15 Lecture 15, Feb 23

Announcements

- HW4 due today @ 11:59PM.
- HW5 (Newton's method, handwritten digit recognition) posted. Due next Tue Mar 1 @ 11:59PM.
- Quiz 3 this Thu 2/25. In class, closed book.

Last time

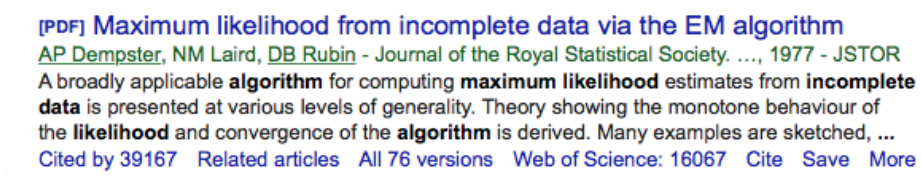
- Newton-Raphson and Fisher scoring method.
- GLMs: Fisher scoring algorithm (IRWLS).
- Non-linear regression: Gauss-Newton algorithm.
- EM algorithm: introduction.

Today

- EM algorithm.
- Examples of EM algorithm.
- MM algorithm.
- Examples of MM algorithm.

EM algorithm (KL Chapter 13)

- History: Dempster et al. (1977b).


[PDF] Maximum likelihood from incomplete data via the EM algorithm
AP Dempster, NM Laird, DB Rubin - Journal of the Royal Statistical Society. ..., 1977 - JSTOR
A broadly applicable **algorithm** for computing **maximum likelihood** estimates from **incomplete data** is presented at various levels of generality. Theory showing the monotone behaviour of the **likelihood** and convergence of the **algorithm** is derived. Many examples are sketched, ...
Cited by 39167 Related articles All 76 versions Web of Science: 16067 Cite Save More

Same idea appears in parameter estimation in HMM (Baum-Welch algorithm) (Baum et al., 1970).

A maximization technique occurring in the **statistical** analysis of probabilistic functions of Markov chains
LE Baum, T Petrie, G Soules, N Weiss - The annals of mathematical **statistics**, 1970 - JSTOR
PYI... YT (**A**, **a**, f) and the difficult analysis of **maximizing** this function of **A** for very special choices of f presented in [2],[8] that **a** simple explicit procedure for **maximization** for **a** general f would be quite difficult; however, this is not the case.
Cited by 3102 Related articles All 4 versions Cite

- 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 easy.
- Let $f(\boldsymbol{x}|\boldsymbol{\theta}) = f(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{\theta})$ be the density of complete data.
- Iterative two step procedure
  - 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

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

- (Ascent property of EM algorithm) By the information inequality,

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

Rearranging shows that (fundamental inequality of EM)

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

for all $\boldsymbol{\theta}$ and in particular

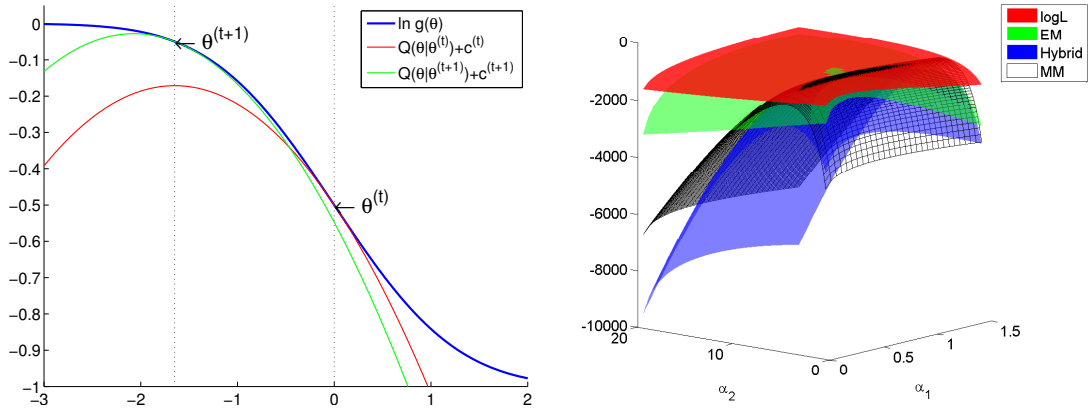
$$\begin{aligned} \ln g(\boldsymbol{y} \mid \boldsymbol{\theta}^{(t+1)}) &\geq Q(\boldsymbol{\theta}^{(t+1)} \mid \boldsymbol{\theta}^{(t)}) - Q(\boldsymbol{\theta}^{(t)} \mid \boldsymbol{\theta}^{(t)}) + \ln g(\boldsymbol{y} \mid \boldsymbol{\theta}^{(t)}) \\ &\geq \ln g(\boldsymbol{y} \mid \boldsymbol{\theta}^{(t)}).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

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*).

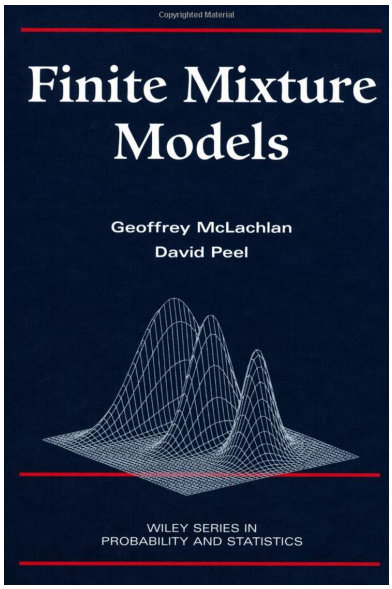
- Intuition? Keep these pictures in mind

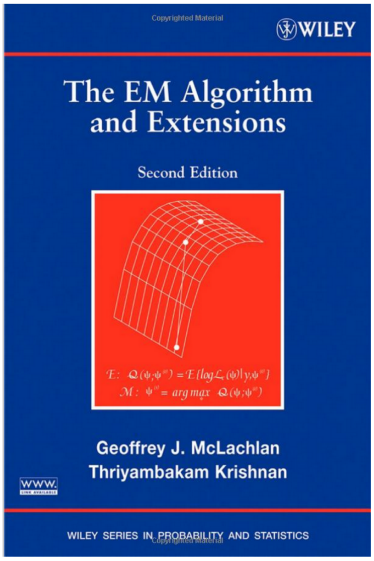


- Under mild regularity conditions, $\boldsymbol{\theta}^{(t)}$ converges to a stationary point of $\ln g(\boldsymbol{y}|\boldsymbol{\theta})$ (Wu, 1983).

- Numerous applications of EM:
  finite mixture model, HMM (Baum-Welch algorithm), factor analysis, variance components model aka linear mixed model (LMM), hyper-parameter estimation in empirical Bayes procedure $\max_{\boldsymbol{\alpha}} \int f(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{\alpha})\, d\boldsymbol{\theta}$, missing data, group/censorized/truncated model, ...

A canonical example: finite mixture models





- Gaussian finite mixture models: mixture density

$$h(\boldsymbol{y}) = \sum_{j=1}^{k} \pi_j h_j(\boldsymbol{y} \mid \boldsymbol{\mu}_j, \boldsymbol{\Omega}_j), \quad \boldsymbol{y} \in \mathbb{R}^d,$$

  where

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

  are multivariate normals $N_d(\boldsymbol{\mu}_j, \boldsymbol{\Omega}_j)$.

- Given data $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$, want to estimate parameters

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

  subject to constraint $\pi_j \geq 0, \sum_{j=1}^{d} \pi_j = 1, \boldsymbol{\Omega}_j \succeq \mathbf{0}$. (Incomplete) data log-likelihood is

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

- 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: need to evaluate conditional expectation

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

By Bayes rule, we have

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

So the Q function becomes

$$\begin{aligned} & Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \\ = \; & \sum_{i=1}^{n}\sum_{j=1}^{k} w_{ij}^{(t)} \ln \pi_j + \sum_{i=1}^{n}\sum_{j=1}^{k} w_{ij}^{(t)} \left[-\frac{1}{2}\ln\det \boldsymbol{\Omega}_j - \frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{\mu}_j)^\intercal \boldsymbol{\Omega}_j^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu}_j)\right]. \end{aligned}$$

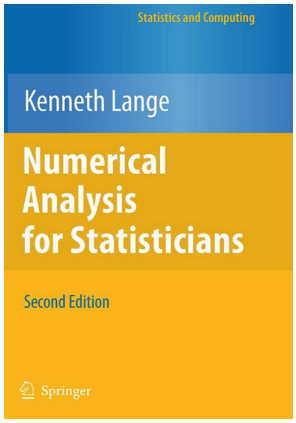
- M step: maximizer of the Q function gives the next iterate

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

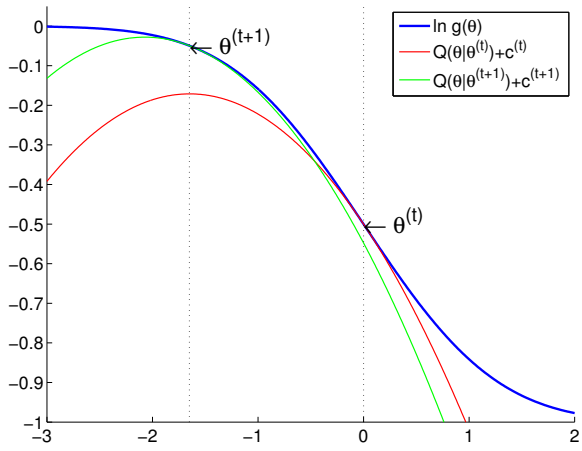
See KL Example 11.3.1 for multinomial MLE. See KL Example 11.2.3 for multivariate normal MLE.

- Compare these extremely simple updates to Newton type algorithms!
- Also note the ease of parallel computing with this EM algorithm. See, e.g., **Suchard, M. A.**; Wang, Q.; Chan, C.; Frelinger, J.; Cron, A. & West, M. Understanding GPU programming for statistical computation: studies in massively parallel massive mixtures. *Journal of Computational and Graphical Statistics*, 2010, 19, 419-438.
- In general, EM/MM algorithms are particularly attractive for parallel computing. See, e.g.,
  H Zhou, K Lange, & M Suchard. (2010) Graphical processing units and high-dimensional optimization, *Statistical Science*, 25:311-324.

MM algorithm (KL Chapter 12)



- Recall our picture for understanding the ascent property of EM



- EM as a minorization-maximization (MM) algorithm
  - The Q function constitutes a *minorizing* function of the objective function up to an additive constant

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

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

- Questions:
  - Is EM principle only limited to maximizing likelihood model?
  - Is there any other way to produce such surrogate function?
  - Can we flip the picture and apply same principle to *minimization* problem?

  These thoughts lead to a powerful tool – MM principle.
  Lange, K., Hunter, D. R., and Yang, I. (2000). Optimization transfer using surrogate objective functions. *J. Comput. Graph. Statist.*, 9(1):159. With discussion, and a rejoinder by Hunter and Lange.

- For maximization of $f(\boldsymbol{\theta})$ – minorization-maximization (MM) algorithm
  - **M**inorization step: Construct a surrogate function $g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ such that

$$\begin{aligned} f(\boldsymbol{\theta}) &\geq g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \quad \text{(dominance condition)} \\ f(\boldsymbol{\theta}^{(t)}) &= g(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) \quad \text{(tangent condition).} \end{aligned}$$

– **M**aximization step:

$$\boldsymbol{\theta}^{(t+1)} = \text{argmax}\, g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}).$$

- *Ascent* property of minorization-maximization algorithm

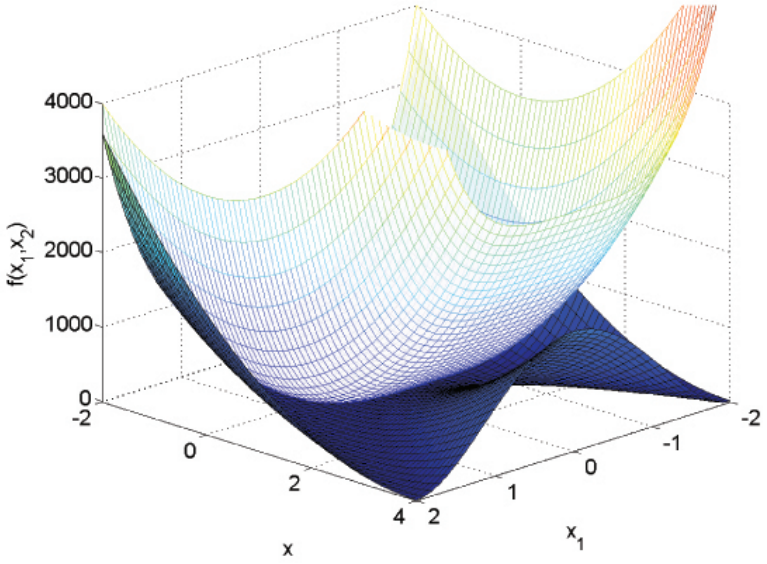
$$f(\boldsymbol{\theta}^{(t)}) = g(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) \leq g(\boldsymbol{\theta}^{(t+1)}|\boldsymbol{\theta}^{(t)}) \leq f(\boldsymbol{\theta}^{(t+1)}).$$

- EM is a special case of minorization-maximization (MM) algorithm.
- For minimization of $f(\boldsymbol{\theta})$ – majorization-minimization (MM) algorithm
  - **M**ajorization step: Construct a surrogate function $g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ such that

$$\begin{aligned} f(\boldsymbol{\theta}) &\leq g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \quad \text{(dominance condition)} \\ f(\boldsymbol{\theta}^{(t)}) &= g(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) \quad \text{(tangent condition)}. \end{aligned}$$

  - **M**inimization step:

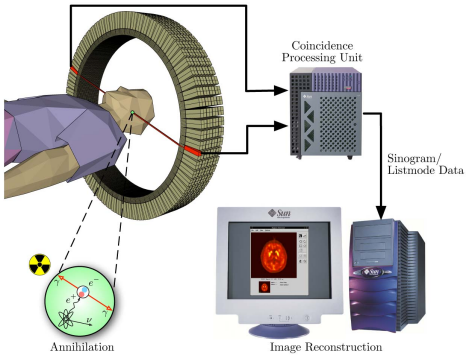
$$\boldsymbol{\theta}^{(t+1)} = \text{argmin}\, g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}).$$

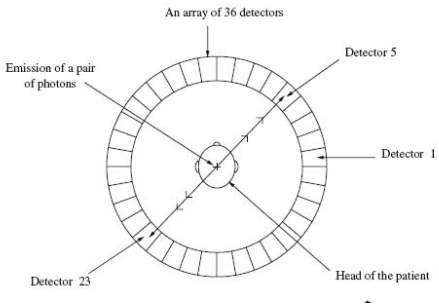


- Similarly we have the *descent* property of majorization-minimization algorithm.
- Same convergence theory as EM.

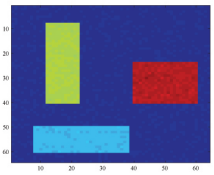
- Rational of the MM principle for developing optimization algorithms
  - Free the derivation from missing data structure.
  - Avoid matrix inversion.
  - Linearize an optimization problem.
  - Deal gracefully with certain equality and inequality constraints.
  - Turn a non-differentiable problem into a smooth problem.
  - Separate the parameters of a problem (perfect for massive, *fine-scale* parallelization).
- Generic methods for majorization and minorization – *inequalities*
  - Jensen's (information) 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
  - ...
- Numerous examples in KL chapter 12. Take Kenneth Lange's optimization course BIOMATH 210.
- History: the name *MM algorithm* originates from the discussion (by Xiaoli Meng) and rejoinder of the Lange et al. (2000) paper.

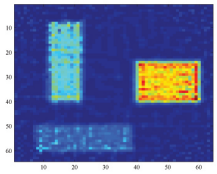
Example: PET imaging

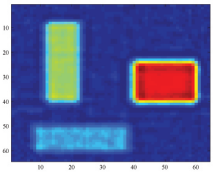




- Data: tube readings $\boldsymbol{y} = (y_1, \ldots, y_d)$.
- Estimate: photon emission intensities (pixels) $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)$.
- Poisson Model:
$$Y_i \sim \text{Poisson}\left(\sum_{j=1}^{p} c_{ij}\lambda_j\right),$$
where c_{ij} is the (pre-calculated) cond. prob. 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 $\lambda_j \geq 0$.
- Which algorithm?
  - Fisher scoring (IRWLS) = Newton.
    Need to solve a large linear system at each iteration ☹
  - EM algorithm: Lange and Carson (1984), Vardi et al. (1985)
$$\lambda_j^{(t+1)} = \lambda_j^{(t)} \sum_i \frac{y_i c_{ij}}{\sum_k c_{ik}\lambda_k^{(t)}}.$$
    Scales well with data size. Converges to the global maximum under mild conditions.
  - Exercise: derive the EM algorithm. (Hint: missing data z_{ij} = # of photons emitted from pixel i and received by detector j.)
- Issues: *grainy image* and *slow convergence*







- Regularized log-likelihood for smoother image:

$$\begin{aligned} & L(\boldsymbol{\lambda}|\boldsymbol{y}) - \frac{\mu}{2} \sum_{\{j,k\} \in \mathcal{N}} (\lambda_j - \lambda_k)^2 \\ = & \sum_i \left[y_i \ln \left(\sum_j c_{ij} \lambda_j \right) - \sum_j c_{ij} \lambda_j \right] - \frac{\mu}{2} \sum_{\{j,k\} \in \mathcal{N}} (\lambda_j - \lambda_k)^2, \end{aligned}$$

  where $\mu \geq 0$ is a tuning constant.

- EM algorithm does not (or is hard to) apply to the regularization term. Let's derive an MM algorithm.

- Minorization step:

  - By concavity of the $\ln s$ function

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

    Remark: this minorization depends on the positivity of both c_{ij} and λ_j.

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

$$\begin{aligned} -(\lambda_j - \lambda_k)^2 &= -\left(\frac{2\lambda_j - \lambda_j^{(t)} - \lambda_k^{(t)}}{2} + \frac{-2\lambda_k + \lambda_j^{(t)} + \lambda_k^{(t)}}{2} \right)^2 \\ &\geq -\frac{1}{2}(2\lambda_j - \lambda_j^{(t)} - \lambda_k^{(t)})^2 - \frac{1}{2}(2\lambda_k - \lambda_j^{(t)} - \lambda_k^{(t)})^2. \end{aligned}$$

  - 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 all λ_j are separated!
  - Solving for the root of

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

    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*.

- A simulation example with $n = 2016$ and $p = 4096$ (provided by Ravi Varadhan)

  CPU: i7 @ 3.20GHZ (1 thread), implemented using BLAS in the GNU Scientific Library (GSL)
  GPU: NVIDIA GeForce GTX 580, implemented using cuBLAS

	CPU			GPU			
Penalty μ	Iters	Time	Function	Iters	Time	Function	Speedup
0	100000	11250	-7337.152765	100000	140	-7337.153387	80
10^{-7}	24506	2573	-8500.082605	24506	35	-8508.112249	74
10^{-6}	6294	710	-15432.45496	6294	9	-15432.45586	79
10^{-5}	589	67	-55767.32966	589	0.8	-55767.32970	84

Example: Netflix and matrix completion

- Snapshot of the kind of data collected by Netflix. Only 100,480,507 ratings (1.2% entries of the 480K-by-18K matrix) are observed

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*

- Netflix challenge: impute the unobserved ratings for personalized recommendation. `http://en.wikipedia.org/wiki/Netflix_Prize`

- *Matrix completion problem.* Observe a very sparse matrix $\boldsymbol{Y} = (y_{ij})$. Want to impute all the missing entries. It is possible only when the matrix is structured, e.g., of *low rank*.

- Let $\Omega = \{(i,j) : \text{observed entries}\}$ index 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:

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

where $\boldsymbol{Z}^{(t)} = P_\Omega(\boldsymbol{Y}) + P_{\Omega^\perp}(\boldsymbol{X}^{(t)})$. "fill in missing entries by current imputation"

- Minimization step:

  Rewrite the surrogate function

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

  Let σ_i be the singular values of $\boldsymbol{X}$ and ω_i be the singular values of $\boldsymbol{Z}^{(t)}$. 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 the Fan-von Neuman's inequality

$$\mathrm{tr}(\boldsymbol{X}^\intercal\boldsymbol{Z}^{(t)}) \le \sum_i \sigma_i\omega_i$$

  with equality achieved if and only if the left and right singular vectors of the two matrices coincide. Thus we can choose $\boldsymbol{X}$ to have 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 $\sigma_i^{(t+1)} = (\omega_i - \lambda)_+$. "Singular value thresholding"

- 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: $\boldsymbol{U}\mathrm{diag}(\boldsymbol{w})\boldsymbol{V}^\intercal \leftarrow \boldsymbol{Z}^{(t+1)}$
    $\boldsymbol{X}^{(t+1)} \leftarrow \boldsymbol{U}\mathrm{diag}[(\boldsymbol{w} - \lambda)_+]\boldsymbol{V}^\intercal$
  until objective value converges

- "Golub-Kahan-Reinsch" algorithm takes $4m^2n + 8mn^2 + 9n^3$ flops for a $m \ge n$ matrix and is not going to work for 480K-by-18K Netflix matrix.

Notice only top singular values are needed since low rank solutions are achieved by large λ. Lanczos/Arnoldi algorithm is the way to go. Matrix-vector multiplication $\boldsymbol{Z}^{(t)}\boldsymbol{v}$ is fast (why?)