UNIVERSITY OF
COPENHAGEN

Likelihood Optimization and the EM Algorithm

Computational Statistics

Johan Larsson

Department of Mathematical Sciences, University of Copenhagen

March 31, 2026

Last Time

Testing

Systematic approach to find errors in code.

Debugging

Finding and fixing bugs in code.

Today

Optimization for Multinomial Models

We will look at optimization for multinomial models, using an example on Peppered Moths.

We will introduce two different optimization methods for constrained problems:

- General optimization using extended-value functions and zero-order methods.
- Constrained optimization using the barrier method.

The EM Algorithm

We will introduce the EM algorithm. A useful method for likelihood optimization in the presence of missing data or latent variables.

Multinomial Models

The Peppered Moth

Alleles
C, Ci, T with frequencies p_C, p_I, p_T and

$$p_C + p_I + p_T = 1.$$

Genotypes
CC, CI, CT, II, IT, TT

Phenotypes
Black, mottled, light-colored

Table 1: Genotype to Phenotype Map

	C	I	T
C	Black	Black	Black
I	Black	Mottled	Mottled
T	Black	Mottled	Light-colored

According to the Hardy–Weinberg equilibrium, the genotype frequencies are

$$p_C^2, 2p_Cp_I, 2p_Cp_T, p_I^2, 2p_Ip_T, p_T^2.$$

The complete multinomial log-likelihood is

$$\begin{aligned} &2n_{CC}\log(p_C) + n_{CI}\log(2p_Cp_I) + n_{CT}\log(2p_Cp_T) \\ &+ 2n_{II}\log(p_I) + n_{IT}\log(2p_Ip_T) + 2n_{TT}\log(p_T), \end{aligned}$$

We only observe (n_C, n_I, n_T), where

$$n = \underbrace{n_{CC} + n_{CI} + n_{CT}}_{=n_C} + \underbrace{n_{IT} + n_{II}}_{=n_I} + \underbrace{n_{TT}}_{=n_T}.$$

As a specific data example we have the observation $n_C = 85$, $n_I = 196$, and $n_T = 341$.

Multinomial Cell Collapsing

The Peppered Moth example is an example of *cell collapsing* in a multinomial model.

In general, let $A_1 \cup ... \cup A_{K_0} = \{1, ..., K\}$ be a partition and let

$$M : \mathbb{N}_0^K \to \mathbb{N}_0^{K_0}$$

be the map given by

$$M((n_1, ..., n_K))_j = \sum_{i \in A_j} n_i.$$

If $Y \sim \text{Mult}(p, n)$ with $p = (p_1, \dots, p_K)$ then

$$X = M(Y) \sim \text{Mult}(M(p), n).$$

For the peppered moths, $K = 6$ corresponding to the six genotypes, $K_0 = 3$ and the partition corresponding to the phenotypes is

$$\{1, 2, 3\} \cup \{4, 5\} \cup \{6\} = \{1, \dots, 6\},$$

and

$$M(n_1, \dots, n_6) = (n_1 + n_2 + n_3, n_4 + n_5, n_6).$$

In terms of the (p_C, p_I) parametrization, $p_T = 1 - p_C - p_I$ and

$$p = (p_C^2, 2p_Cp_I, 2p_Cp_T, p_I^2, 2p_Ip_T, p_T^2).$$

Hence

$$M(p) = (p_C^2 + 2p_Cp_I + 2p_Cp_T, p_I^2 + 2p_Ip_T, p_T^2).$$

The log-likelihood is

$$\begin{aligned}\ell(p_C, p_I) = n_C \log(p_C^2 + 2p_Cp_I + 2p_Cp_T) \\ + n_I \log(p_I^2 + 2p_Ip_T) + n_T \log(p_T^2)\end{aligned}$$

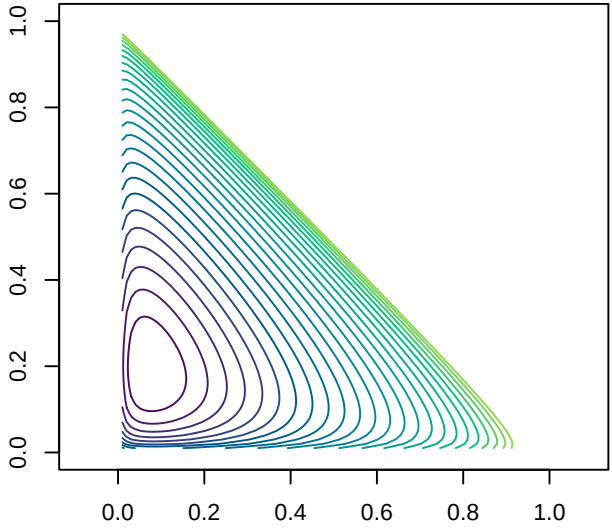
Standard Form

This is an optimization problem of the following standard form:

$$\begin{aligned} \underset{p_C, p_I}{\text{minimize}} \quad & -\ell(p_C, p_I) \\ \text{subject to} \quad & p_C + p_C - 1 \leq 0 \\ & -p_C \leq 0 \\ & -p_I \leq 0 \\ & p_I - 1 \leq 0 \\ & p_C - 1 \leq 0. \end{aligned}$$

Convex?
Yes, because the negative log-likelihood is convex and so are the constraints (affine).

Can We Solve it Using Gradient Descent or Newton's Method?
No (not directly), since the problem is constrained.


0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

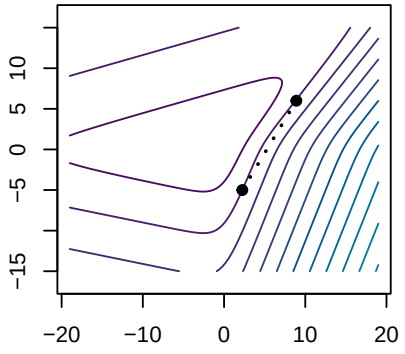
Change of Variables

Let

$$p_j = \frac{\exp(\theta_j)}{\sum_{i=1}^{K} \exp(\theta_i)}.$$

Seems like a good idea! Constraints are automatically satisfied.

But is it Convex?
No, f is no longer convex.



So, we cannot use our existing toolbox. What to do?

We will try two different options.

General Optimization
Define **extended-value extension**, and use general (zero-order) optimization.

Constrained Optimization (Barrier Method)
Use the barrier method to directly solve the **constrained** optimization problem.

Extended-Value Extension

If f is convex, its extended-value extension is

$$\tilde{f}(x) = \begin{cases} f(x), & x \in \operatorname{dom} f, \\ +\infty, & \text{otherwise.} \end{cases}$$

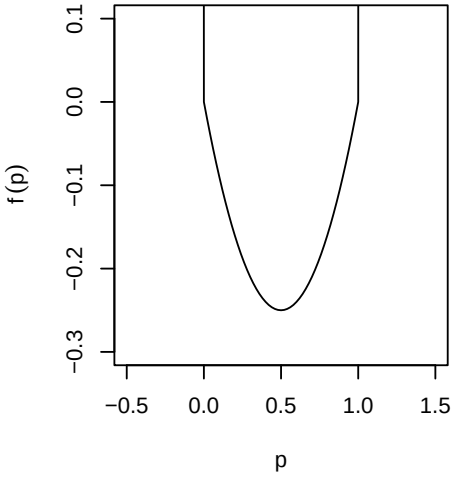
Example

Consider the function

$$f(p) = p(p-1), \quad p \in [0,1].$$

Its extended-value extension is

$$\tilde{f}(p) = \begin{cases} p(p-1), & p \in [0,1], \\ +\infty, & \text{otherwise.} \end{cases}$$



We can code a problem-specific (extended-value) version of the negative log-likelihood.

```
neg_loglik_pep <- function(par, x) {
  pC <- par[1]
  pI <- par[2]
  pT <- 1 - pC - pI

  if (pC > 1 || pC < 0 || pI > 1 || pI < 0 || pT < 0) {
    return(Inf)
  }

  p_dar <- pC^2 + 2 * pC * pI + 2 * pC * pT
  p_mot <- pI^2 + 2 * pI * pT
  p_lig <- pT^2

  -(x[1] * log(p_dar) + x[2] * log(p_mot) + x[3] * log(p_lig))
}
```

optim()

R provides a general-purpose optimization function `optim()`.

Supports several optimization methods, including:

- Nelder-Mead (default)
- BFGS (quasi-Newton)
- CG (conjugate gradient)
- L-BFGS-B (box-constrained quasi-Newton)
- SANN (simulated annealing)

Can supply gradient, but also compute it numerically (default).

optim() Example

```r
obj <- function(x) {
  (x[1] - 1)^2 + (x[2] - 2.5)^2
}
optim(c(0, 0), obj, method = "BFGS")
```

```
$par
[1] 1.0 2.5

$value
[1] 4.382e-26

$counts
function gradient
       9        3

$convergence
[1] 0
```

```
x <- c(85, 196, 341)
x0 <- c(1 / 3, 1 / 3)
optim(x0, neg_loglik_pep, x = x)
```

```
$par
[1] 0.07082 0.18879

$value
[1] 600.5

$counts
function gradient
      65       NA

$convergence
[1] 0
```

Feasible Starting Point

Some thought has to go into the initial parameter choice.

```
optim(
  c(0, 0),
  neg_loglik_pep,
  x = x
)
```

```
Error in optim(c(0, 0),
   neg_loglik_pep, x = x): function
   cannot be evaluated at initial
    parameters
```

Barrier Function

Another option is to use the barrier method.

Transform

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && g_i(x) \leq 0, \qquad i = 1, \dots, m. \end{aligned}$$

into

$$\text{minimize} \quad tf(x) + \phi(x)$$

with the barrier function

$$\phi(z) = -\sum_{i=1}^{m} \log(-g_i(z)).$$

For t small, the barrier term dominates and the solution is far from the boundary.

Increasing t puts more weight on the original objective and the solution approaches the constrained optimum.

Example

Let's assume we are solving

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) = (x_1 - 2)^2 + (x_2 - 1)^2 \\ \text{subject to} \quad & x_1 - 1 \leq 0. \end{aligned}$$

The unconstrained optimum would be at $(2, 1)$, but this is not feasible.

Using the barrier method, we solve a sequence of unconstrained problems

$$\underset{x}{\text{minimize}} \quad tf(x) - \log(1 - x_1)$$

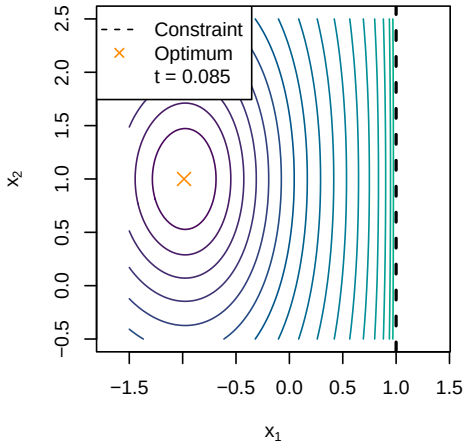


Figure 3: Contour plots of the barrier-penalized objective.

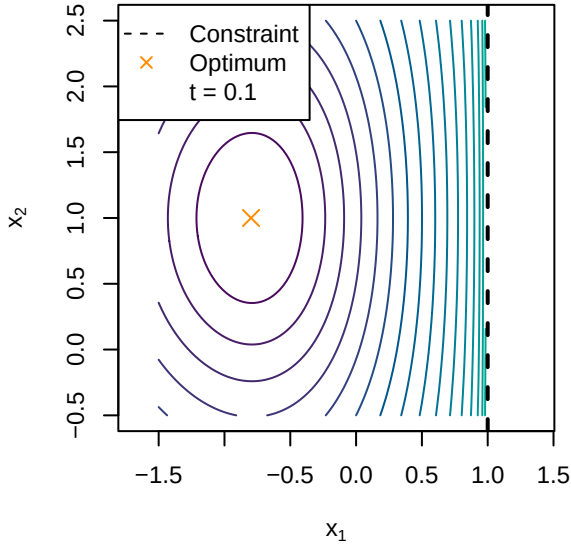


Figure 4: Contour plots of the barrier-penalized objective.

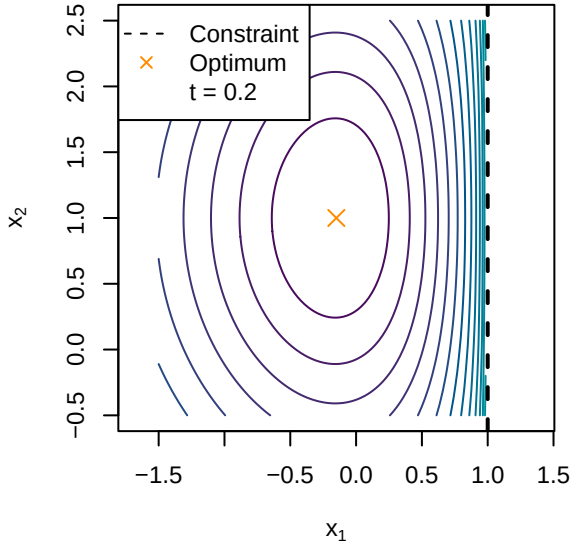


Figure 5: Contour plots of the barrier-penalized objective.

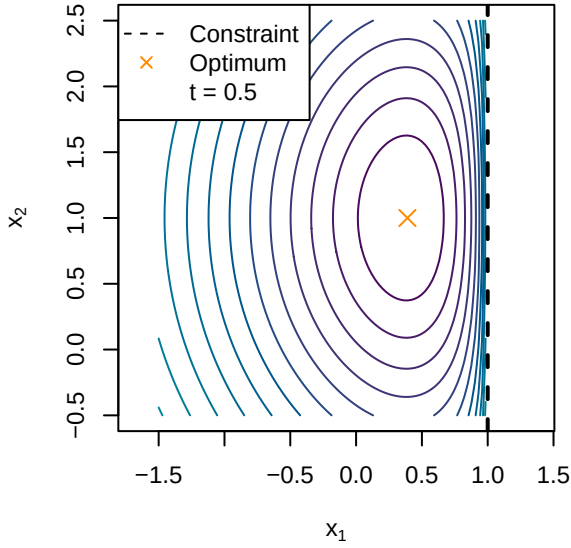


Figure 6: Contour plots of the barrier-penalized objective.

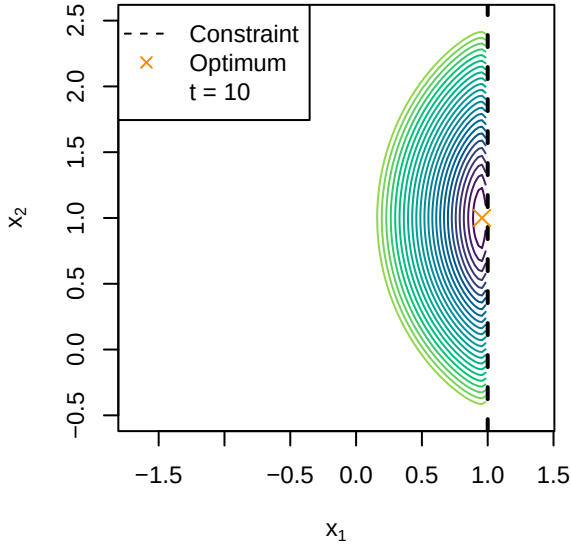


Figure 7: Contour plots of the barrier-penalized objective.

Barrier Method Algorithm

Algorithm 1: Barrier Method

Data: $t > 0$, x strictly feasible, $\mu > 1$,, $\varepsilon > 0$, m number of constraints

repeat

$x \leftarrow \operatorname{argmin}_x \, tf(x) + \phi(x)$;

$t \leftarrow \mu t$;

until $m/t < \varepsilon$;

return x

Guarantees convergence to a ε-suboptimal solution.

constrOptim()

The barrier method can be used via `constrOptim()`.

Solves problems with affine/linear inequality constraints of the form

$$Ax \succeq b$$

or, in terms of arguments: `ui %*% theta >= ci`.

Moths Example

$$A = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \\ -1 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ -1 \\ 0 \\ -1 \\ -1 \end{bmatrix}.$$

Optimization Problem

$$\begin{aligned} \underset{p}{\text{minimize}} \quad & -\ell(p) \\ \text{subject to} \quad & Ap \succeq b. \end{aligned}$$

with $p = (p_C, p_I)$.

Not standard form—but what `constrOptim()` expects.

constrOptim()

```
A <- rbind(
  c(1, 0),
  c(-1, 0),
  c(0, 1),
  c(0, -1),
  c(-1, -1)
)

b <- c(0, -1, 0, -1, -1)

constrOptim(x0, neg_loglik_pep, NULL, ui = A, ci = b, x = x)$par
```

```
[1] 0.07082 0.18879
```

Still Requires Feasible Initial Point

```
constrOptim(
  c(0.0, 0.3),
  neg_loglik_pep,
  NULL,
  ui = A,
  ci = b,
  x = x
)
```

```
Error in constrOptim(c(0, 0.3), neg_loglik_pep, NULL, ui = A, ci = b, : initi
    value is not in the interior of the feasible region
```

The barrier method is the basis of the interior-point method.

Phase I: Initialization
Find a feasible point $x^{(0)}$ by solving the optimization problem

$$\begin{aligned} &\underset{(x,s)}{\text{minimize}} && s \\ &\text{subject to} && g_i(x) \leq s, \qquad i = 1, \dots, m. \end{aligned}$$

If $s^* \leq 0$, then $x^* = x^{(0)}$ is feasible.

Easy in the peppered moths case! But this is not always the case.

Phase II: Barrier Method
Use a barrier method to solve the constrained problem, starting at $x^{(0)}$.

Multinomial Conditional Distributions

Distribution of $Y_{A_j} = (Y_i)_{i \in A_j}$ conditional on X can be found too:

$$Y_{A_j} \mid X = x \sim \text{Mult}\left(\frac{p_{A_j}}{M(p)_j}, x_j\right).$$

Expected Genotype Frequencies

Hence for $k \in A_j$,

$$\text{E}(Y_k \mid X = x) = \frac{x_j p_k}{M(p)_j}.$$

```
group <- c(1, 1, 1, 2, 2, 3)
p <- prob(moth_optim$par)
x[group] * p / M(p, group)[group]
```

```
[1]   3.12  16.63  65.25  22.15 173.85 341.00
```

The EM Algorithm

Goal

What we want is the MLE

$$\hat{\theta} = \arg\max_{\theta} \ell(\theta \mid X)$$

with $\ell(\theta \mid X) = \log p(X \mid \theta)$.

Problem

The likelihood requires summing/integrating over unobserved variables Z:

$$p(X \mid \theta) = \sum_{Z} p(X, Z \mid \theta).$$

This is often computationally infeasible.

Key Idea

Work instead with the complete-data likelihood:

$$p(X, Z \mid \theta),$$

which is often easier to handle.

The Algorithm

E-step

Compute expected complete log-likelihood:

$$Q(\theta \mid \theta^{(t)}) = \mathrm{E}_{Z|X,\theta^{(t)}} \left(\log p(X, Z \mid \theta) \right)$$

M-Step

Maximize expected complete log-likelihood:

$$\theta^{(t+1)} = \arg\max_{\theta} Q(\theta \mid \theta^{(t)}).$$

Repeat

Until convergence.

Why EM Works

Key Inequality

$$\ell(\theta) = \log p(X \mid \theta) \geq \mathrm{E}_{q(Z)}[\log p(X, Z \mid \theta)] - \mathrm{E}_{q(Z)}[\log q(Z)].$$

E-Step

Choose

$$q^{(t)}(Z) = p(Z \mid X, \theta^{(t)}).$$

Then inequality is tight, since

$$\mathrm{KL}(q^{(t)} || p(Z \mid X, \theta^{(t)})) = 0.$$

M-Step

Update θ by maximizing the bound. This guarantees non-decreasing log-likelihood:

$$\ell(\theta^{(t+1)}) \geq \ell(\theta^{(t)}).$$

From the Lower Bound to Q

Variational Bound

$$F(q, \theta) = \mathrm{E}_q \left(\log p(X, Z \mid \theta) \right) - \mathrm{E}_q \left[\log q(Z) \right].$$

With

$$q^{(t)}(Z) = p(Z \mid X, \theta^{(t)})$$

we have

$$F(q^{(t)}, \theta) = Q(\theta \mid \theta^{(t)}) + H(q^{(t)}),$$

where $H(q^{(t)})$ is entropy, independent of θ.

Consequence

M-step:

$$\theta^{(t+1)} = \arg\max_{\theta} Q(\theta \mid \theta^{(t)}).$$

So EM maximizes the variational lower bound.

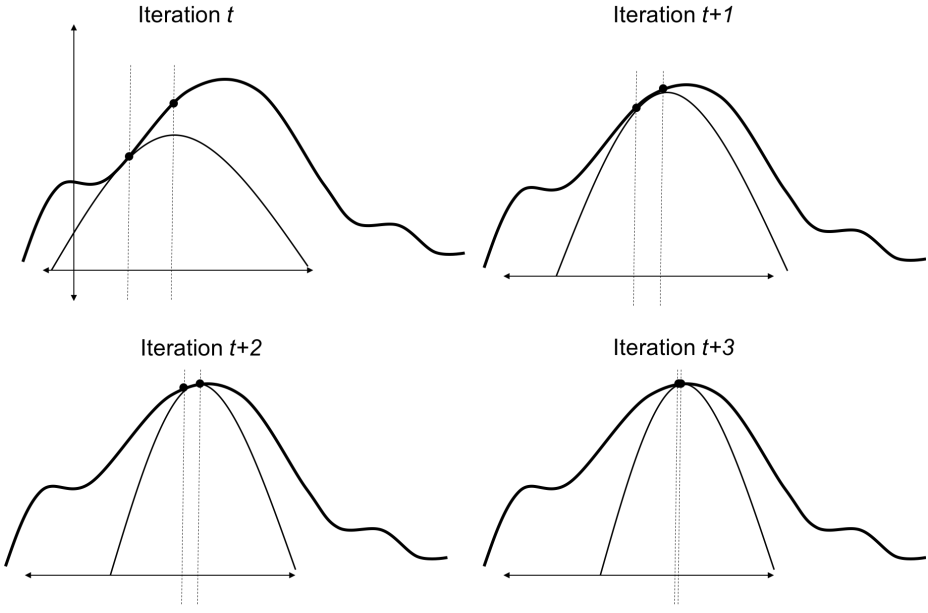


Figure 8: Four iterations of the EM algorithm.

EM Example: Two Hidden Coins

Setup

- Two coins, A and B, unknown biases θ_A and θ_B.
- Each experiment: pick a coin at random (latent variable Z_i) and flip it 10 times.
- Observed data: X_i: number of heads in each experiment.

Goal

Estimate θ_A and θ_B (MLE) using EM.

Complete Likelihood

If we knew which coin was used in each experiment, the complete-data likelihood would be

$$p(X, Z \mid \theta) = \sum_i \left(\mathbf{1}_{Z_i=A} \text{Binomial}(X_i \mid 10, \theta_A) + \mathbf{1}_{Z_i=B} \text{Binomial}(X_i \mid 10, \theta_B) \right).$$

The E-Step: The Q-Function

The log-likelihood of the complete data (for a single experiment) is

$$\log p(X_i, Z_i \mid \theta) = \mathbf{1}_{Z_i=A} \log \mathsf{Bin}(X_i \mid 10, \theta_A) + \mathbf{1}_{Z_i=B} \log \mathsf{Bin}(X_i \mid 10, \theta_B).$$

Take expection over Z_i given X_i and current parameters:

$$Q(\theta \mid \theta^{(t)}) = \sum_i \gamma_i^{(t)} \log \mathsf{Bin}(X_i \mid 10, \theta_A) + \sum_i (1 - \gamma_i^{(t)}) \log \mathsf{Bin}(X_i \mid 10, \theta_B).$$

$$\begin{aligned}
\gamma_i^{(t)} &= \mathrm{E}_Z \left(\mathbf{1}_{Z_i=A} \mid X_i, \theta_A^{(t)}, \theta_B^{(t)} \right) \\
&= \Pr(Z_i = A \mid X_i, \theta_A^{(t)}, \theta_B^{(t)}) \\
&= \frac{\mathsf{Binomial}(X_i \mid 10, \theta_A^{(t)})}{\mathsf{Binomial}(X_i \mid 10, \theta_A^{(t)}) + \mathsf{Binomial}(X_i \mid 10, \theta_B^{(t)})}.
\end{aligned}$$

This called the **responsibility** of coin A for experiment i.

Table 2: Example After One Iteration

Experiment	Heads	γ
1	7	0.78
2	4	0.35
3	9	0.90

M-Step: Update Coin Biases

Maximizing Q with respect to θ_A and θ_B gives

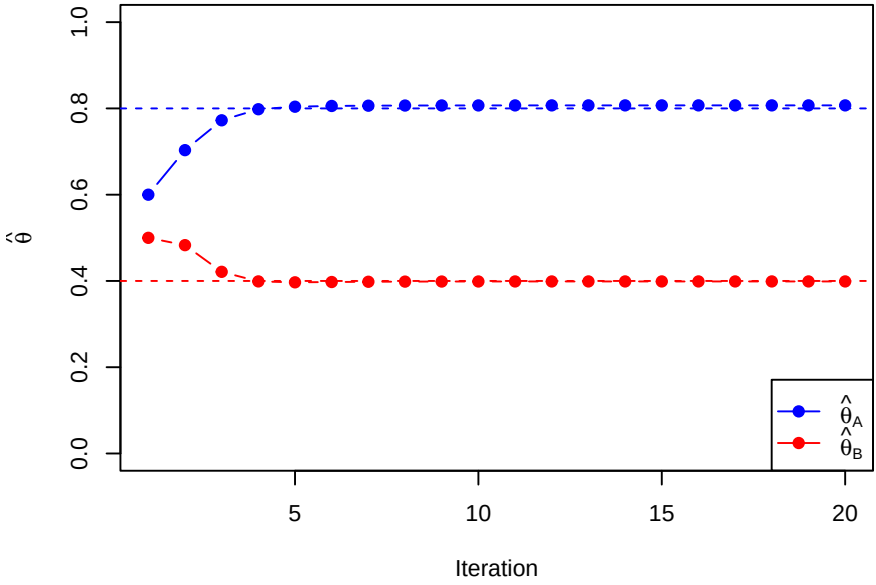
$$\theta_A^{(t+1)} = \frac{\sum_i \gamma_i^{(t)} X_i}{\sum_i 10 \cdot \gamma_i^{(t)}}, \quad \theta_B^{(t+1)} = \frac{\sum_i (1-\gamma_i^{(t)}) X_i}{\sum_i 10 \cdot (1-\gamma_i^{(t)})}.$$

Intuition

Compute expected # of heads for each coin and divide by expected # of flips.

Table 3: Example After One Iteration

Coin	Updated θ
A	0.80
B	0.42


1.0
0.8
0.6
0.4
0.2
0.0
θ̂
5
10
15
20
Iteration
θ̂A
θ̂B

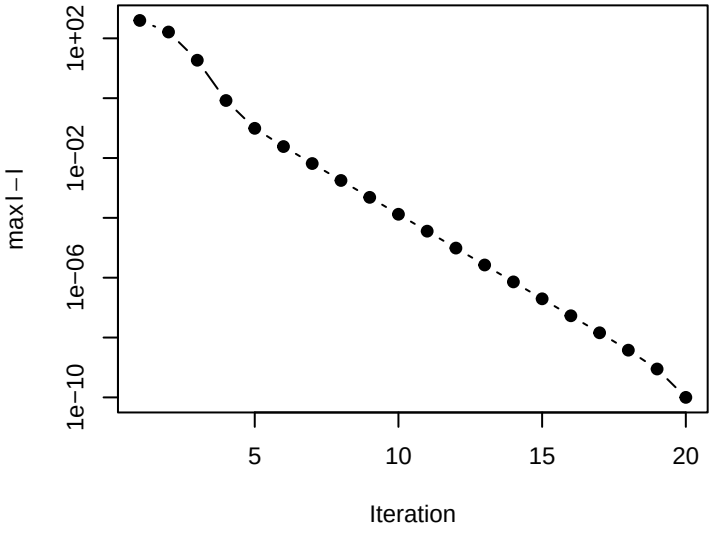


Figure 10: Convergence of the log-likelihood in the two-coin EM example.

Connection to Q-Function

Recall:

$$Q(\theta \mid \theta^{(t)}) = \mathrm{E}_{Z|X,\theta^{(t)}}[\log p(X, Z \mid \theta)]$$

In the coin-flip example:

$$Q(\theta_A, \theta_B \mid \theta_A^{(t)}, \theta_B^{(t)}) = \sum_i \gamma_i^{(t)} \log \mathsf{Bin}(X_i \mid 10, \theta_A) + \sum_i (1 - \gamma_i^{(t)}) \log \mathsf{Bin}(X_i \mid 10, \theta_B)$$

- **E-step:** compute $\gamma_i^{(t)}$, the expected values (responsibilities) of the latent variables.
- **M-step:** maximize Q with respect to θ_A and θ_B: weighted averages of heads.

Intuition: Q is the expected complete-data log-likelihood; EM alternates between **expectation** (filling in latent variables) and **maximization** (updating parameters).

Summary

Likelihood Optimization

Peppered moths example is simple and the log-likelihood for the observed data can easily be computed.

Constrained optimization can be solved using the barrier method.

The EM Algorithm

The EM algorithm is a general approach to likelihood optimization in the presence of missing data or latent variables.

Each iteration is guaranteed to increase the observed data log-likelihood.

Exercise: Absolute-Value Constraints

Solve the following optimization problem:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \frac{1}{2}(x-2)^2 \\ \text{subject to} \quad & |x| \leq 1. \end{aligned}$$

Steps

1. Rewrite the problem in standard form.
2. Write up the constraints in the form `ui %*% theta >= ci`.
3. Use `constrOptim()` with `ui` and `ci` to solve the problem. You need to provide the objective and gradient functions too.

Next Time

We will continue with more examples of the EM algorithm.

We will show how to solve the peppered moths example using EM.

We will also talk about convergence and the **generalized** EM algorithm.