Monte Carlo and Optimization

Monte Carlo EM (MCEM) / Monte Carlo MLE

Simulated Annealing

Monte Carlo EM

Wei and Tanner (1990), Guo and Thompson (1992), Levine and Casella (2001)

As we have seen before, a function we wish to optimize might be an integral.

For example, in the EM setup, the likelihood to be optimized is of the form

$$g\left(X|\theta\right)=\int f\left(X,Y|\theta\right)dY$$

where x is the observed data and y is the "missing" data.

In EM, optimization of $g(x|\theta)$ is performed by the optimization of the sequence of

$$Q(\theta|\theta_n) = E\left[\log f(X,Y|\theta)|X,\theta_n\right]$$

where Y is chosen so that $Q(\theta|\theta_n)$ is easy to determine and optimize.

In particular, Y is chosen so that the integral

$$\int \log f(X,Y|\theta) f(Y|X,\theta_n)\,dY$$

is easy to evaluate.

One idea is to replace the direct integration with Monte Carlo Integration.

Monte Carlo EM (Wei and Tanner version)

MC E-step

Sample $y_1,\ldots,y_m \sim f(Y \mid X,\theta_n)$

Let $\hat{Q}(\theta|\theta_n) = \frac{1}{m}\sum_{i=1}^{m} \log f(X,y_i|\theta)$

$\hat{Q}(\theta|\theta_n)$ is a MC estimate of $Q(\theta|\theta_n)$

MC M-step

Set

$$\theta_{n+1} = \arg\sup \hat{Q}\left(\theta \middle| \theta_n\right)$$

Example: Genetic Linkage

$$\left(Y_1, Y_2, Y_3, Y_4\right)$$

$$\sim \text{Multi}\left(197, \left(\frac{\lambda}{4}, \frac{1-\lambda}{4}, \frac{1-\lambda}{4}, \frac{2+\lambda}{4}\right)\right)$$

$$\left(X_1, X_2, X_3, X_4, X_5\right)$$

$$\sim \text{Multi}\left(197, \left(\frac{\lambda}{4}, \frac{1-\lambda}{4}, \frac{1-\lambda}{4}, \frac{\lambda}{4}, \frac{1}{2}\right)\right)$$

MC E-step:

$$Q\left(\lambda \middle| \lambda_n\right)$$

$$= E\left[\left(X_1 + X_4\right)\log\lambda + \left(X_2 + X_3\right)\log\left(1-\lambda\right) \middle| Y, \lambda_n\right]$$

$$Q\left(\lambda \middle| \lambda_n\right) = Y_1 \log\lambda + \left(Y_2 + Y_3\right)\log\left(1-\lambda\right)$$

$$+ E\left[X_4 \log\lambda \middle| Y, \lambda_n\right]$$

$$= Y_1 \log\lambda + \left(Y_2 + Y_3\right)\log\left(1-\lambda\right)$$

$$+ E\left[X_4 \log\lambda \middle| Y_4, \lambda_n\right]$$

$$= \left(Y_1 + \hat{X}_4\right)\log\lambda + \left(Y_2 + Y_3\right)\log\left(1-\lambda\right)$$

where $\hat{X}_4 = E\left[X_4 \middle| Y_4, \lambda_n\right]$.

Now $X_4 \mid Y_4 \sim \text{Bin}\left(Y_4, \dfrac{\lambda}{\lambda+2}\right)$ so

sample $x_4^{(1)}, \ldots, x_4^{(m)} \sim \text{Bin}\left(Y_4, \lambda_n\right)$

$$\begin{aligned} \hat{X}_4 &= \frac{1}{m}\sum_{i=1}^{m} x_4^{(i)} \\ &= \hat{E}\left[X_4 \middle| Y_4, \lambda_n\right] \end{aligned}$$

M-step:

$$\lambda_{n+1} = \frac{Y_1 + \hat{X}_4}{Y_1 + Y_2 + Y_3 + \hat{X}_4}$$

Sample Run

$m = 1000$

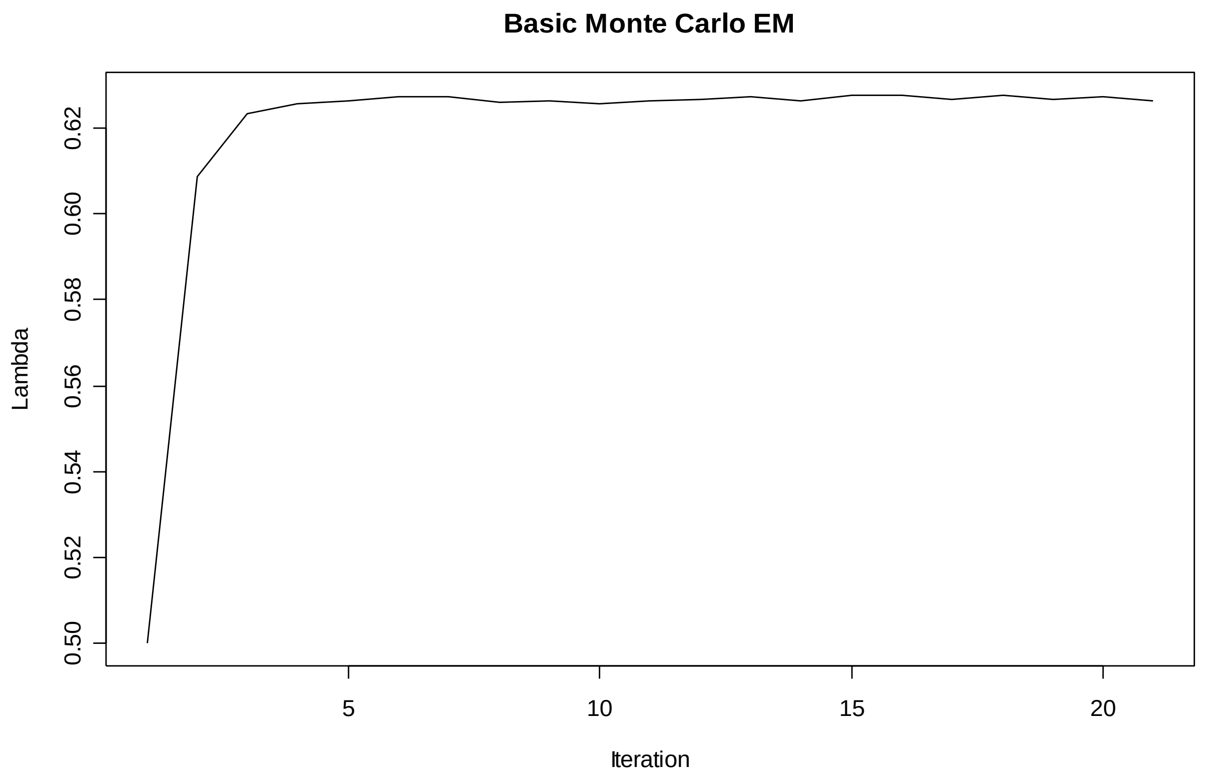
$\text{TOL} = 1 \times 10^{-6}$

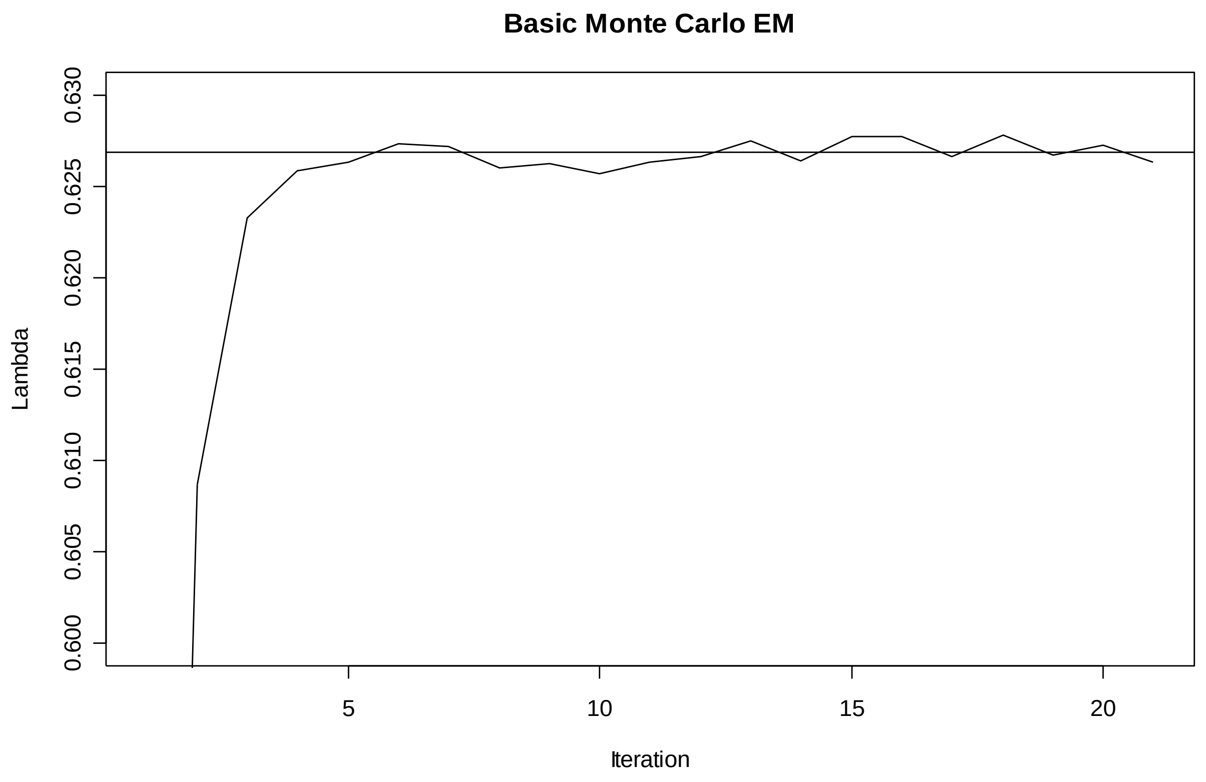
$n_{\max} = 20$

$\lambda_0 = 0.5$

Iteration	λ_n
0	0.5000000
1	0.6086751
2	0.6232812
3	0.6258296
4	0.6262932
5	0.6273085
6	0.6271988
7	0.6259806
8	0.6262454
9	0.6256674
10	0.6263483
11	0.6265943
12	0.6274692

and so up to $n_{\max} = 20$


Basic Monte Carlo EM
Lambda
0.50
0.52
0.54
0.56
0.58
0.60
0.62
Iteration
5
10
15
20


Basic Monte Carlo EM
Lambda
0.600
0.605
0.610
0.615
0.620
0.625
0.630
Iteration
5
10
15
20

Notes:

1) The number of imputations at each step can vary. Often you want to start with m_n relatively small and increase it towards the end.

2) Any sampling scheme can be used here, independent, importance, or MCMC.

3) Can be combined with exact calculation. Split the missing data into two pieces. Simulate one piece and calculate conditional expectations of second piece based on simulated data. (Hybrid MCEM)

3) In the M-step, a maximizer is not needed, just a value that increases $\hat{Q}\left(\theta|\theta_n\right)$.

4) Unlike normal EM, this procedure does not converge to a stationary point of the likelihood surface, but to a stochastic process centered at a stationary point of the likelihood surface.

5) This method is similar to the Stochastic EM method of Diebolt and Ip (1996). In there approach $m = 1$ but you keep the old samples in calculating $\hat{Q}\left(\theta|\theta_n\right)$.

Reweighted Monte Carlo EM (Irwin, 1995, Levine and Casella, 2001)

Sampling step

Sample $y_1,\ldots,y_m \sim f\left(Y \mid X,\theta^*\right)$

MC E-step

Update weights

$$w_n\left(y_i\right) = \frac{f\left(X, y_i \middle| \theta_n\right)}{f\left(X, y_i \middle| \theta^*\right)}$$

$$\tilde{w}_n\left(y_i\right) = \frac{w_n\left(y_i\right)}{\sum w_n\left(y_j\right)}$$

Let $\hat{Q}\left(\theta \middle| \theta_n\right) = \sum_{i=1}^{m} \tilde{w}_n\left(y_i\right) \log f\left(X, y_i \middle| \theta\right)$

$\hat{Q}\left(\theta \middle| \theta_n\right)$ is a MC estimate of $Q\left(\theta \middle| \theta_n\right)$

MC M-step

Set

$$\theta_{n+1} = \arg\sup \hat{Q}\left(\theta \middle| \theta_n\right)$$

Sample Run

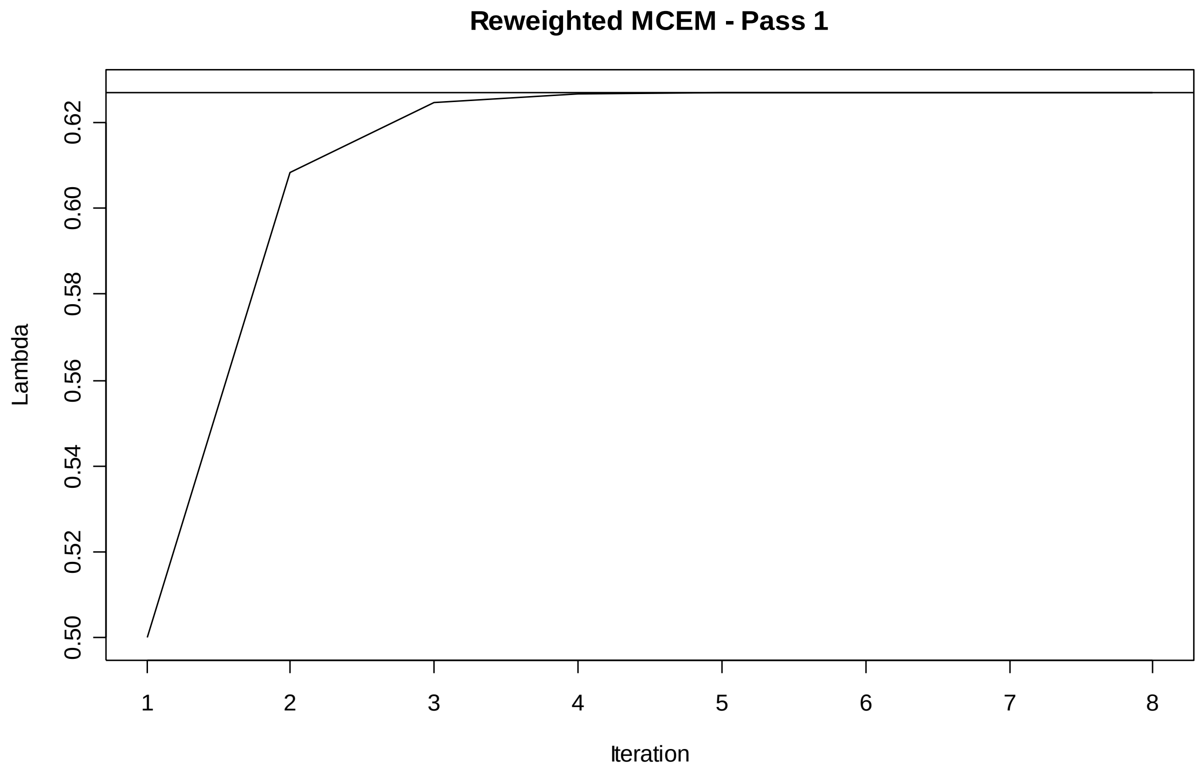
$m_1 = m_2$ = 1000, $m_3 = 10000$

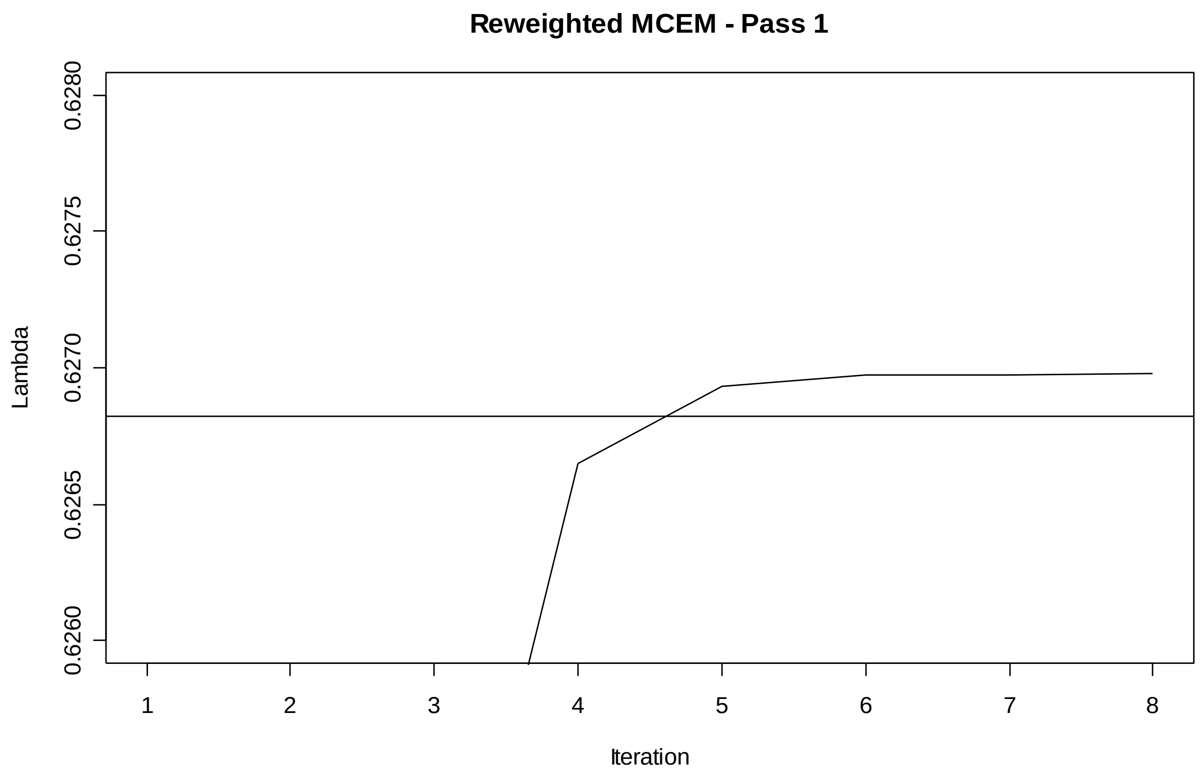
TOL = 1×10^{-6}

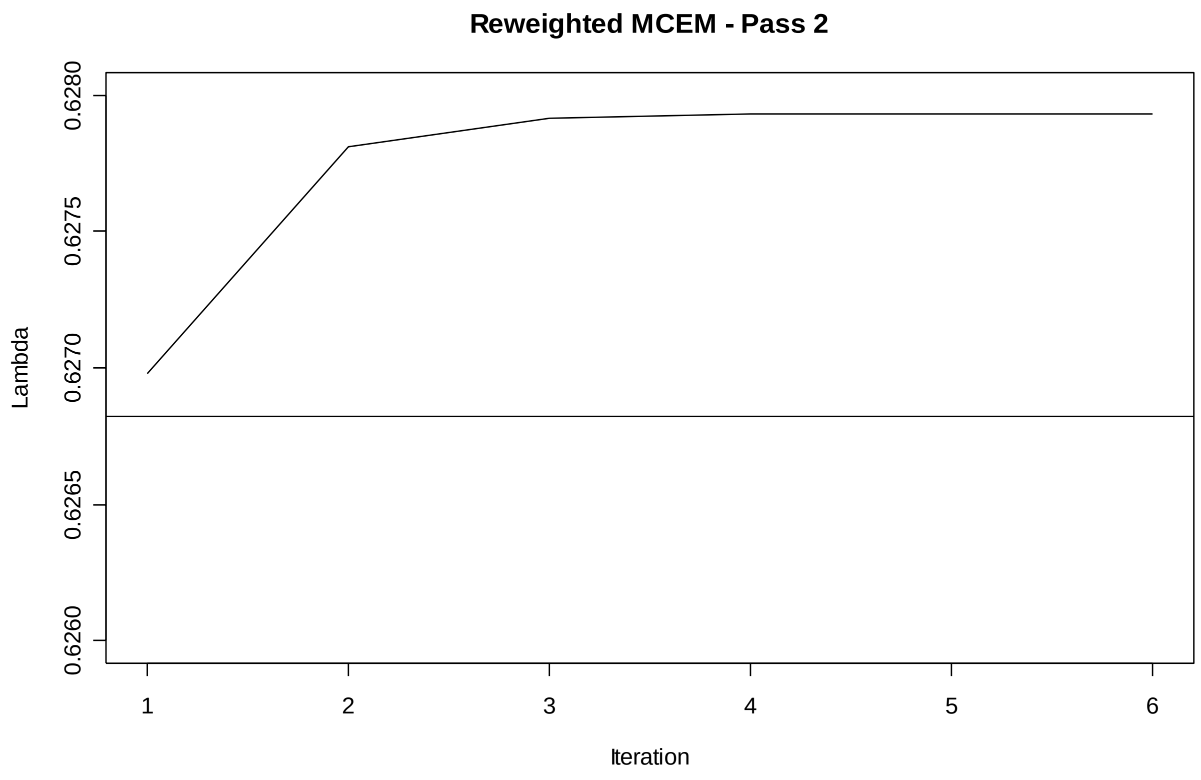
$n_{\max} = 20$

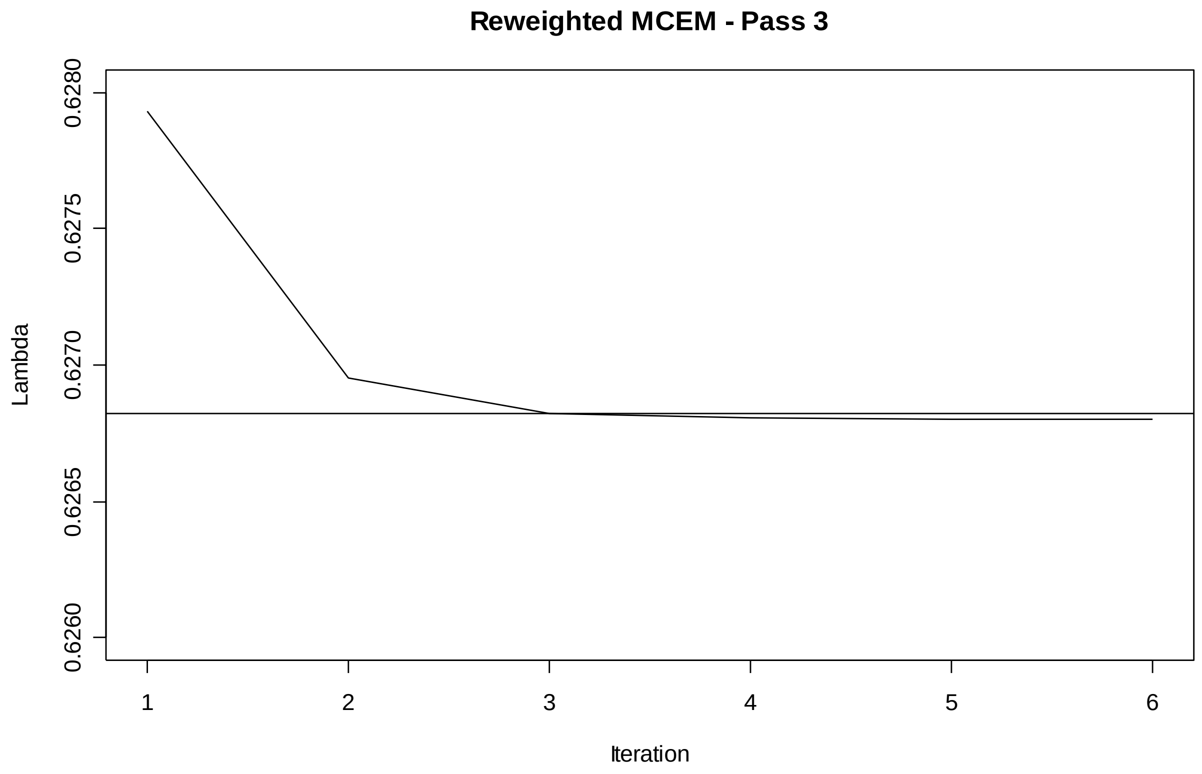
Pass 1: $\lambda_0 = \lambda^* = 0.5$

Pass 2 & 3: $\lambda_0 = \lambda^*$ = estimate from previous pass




Reweighted MCEM - Pass 1
Lambda
0.6260
0.6265
0.6270
0.6275
0.6280
Iteration
1
2
3
4
5
6
7
8


Reweighted MCEM - Pass 2
Lambda
0.6260
0.6265
0.6270
0.6275
0.6280
Iteration
1
2
3
4
5
6



Notes:

1) Usually you want to run this algorithm at least twice. Once to get in a region of the optimum, and the second one to tighten up the estimate.

2) The number of imputations at each step can vary. Often you want to start with m_1 to be moderate in size and increase it towards the end.

3) Particularly useful when simulation is time consuming. Reweighting is often much quicker and easier than sampling.

4) Any sampling scheme can be used here, independent, importance sampling, or MCMC. Hybrid MCEM also works here and is preferable if possible.

5) In the M-step, a maximizer is not needed, just a value that increases $\hat{Q}(\theta|\theta_n)$

6) The starting point in the first E-step θ_0 need not be the simulation point θ^*. In fact using different θ_0 allows for easy searching for multiple modes.

7) Weight properties

$$w_\theta(y) = \frac{p_\theta(y)}{p_{\theta^*}(y)}$$

$$E\left[w_\theta(Y)\middle|X,\theta^*\right] = \frac{g(X|\theta)}{g(X|\theta^*)} = l(\theta,\theta^*)$$

$$\hat{l}(\theta,\theta^*) = \frac{1}{m}\sum_{i=1}^{m} w_\theta(y_i)$$

As seen before $\hat{l}(\theta,\theta^*)$ is an unbiased estimator of $l(\theta,\theta^*)$.

8) Convergence properties

i) $\hat{l}\left(\theta_{n+1},\theta^*\right) \geq \hat{l}\left(\theta_n,\theta^*\right)$.

ii) procedure converges to stationary point of $\hat{l}\left(\theta,\theta^*\right)$.

9) Properties of MC MLE:

MLE: $\hat{\theta}$ maximizer of $g\left(X\middle|\theta\right)$ (or $l\left(\theta,\theta^*\right)$).

MC MLE: $\tilde{\theta}$ maximizer of $\hat{l}\left(\theta,\theta^*\right)$.

i) $\tilde{\theta}$ is a consistent estimator of $\hat{\theta}$.

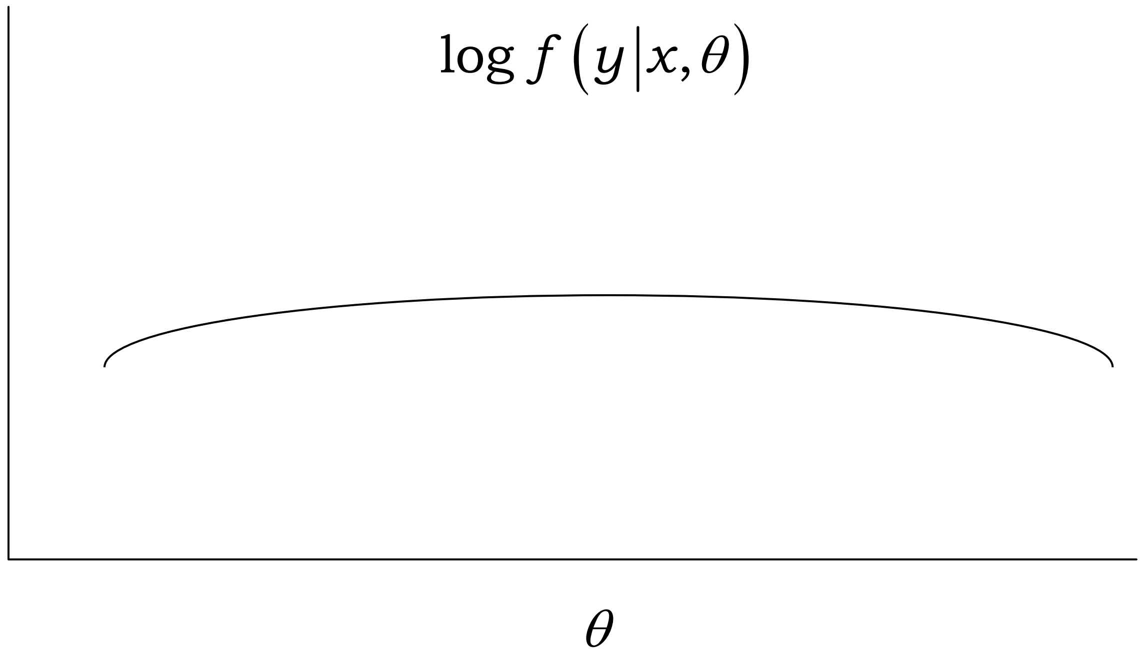
ii) Asympotically normal

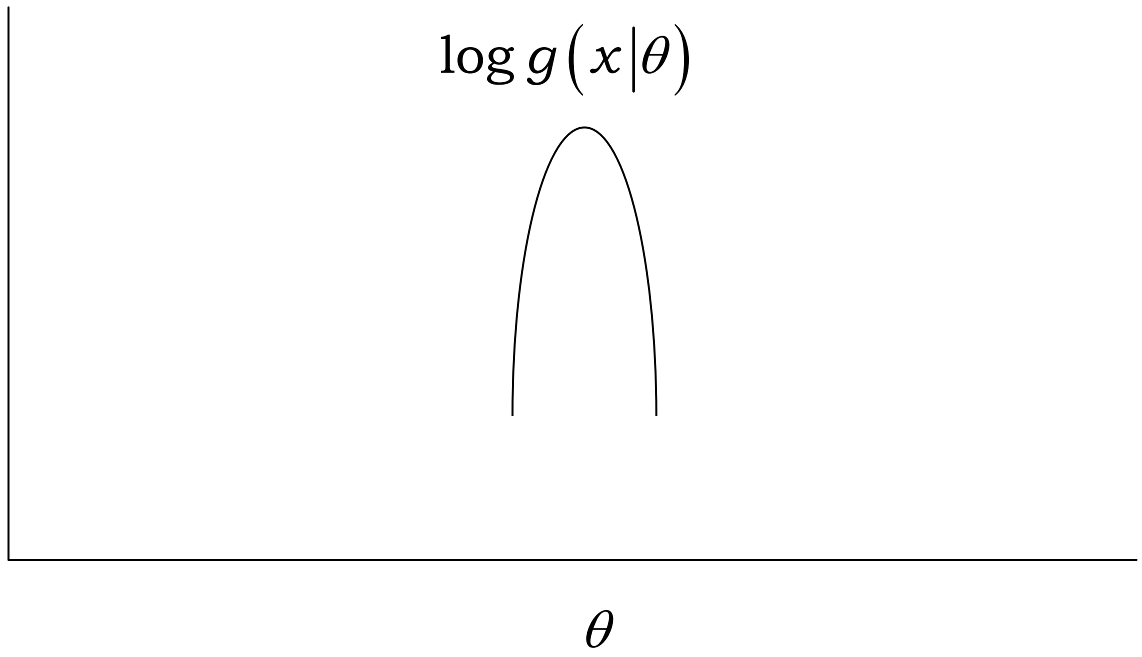
iii) $\operatorname{Var}\left(\tilde{\theta}\right) = \frac{1}{m} I^{-1}\left(\hat{\theta}\right) V\left(\hat{\theta}\right) I^{-1}\left(\hat{\theta}\right)$

where

$$V\left(\theta\right) = \operatorname{Var}\left(\frac{f\left(y\middle|x,\theta\right)}{f\left(y\middle|x,\theta^*\right)} \frac{D\log f\left(y\middle|x,\theta\right)}{D\theta}\middle|x,\theta^*\right)$$

$$I\left(\theta\right) = \frac{D^2 \log f\left(x\middle|\theta\right)}{D^2\theta}$$





This variance formula may only hold when the sampler generates independent realizations. I'm not sure about MCMC.

Standard Errors

Louis' formula can easily be approximated by Monte Carlo. For Reweighted MCEM, the estimator of the information matrix is

$$\begin{aligned}\hat{I}(\theta) = &-\sum \tilde{w}(y_i)\frac{D^2 \log f\left(x, y_i \middle| \theta\right)}{D^2\theta} \\ &- \sum \tilde{w}(y_i)\left(\frac{D \log f\left(x, y_i \middle| \theta\right)}{D\theta}\right)^2 \\ &+ \left(\sum \tilde{w}(y_i)\frac{D \log f\left(x, y_i \middle| \theta\right)}{D\theta}\right)^2\end{aligned}$$

Simulating Annealing

Another Monte Carlo approach for optimizing a function.

Approach for determining global optima, not local.

Useful for high dimensional, multimodal, complicated functions.

Examples where useful

1) Phylogentic tree reconstruction (Salter, 2000)

   Want to infer the evolutionary history based n observations (morphology, genetic, etc data) and describe it with a binary tree.

   Number of possible unrooted binary trees is

$$\prod_{i=1}^{n-1} (2i - 3)$$

   For $n = 30$, this is 8.69×10^{36}.

2) Traveling Salesman Problem

   A salesman has n towns to be visited. Wants to minimize the traveling distance and finish at his starting point

   This is known to be an NP complete problem

The name comes from condensed matter physics. Annealing is a thermal process for obtaining low-energy states of a solid in a heat path. The procedure has two steps

1) Raise the temperature of the heat bath enough for the solid (metal) to melt.

2) Decrease the temperature slowly to near zero so that the particles in the system can arrange themselves in the ground state of the solid (i.e. crystallize).

Problem: Want to minimize a function $h(x)$ or maximize a function $-h(x)$. This is equivalent to maximizing the function $\exp\left(-h(x)/T\right)$ for any given temperature *T*.

Method:

1) Let $T_1 > T_2 > T_3 > \dots$ be a sequence of temperatures in which T_1 is relatively large and $\lim_{k\to\infty} T_k = 0$.

2) At each temperature T_k run N_k steps of a M-H sampler with stationary distribution

$$\pi_k(x) \propto \exp\left(-h(x)/T_k\right)$$

Pass the final configuration of x to the next iteration and set k to $k + 1$.

Why does SA work:

For any system in which

$$\int \exp\left(-h(x)/T\right) dx < 0$$

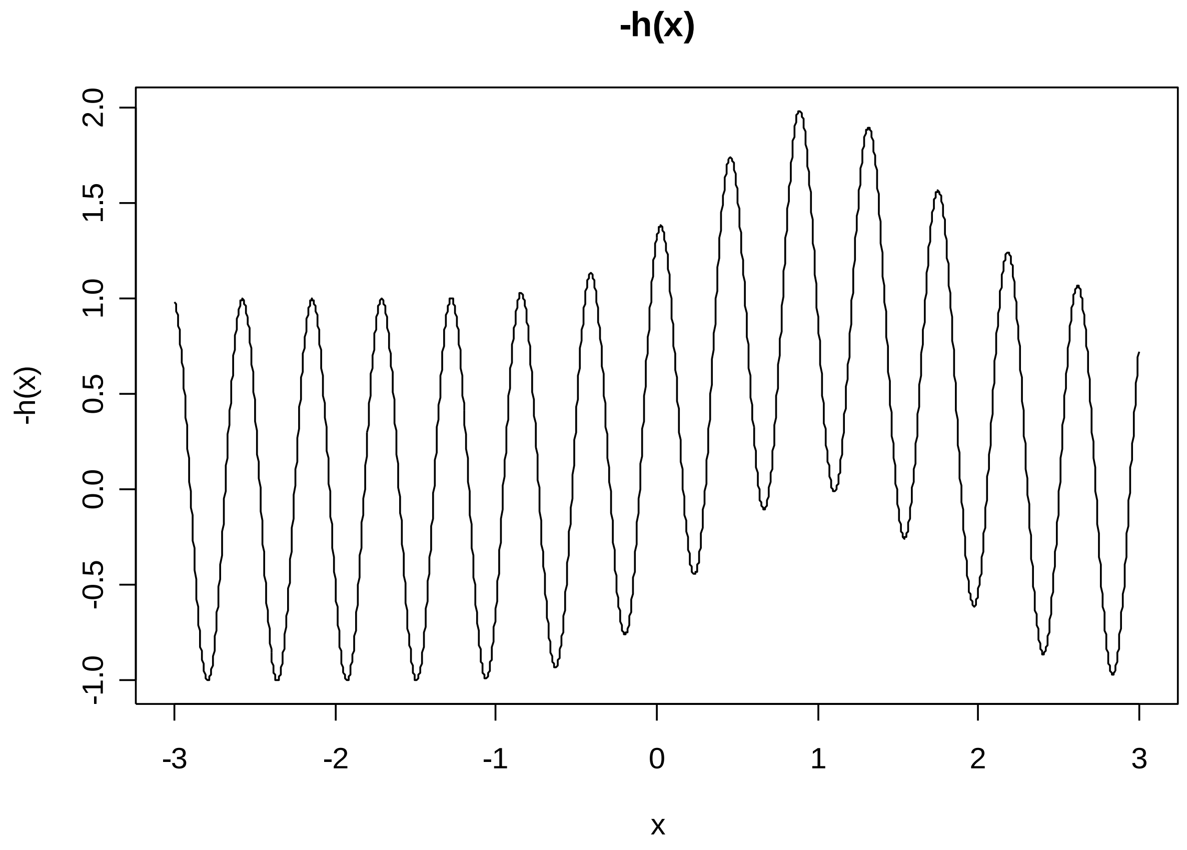
for all T, distribution π_k, as k increases puts more and more of its probability mass (converging to 1) into the vicinity of the global maximum of $-h(x)$.

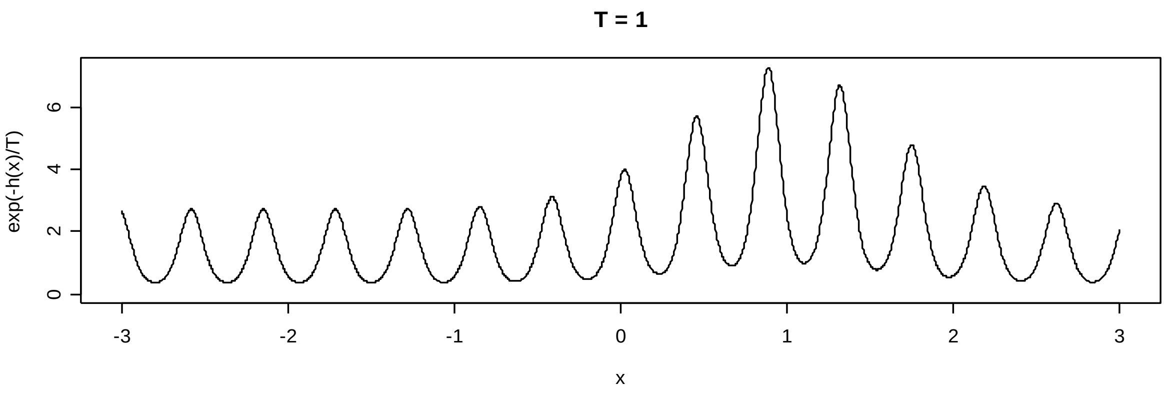
Hence a sample from π_k will almost certainly be in the vicinity of the global optimum of $h(x)$ when T_k is close to 0.

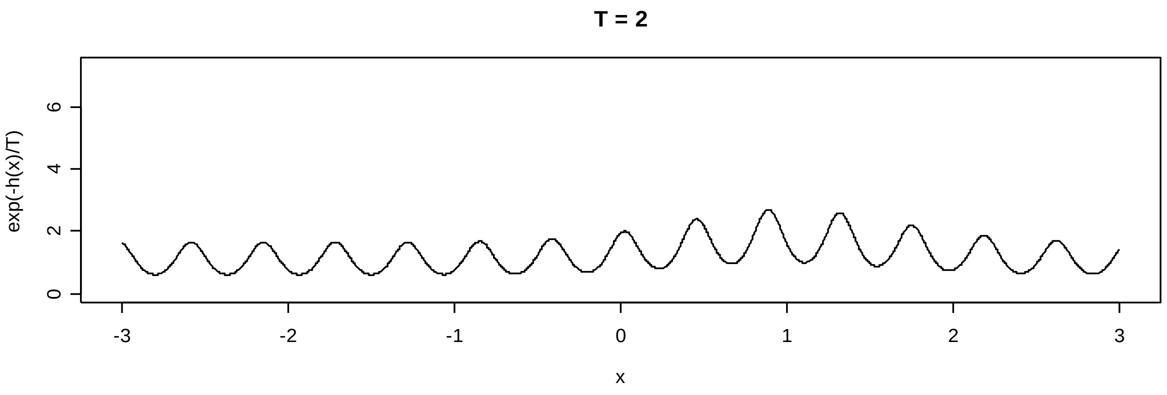
And as $T \to \infty$, $\exp\left(-h(x)/T\right)$ goes to a Uniform distribution

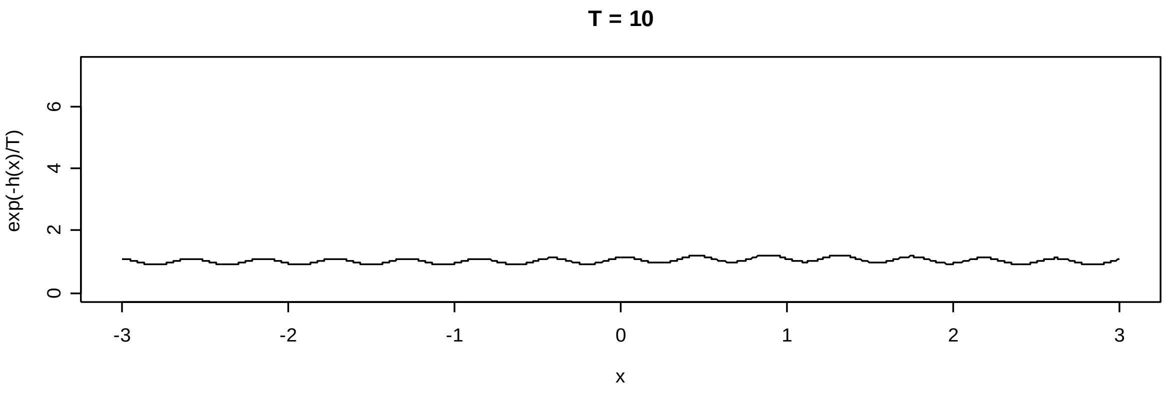
Example function

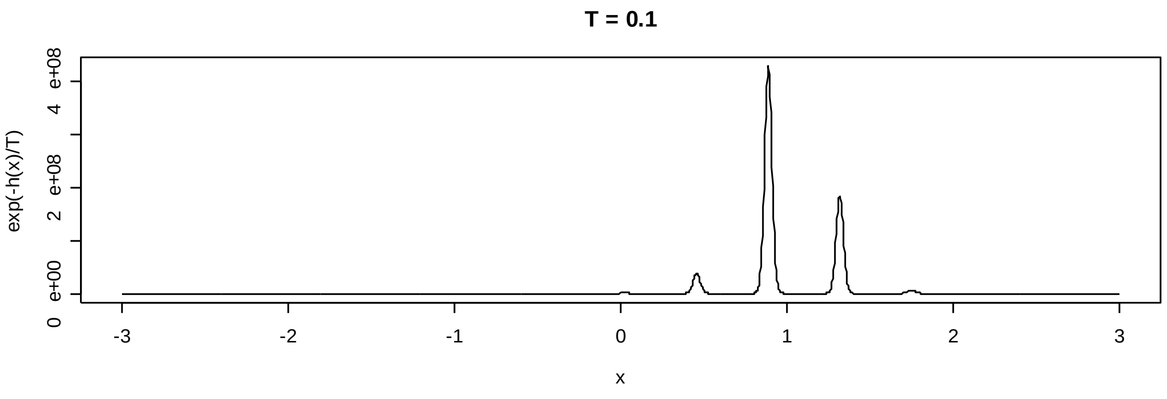
$$-h(x) = \cos(14.5x - 0.3) + \exp\left(-(x-1)^2\right)$$




T = 1
exp(-h(x)/T)
0
2
4
6
-3
-2
-1
0
1
2
3
x


T = 2
exp(-h(x)/T)
0
2
4
6
-3
-2
-1
0
1
2
3
x


T = 10
exp(-h(x)/T)
0
2
4
6
-3
-2
-1
0
1
2
3
x


T = 0.1
exp(-h(x)/T)
0 e+00
2 e+08
4 e+08
-3
-2
-1
0
1
2
3
x

In addition, as we have seen (sort of) M-H algorithms will tend to accept more likely proposals. They definitely will with symmetric proposal distributions.

So SA will tend to move to points when $-h(x)$ increases, but can move to points where $-h(x)$ is lower so it can move out of a local, but lower, mode, into a better mode.

It can be shown that the global optimum can be reached with SA with probability 1 if the temperature variable T_k decreases sufficiently slowly enough ($O\left(\log L_k^{-1}\right)$) where $L_k = N_1 + \ldots + N_k$.

This slow temperature decrease is required to make sure the chain has run long enough to get into the dominant mode.

The large T_k allow for the space to be completely sampled and the small T_k focus on the global optimum.

In practice cooling schedules like these aren't used. Linear or exponential is more common as these give more manageable computational burden.

However you are no longer guaranteed to find the global optima. However you usually find a reasonable answer.