it is a univariate optimization problem. This gives a different update from the vanilla EM(=ECM)

- Above derivation gives an MCM (minorization conditional maximization) algorithm, which turns out to be different from ECM
- Wait a minute, why don't we maximize the original objective function directly when updating ν? MCM Either!
- MCME = ECME. Why?

Yet another MM for t ...

Assume ν known

- Notice

$$\begin{aligned} & -\frac{1}{2}|\boldsymbol{\Sigma}| - \frac{\nu+p}{2}\ln[\nu+\delta(\boldsymbol{w}_j,\boldsymbol{\mu};\boldsymbol{\Sigma})] \\ = \quad & -\frac{\nu+p}{2}\ln\{|\boldsymbol{\Sigma}|^a[\nu+\delta(\boldsymbol{w}_j,\boldsymbol{\mu};\boldsymbol{\Sigma})]\} \end{aligned}$$

  where $a = 1/(\nu+p)$ is the working parameter

- Minorize via

$$\begin{aligned} & -\ln\{|\boldsymbol{\Sigma}|^a[\nu+\delta(\boldsymbol{w}_j,\boldsymbol{\mu};\boldsymbol{\Sigma})]\} \\ \geq \quad & -\frac{|\boldsymbol{\Sigma}|^a[\nu+\delta(\boldsymbol{w}_j,\boldsymbol{\mu};\boldsymbol{\Sigma})]}{|\boldsymbol{\Sigma}^{(t)}|^a[\nu^{(t)}+\delta(\boldsymbol{w}_j,\boldsymbol{\mu}^{(t)};\boldsymbol{\Sigma}^{(t)})]} + 1 \\ & -\ln\{|\boldsymbol{\Sigma}^{(t)}|^a[\nu^{(t)}+\delta(\boldsymbol{w}_j,\boldsymbol{\mu}^{(t)};\boldsymbol{\Sigma}^{(t)})]\} \end{aligned}$$

- Now working out the maximization step (do it!) yields the efficient data augmentation algorithm in (Meng and van Dyk, 1997)

Non-negative matrix factorization (NNMF)

Lee and Seung (1999, 2001)

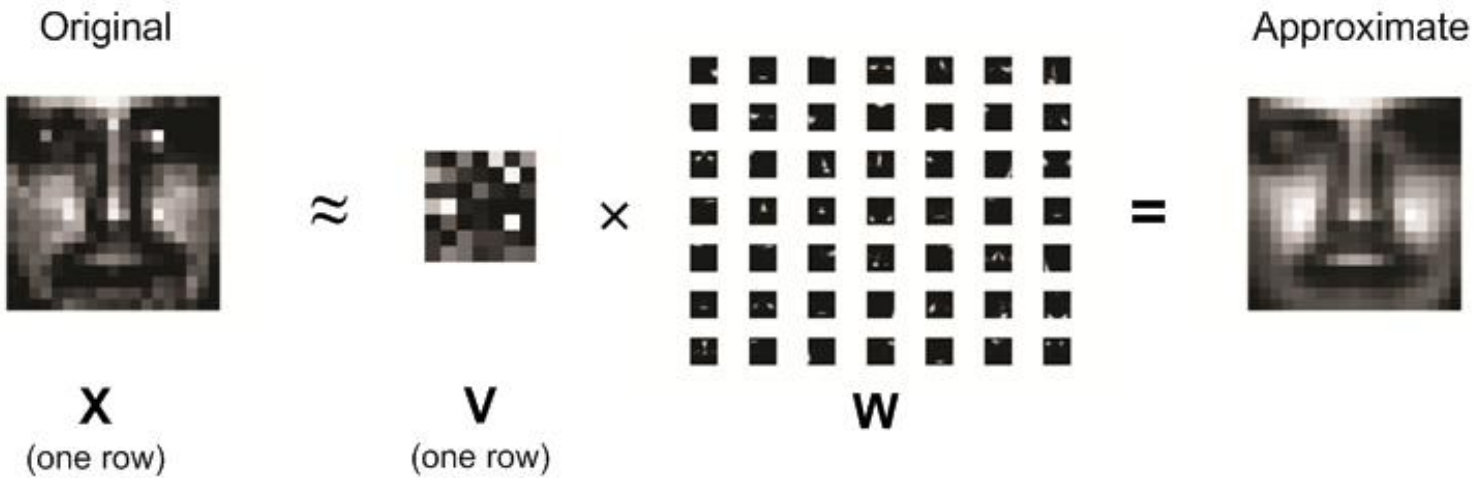
- Goal: Find a low rank r factorization $\mathbf{V}_{m\times r}\mathbf{W}_{r\times n}$ of non-negative data matrix $\mathbf{X}_{m\times n}$, such that

$$||\mathbf{X} - \mathbf{VW}||_{\mathrm{F}}$$

  is minimized subject to constraints $v_{ij}, w_{jk} \geq 0$
- Resembler SVD except the non-negativity constraint
- A popular alternative to PCA and vector quantization
- Used for data compression, clustering, ...

Image Factorization

- MIT CBCL Face Images: 2429 faces, each $19 \times 19 = 361$ pixels
- A striking feature of NNMF basis images is they correspond to different parts of face (eyes, nose, ...)



approximation based on a rank-49 NNMF

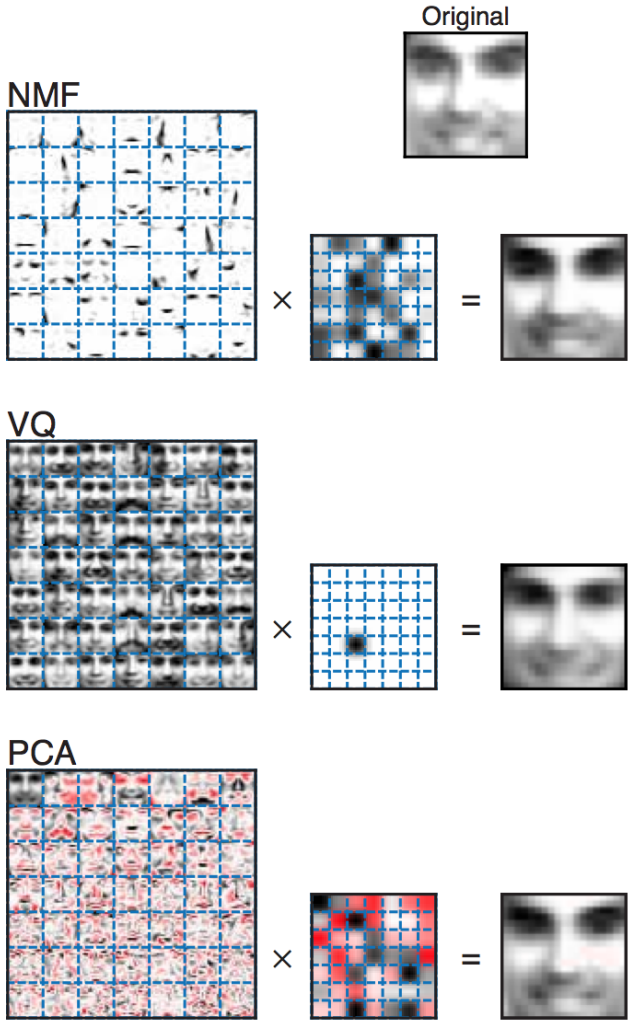


FIGURE 14.33. *Non-negative matrix factorization (NMF), vector quantization (VQ, equivalent to k-means clustering) and principal components analysis (PCA) applied to a database of facial images. Details are given in the text. Unlike VQ and PCA, NMF learns to represent faces with a set of basis images resembling parts of faces.*

MM for NNMF

- Objective

$$\|\boldsymbol{X} - \boldsymbol{V}\boldsymbol{W}\|_{\mathrm{F}} = \sum_i \sum_j \left(x_{ij} - \sum_k v_{ik} w_{kj} \right)^2$$

- Majorization: by convexity of s^2 function

$$\left(x_{ij} - \sum_k v_{ik} w_{kj} \right)^2 \leq \sum_k \frac{v_{ik}^{(t)} w_{kj}^{(t)}}{\sum_l v_{il}^{(t)} w_{lj}^{(t)}} \left(x_{ij} - \frac{\sum_l v_{il}^{(t)} w_{lj}^{(t)}}{v_{ik}^{(t)} w_{kj}^{(t)}} v_{ik} w_{kj} \right)^2$$

- Minimization: alternate update of $\boldsymbol{V}$ and $\boldsymbol{W}$

MM Algorithm for NNMF

Initialize: $\mathbf{V} = \text{rand}(m, r)$ and $\mathbf{W} = \text{rand}(r, n)$.
repeat
$\mathbf{V}_{..} \leftarrow \mathbf{V}_{..} \times (\mathbf{X}\mathbf{W}^T)_{..} / (\mathbf{V}\mathbf{W}\mathbf{W}^T)_{..}$ for all $m \times r$ elements
$\mathbf{W}_{..} \leftarrow \mathbf{W}_{..} \times (\mathbf{V}^T\mathbf{X})_{..} / (\mathbf{V}^T\mathbf{V}\mathbf{W})_{..}$ for all $r \times n$ elements
until convergence

- Updates are extremely simple
- Non-negativity constraints are satisfied
- Utilize high throughput of BLAS 3 routines, either on CPU or on GPU
- Lasso or other sparsity penalty can be incorporated

Exercise

- In the original paper, Lee and Seung (1999) pose a Poisson model for the image pixels $x_{ij} \sim \text{Poisson}(\sum_k v_{ik} w_{kj})$
- The log-likelihood is

$$L(\boldsymbol{V}, \boldsymbol{W}) = \sum_i \sum_j \left[x_{ij} \ln \left(\sum_k v_{ik} w_{kj} \right) - \sum_k v_{ik} w_{kj} \right]$$

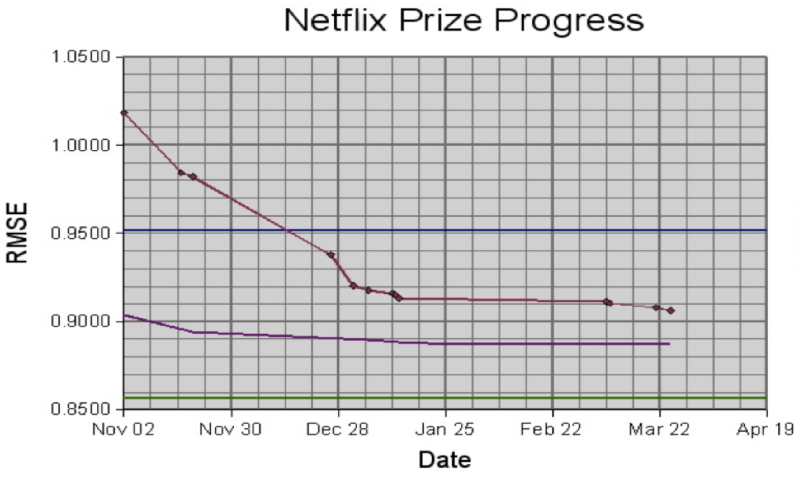
- MLE under the Poisson model is amenable to another MM algorithm
- Derive it

Netflix movie rating data and matrix completion

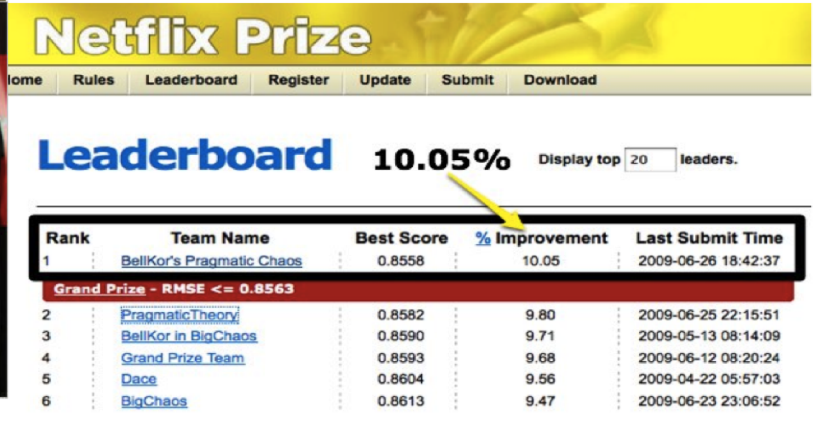
ID	movie 1	movie 2	movie 3	movie 4	movie 5	movie 6	...	movie 17,770
user 1	5	3	4	3	3	*NA*	...	1
user 2	4	*NA*	*NA*	*NA*	*NA*	*NA*	...	*NA*
user 3	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	...	*NA*
user 4	4	*NA*	*NA*	*NA*	*NA*	2	...	4
user 5	*NA*	*NA*	*NA*	5	*NA*	*NA*	...	*NA*
user 6	3	*NA*	*NA*	5	1	*NA*	...	3
user 7	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	...	*NA*
user 8	5	*NA*	5	*NA*	*NA*	*NA*	...	*NA*
user 9	*NA*	*NA*	*NA*	*NA*	3	*NA*	...	*NA*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
user 480,189	*NA*	5	*NA*	*NA*	*NA*	*NA*	...	*NA*

- Snapshot of the kind of data collected by Netflix. Only 100,480,507 ratings (1.2% entries of the matrix) are observed
- Goal: impute the unobserved ratings for personalized recommendation


NETFLIX


Netflix Prize Progress
RMSE
1.0500
1.0000
0.9500
0.9000
0.8500
Nov 02
Nov 30
Dec 28
Jan 25
Feb 22
Mar 22
Apr 19
Date


2009
NETFLIX
DATE 09.21.09
PAY TO THE ORDER OF BellKor's Pragmatic Chaos
$ 1,000,000 00
AMOUNT ONE MILLION
00/100
FOR The Netflix Prize
Reed Hastings


Netflix Prize
Home
Rules
Leaderboard
Register
Update
Submit
Download
Leaderboard
10.05%
Display top 20 leaders.
Rank
Team Name
Best Score
% Improvement
Last Submit Time
1
BellKor's Pragmatic Chaos
0.8558
10.05
2009-06-26 18:42:37
Grand Prize - RMSE <= 0.8563
2
PragmaticTheory
0.8582
9.80
2009-06-25 22:15:51
3
BellKor in BigChaos
0.8590
9.71
2009-05-13 08:14:09
4
Grand Prize Team
0.8593
9.68
2009-06-12 08:20:24
5
Dace
0.8604
9.56
2009-04-22 05:57:03
6
BigChaos
0.8613
9.47
2009-06-23 23:06:52

- Observe a very sparse matrix $\boldsymbol{Y} = (y_{ij})$. Want to impute all the missing entries. It is possible when the matrix is structured and of low rank
- Let $\Omega = \{(i,j) : \text{observed entries}\}$ collect the observed entries and $P_\Omega(\boldsymbol{M})$ denote the projection of matrix $\boldsymbol{M}$ to Ω. The problem

$$\min_{\text{rank}(\boldsymbol{X}) \le r} \frac{1}{2} \|P_\Omega(\boldsymbol{Y}) - P_\Omega(\boldsymbol{X})\|_{\text{F}}^2 = \frac{1}{2} \sum_{(i,j)\in\Omega} (y_{ij} - x_{ij})^2$$

  unfortunately is non-convex and difficult
- Convex relaxation (Mazumder et al., 2010)

$$\min_{\boldsymbol{X}} f(\boldsymbol{X}) = \frac{1}{2} \|P_\Omega(\boldsymbol{Y}) - P_\Omega(\boldsymbol{X})\|_{\text{F}}^2 + \lambda \|\boldsymbol{X}\|_*,$$

  where $\|\boldsymbol{X}\|_* = \|\sigma(\boldsymbol{X})\|_1 = \sum_i \sigma_i(\boldsymbol{X})$ is the nuclear norm

Majorization step

A majorizing function

$$\begin{aligned} f(\boldsymbol{X}) &= \frac{1}{2}\sum_{(i,j)\in\Omega}(y_{ij}-x_{ij})^2+\frac{1}{2}\sum_{(i,j)\notin\Omega}0+\lambda\|\boldsymbol{X}\|_* \\ &\leq \frac{1}{2}\sum_{(i,j)\in\Omega}(y_{ij}-x_{ij})^2+\frac{1}{2}\sum_{(i,j)\notin\Omega}(x_{ij}^{(t)}-x_{ij})^2+\lambda\|\boldsymbol{X}\|_* \\ &= \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{Z}^{(t)}\|_{\mathrm{F}}^2+\lambda\|\boldsymbol{X}\|_* \\ &= g(\boldsymbol{X}|\boldsymbol{X}^{(t)}), \end{aligned}$$

where $\boldsymbol{Z}^{(t)} = P_\Omega(\boldsymbol{Y}) + P_{\Omega^\perp}(\boldsymbol{X}^{(t)})$

Minimization step

- Majorizing function

$$g(\boldsymbol{X}|\boldsymbol{X}^{(t)}) = \frac{1}{2}\|\boldsymbol{X}\|_{\mathrm{F}}^2 - \mathrm{tr}(\boldsymbol{X}^{\mathsf{T}}\boldsymbol{Z}^{(t)}) + \frac{1}{2}\|\boldsymbol{Z}^{(t)}\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{X}\|_*$$

- Observe

$$\begin{aligned}\|\boldsymbol{X}\|_{\mathrm{F}}^2 &= \|\sigma(\boldsymbol{X})\|_2^2 = \sum_i \sigma_i^2 \\ \|\boldsymbol{Z}^{(t)}\|_{\mathrm{F}}^2 &= \|\sigma(\boldsymbol{Z}^{(t)})\|_2^2 = \sum_i \omega_i^2\end{aligned}$$

and by Fan-von Neuman's inequality

$$\mathrm{tr}(\boldsymbol{X}^{\mathsf{T}}\boldsymbol{Z}^{(t)}) \leq \sum_i \sigma_i\omega_i$$

with equality achieved if and only if the left and right singular vectors of the two matrices coincide

- Thus we can assume $\boldsymbol{X}$ has same singular vectors as $\boldsymbol{Z}^{(t)}$ and

$$\begin{aligned} g(\boldsymbol{X}|\boldsymbol{X}^{(t)}) &= \frac{1}{2}\sum_i \sigma_i^2 - \sum_i \sigma_i\omega_i + \frac{1}{2}\omega_i^2 + \lambda\sum_i \sigma_i \\ &= \frac{1}{2}\sum_i (\sigma_i - \omega_i)^2 + \lambda\sum_i \sigma_i, \end{aligned}$$

with minimizer given by $\sigma_i^{(t+1)} = (\omega_i - \lambda)_+$

MM algorithm for matrix completion

Initialize $\boldsymbol{X}^{(0)} \in \mathbb{R}^{m \times n}$
repeat
  $\boldsymbol{Z}^{(t+1)} \leftarrow P_\Omega(\boldsymbol{Y}) + P_{\Omega^\perp}(\boldsymbol{X}^{(t)})$
  Compute SVD $\boldsymbol{Z}^{(t+1)} = \boldsymbol{U}\text{diag}(\boldsymbol{w})\boldsymbol{V}^\top$
  $\boldsymbol{X}^{(t+1)} \leftarrow \boldsymbol{U}\text{diag}[(\boldsymbol{w} - \lambda)_+]\boldsymbol{V}^\top$
until objective value converges

- "Golub-Kahan-Reinsch" algorithm takes $4m^2n + 8mn^2 + 9n^3$ flops for a $m \geq n$ matrix and is not going to work for 480K-by-18K Netflix matrix. Notice only top singular values are needed. Lanczos/Arnoldi algorithm is the way to go

Outline

Acceleration

- Slow convergence of EM/MM (linear rate at best) is a serious concern in many problems
- Some proposed acceleration methods
  - Aitken's and Louis' methods (Louis, 1982)
  - Conjugate gradient method (Jamshidian and Jennrich, 1993)
  - Quasi-Newton acceleration (Lange, 1995; Jamshidian and Jennrich, 1997)
  - Ikeda acceleration (Ikeda, 2000)
- We review two recent ones which are particularly attractive in high-dimensional setting

SQUAREM

Varadhan and Roland (2008)

- $F : \mathbb{R}^p \mapsto \mathbb{R}^p$ algorithm mapping

$$\boldsymbol{x} \to F(\boldsymbol{x}) \to F \circ F(\boldsymbol{x}) \to F \circ F \circ F(\boldsymbol{x}) \to \cdots$$

- SQUAREM extrapolation

$$\begin{aligned} \boldsymbol{x}^{(t+1)} &= \boldsymbol{x}^{(t)} - 2s[F(\boldsymbol{x}^{(t)}) - \boldsymbol{x}^{(t)}] + s^2[F \circ F(\boldsymbol{x}^{(t)}) - 2F(\boldsymbol{x}^{(t)}) + \boldsymbol{x}^{(t)}] \\ &= \boldsymbol{x}^{(t)} - 2s\boldsymbol{u} + s^2\boldsymbol{v} \end{aligned}$$

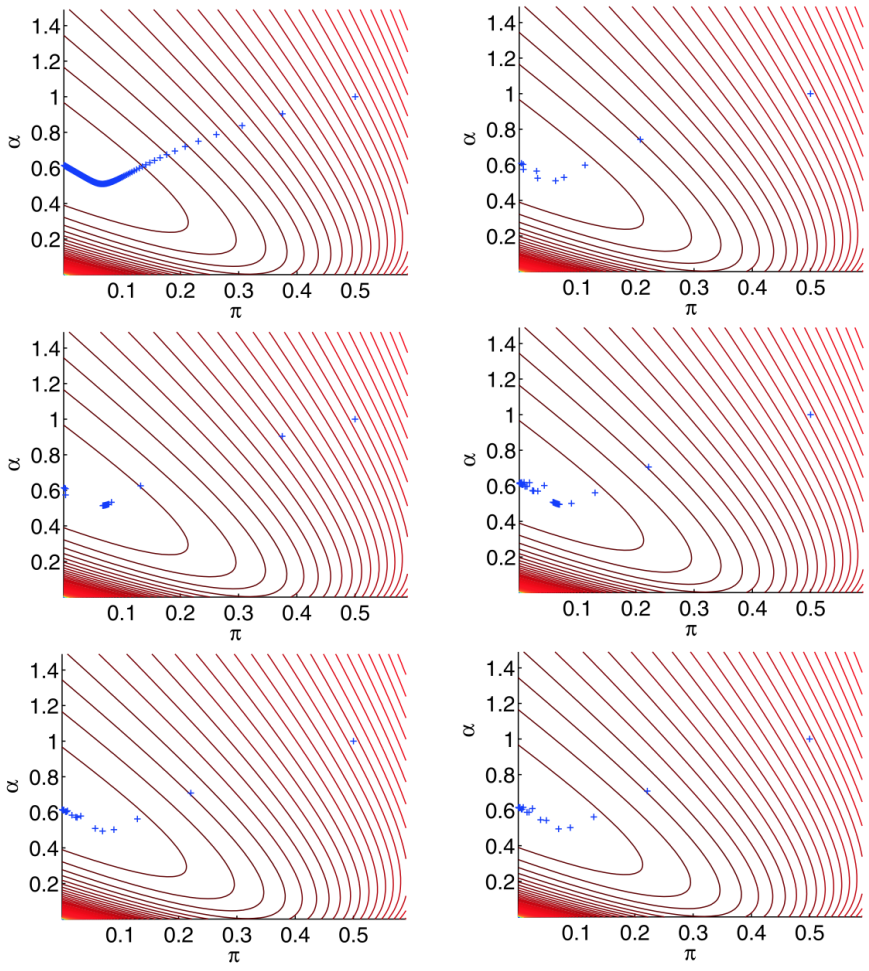
where s is a scalar

  - SqS1: $s = \frac{\boldsymbol{u}^\top \boldsymbol{u}}{\boldsymbol{u}^\top(\boldsymbol{v}-\boldsymbol{u})}$
  - SqS2: $s = \frac{\boldsymbol{u}^\top(\boldsymbol{v}-\boldsymbol{u})}{(\boldsymbol{v}-\boldsymbol{u})^\top(\boldsymbol{v}-\boldsymbol{u})}$
  - SqS3: $s = -\sqrt{\frac{\boldsymbol{u}^\top \boldsymbol{u}}{(\boldsymbol{v}-\boldsymbol{u})^\top(\boldsymbol{v}-\boldsymbol{u})}}$

- Extremely simple and effective
- Ascent property may be lost by the extrapolation; but we can always revert back to the EM/MM update as safeguard

An example with 2 parameters

Fig. 1 Ascent of the different algorithms for the Lidwell and Somerville household type (a) data starting from $(\pi^0, \alpha^0) = (0.5, 1)$ with stopping criterion $\varepsilon = 10^{-9}$. *Top left*: naive MM; *Top right*: $q = 1$; *Middle left*: $q = 2$; *Middle right*: SqS1; *Bottom left*: SqS2; *Bottom right*: SqS3



Quasi-Newton acceleration

Zhou et al. (2011)

- $F : \mathbb{R}^p \mapsto \mathbb{R}^p$ algorithm mapping

$$\boldsymbol{x} \to F(\boldsymbol{x}) \to F \circ F(\boldsymbol{x}) \to F \circ F \circ F(\boldsymbol{x}) \to \cdots$$

- Fixed points of $F \quad \Leftrightarrow \quad$ roots of $G(\boldsymbol{x}) = \boldsymbol{x} - F(\boldsymbol{x})$
- Newton's method for finding the roots of the mapping $G(\boldsymbol{x}) = \boldsymbol{x} - F(\boldsymbol{x})$

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - [I - dF(\boldsymbol{x}^{(t)})]^{-1}[\boldsymbol{x}^{(t)} - F(\boldsymbol{x}^{(t)})]$$

- Idea: Approximate $dF(\boldsymbol{x}^{(t)})$ by a low-rank matrix M and explicitly form the inverse $(I - M)^{-1}$

- Secant condition: $u = F(x^{(t)}) - x^{(t)}, \quad v = F \circ F(x^{(t)}) - F(x^{(t)})$.

$$Mu = v,$$

where $M = dF(x^{(t)})$

- For best results, we collect q secant pairs u_i, v_i, $i = 1, \ldots, q$ and require that

$$Mu_i = v_i, \quad i = 1, \ldots, q, \quad \text{or}$$
$$MU = V.$$

Lemma

The minimum of the strictly convex function $\|M\|_F^2$ subject to the constraint $MU = V$ is attained by the choice $M = V(U^T U)^{-1} U^T$.

- Fortunately, we have the explicit inverse by Woodbury formula

$$(I - M)^{-1} = [I - V(U^T U)^{-1} U^T]^{-1} = I + V[U^T (U - V)]^{-1} U^T$$

- Quasi-Newton update

$$\begin{aligned} x^{(t+1)} &= x^{(t)} - (I - M)^{-1}[x^{(t)} - F(x^{(t)})] \\ &= x^{(t)} - [I - V(U^T U)^{-1} U^T]^{-1}[x^{(t)} - F(x^{(t)})] \\ &= F(x^{(t)}) - V[U^T (U - V)]^{-1} U^T [x^{(t)} - F(x^{(t)})] \end{aligned}$$

Advantages of the new quasi-Newton acceleration scheme

$$x^{(t+1)} = F(x^{(t)}) - V[U^T(U - V)]^{-1}U^T[x^{(t)} - F(x^{(t)})]$$

- The effort per iteration is light: $O(pq^2) + O(q^3)$
- Linear constraints are preserved
- Do not need to store the whole approximate (inverse) Hessian matrix:

$$O(p^2) \text{ vs } O(pq)$$

$p = \#$ parameters, $q = \#$ secant conditions

An example with 2771 parameters

Table 8 Comparison of accelerations for the movie rating problem. Here the starting point is $\pi_i = \alpha_i = \beta_j = 0.5$, the stopping criterion is $\varepsilon = 10^{-9}$, and the number of parameters equals 2,771

Algorithm	$\ln L$	Evals	Time
EM	−119085.2039	671	189.3020
$q = 1$	−119085.2020	215	64.1149
$q = 2$	−119085.1983	116	36.6745
$q = 3$	−119085.1978	153	46.0387
$q = 4$	−119085.1961	156	46.9827
$q = 5$	−119085.1974	161	48.6629
SqS1	−119085.2029	341	127.9918
SqS2	−119085.2019	301	110.9871
SqS3	−119085.2001	157	56.7568

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Statist.*, 41:164–171.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Ser. B*, 57:289–300.

Booth, J. G. and Hobert, J. P. (1999). Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodology)*, 61(1):265–285.

Celeux, G. and Diebolt, J. (1985). The SEM algorithm: A probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quarterly*, 2:73–82.

Celeux, G. and Diebolt, J. (1989). Une version de type recuit simulé de l'algorithme EM. Rapport de Recherche INRIA.

Cooley, J. W. and Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. *Math. Comp.*, 19:297–301.

Cox, D. R. (1972). Regression models and life-tables. *J. Roy. Statist. Soc. Ser. B*, 34:187–220. With discussion by F. Downton, Richard Peto, D. J. Bartholomew, D. V. Lindley, P. W. Glassborow, D. E. Barton, Susannah Howard, B. Benjamin, John J. Gart, L. D. Meshalkin, A. R. Kagan, M. Zelen, R. E. Barlow, Jack Kalbfleisch, R. L. Prentice and Norman Breslow, and a reply by D. R. Cox.

de Leeuw, J. and Heiser, W. (1977). Convergence of correction matrix algorithms for multidimensional scaling. *Geometric Representations of Relational Data*.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Soceity Series B.*, 39(1-38).

Dickey, D. A. and Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *J. Amer. Statist. Assoc.*, 74(366, part 1):427–431.

Efron, B. (1979). Bootstrap methods: another look at the jackknife. *Ann. Statist.*, 7(1):1–26.

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *J. Amer. Statist. Assoc.*, 85(410):398–409.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, New York, second edition.

Ikeda, S. (2000). Acceleration of the EM algorithms. *Systems and Computers in Japan*, 31(2):10–18.

Jamshidian, M. and Jennrich, R. I. (1993). Conjugate gradient acceleration of the EM algorithm. *J. Amer. Statist. Assoc.*, 88(421):221–228.

Jamshidian, M. and Jennrich, R. I. (1997). Acceleration of the EM algorithm by using quasi-Newton methods. *J. Roy. Statist. Soc. Ser. B*, 59(3):569–587.

Kaplan, E. L. and Meier, P. (1958). Nonparametric estimation from incomplete observations. *J. Amer. Statist. Assoc.*, 53:457–481.

Lange, K. (1995). A quasi-Newton acceleration of the EM algorithm. *Statist. Sinica*, 5(1):1–18.

Lange, K., Hunter, D. R., and Yang, I. (2000). Optimization transfer using surrogate objective functions. *J. Comput. Graph. Statist.*, 9(1):1–59. With discussion, and a rejoinder by Hunter and Lange.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

Lee, D. D. and Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562. MIT Press.

Liu, C. and Rubin, D. B. (1994). The ECME algorithm: a simple extension of EM and ECM with faster monotone convergence. *Biometrika*, 81(4):633–648.

Liu, C., Rubin, D. B., and Wu, Y. N. (1998). Parameter expansion to accelerate EM: the PX-EM algorithm. *Biometrika*, 85(4):755–770.

Liu, J. S. and Wu, Y. N. (1999). Parameter expansion for data augmentation. *J. Amer. Statist. Assoc.*, 94(448):1264–1274.

Louis, T. A. (1982). Finding the observed information matrix when using the EM algorithm. *J. Roy. Statist. Soc. Ser. B*, 44(2):226–233.

Mazumder, R., Hastie, T., and Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11:2287–2322.

Meng, X.-L. and Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: a general framework. *Biometrika*, 80(2):267–278.

Meng, X.-L. and van Dyk, D. (1997). The EM algorithm—an old folk-song sung to a fast new tune. *J. Roy. Statist. Soc. Ser. B*, 59(3):511–567. With discussion and a reply by the authors.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

Neal, R. M. and Hinton, G. E. (1999). A view of the EM algorithm that justifies incremental, sparse, and other variants. In Jordan, M. I., editor, *Learning in graphical models*, pages 355–368. MIT Press, Cambridge, MA, USA.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer-Verlag, New York, second edition.

Suchard, M. A., Wang, Q., Chan, C., Frelinger, J., Cron, A., and West, M. (2010). Understanding GPU programming for statistical computation: studies in massively parallel massive mixtures. *Journal of Computational and Graphical Statistics*, 19(2):419–438.

Tanner, M. A. and Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *J. Amer. Statist. Assoc.*, 82(398):528–550. With discussion and with a reply by the authors.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B*, 58(1):267–288.

Tropp, J. A. (2006). Just relax: convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030–1051.

Varadhan, R. and Roland, C. (2008). Simple and globally convergent methods for accelerating the convergence of any EM algorithm. *Scand. J. Statist.*, 35(2):335–353.

Wei, G. C. G. and Tanner, M. A. (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704.

Zhou, H., Alexander, D., and Lange, K. (2011). A quasi-Newton acceleration for high-dimensional optimization algorithms. *Statistics and Computing*, 21:261–273.