

The EM algorithm

Yue Jiang

Duke University

A disclaimer

Today's (and next time's) lectures are introductory surface level treatments of the EM algorithm. We focus on applications and use cases -- there are no theoretical results presented (even for important subjects like variance estimation).

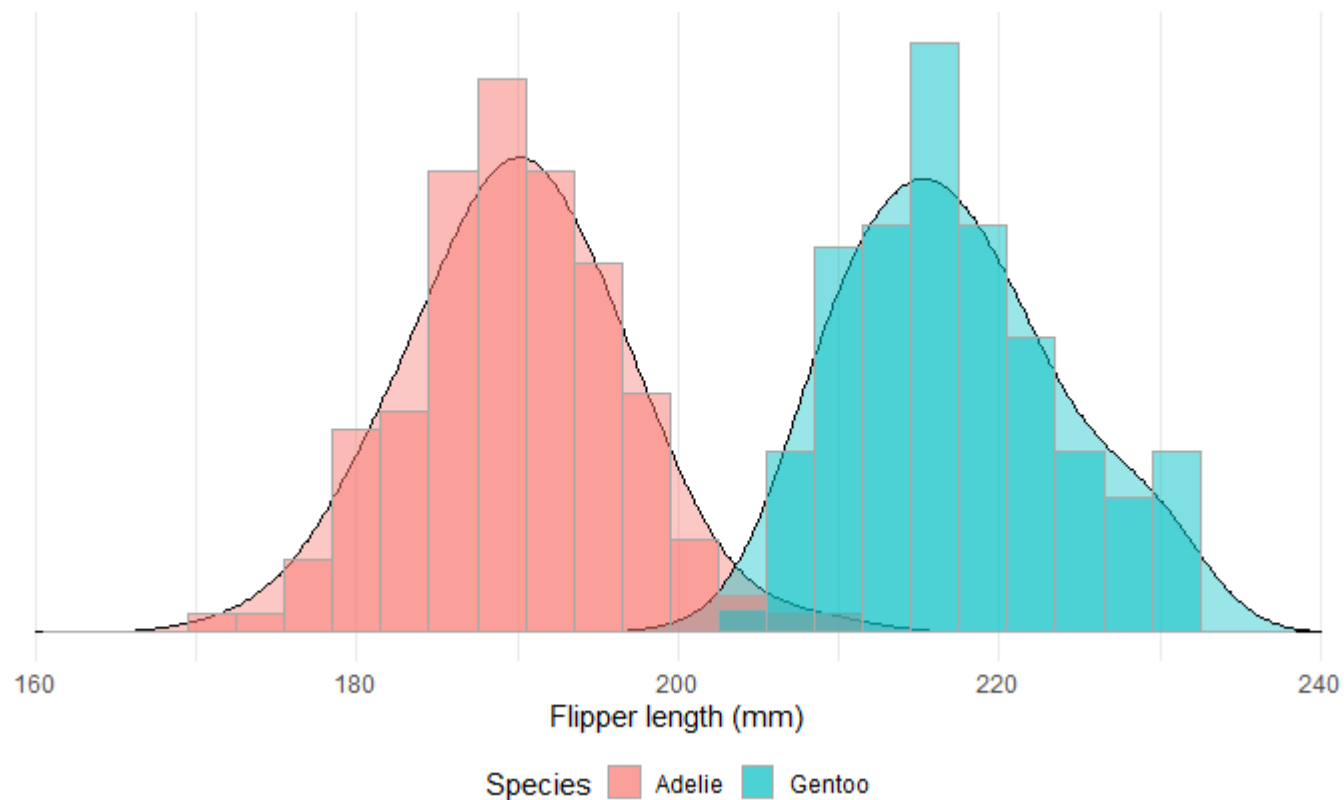
As well, we are slightly "hand-wavy" at times. In formal treatments of the EM algorithm you may see certain notational conventions (e.g., defining Q functions) that we will explain more intuitively with words and visualizations.

There is much to discuss regarding the EM algorithm both theoretically and in application. In STA 440, we will focus on using and implementing the EM algorithm in practice to tackle real-world datasets instead of focusing on theoretical considerations.

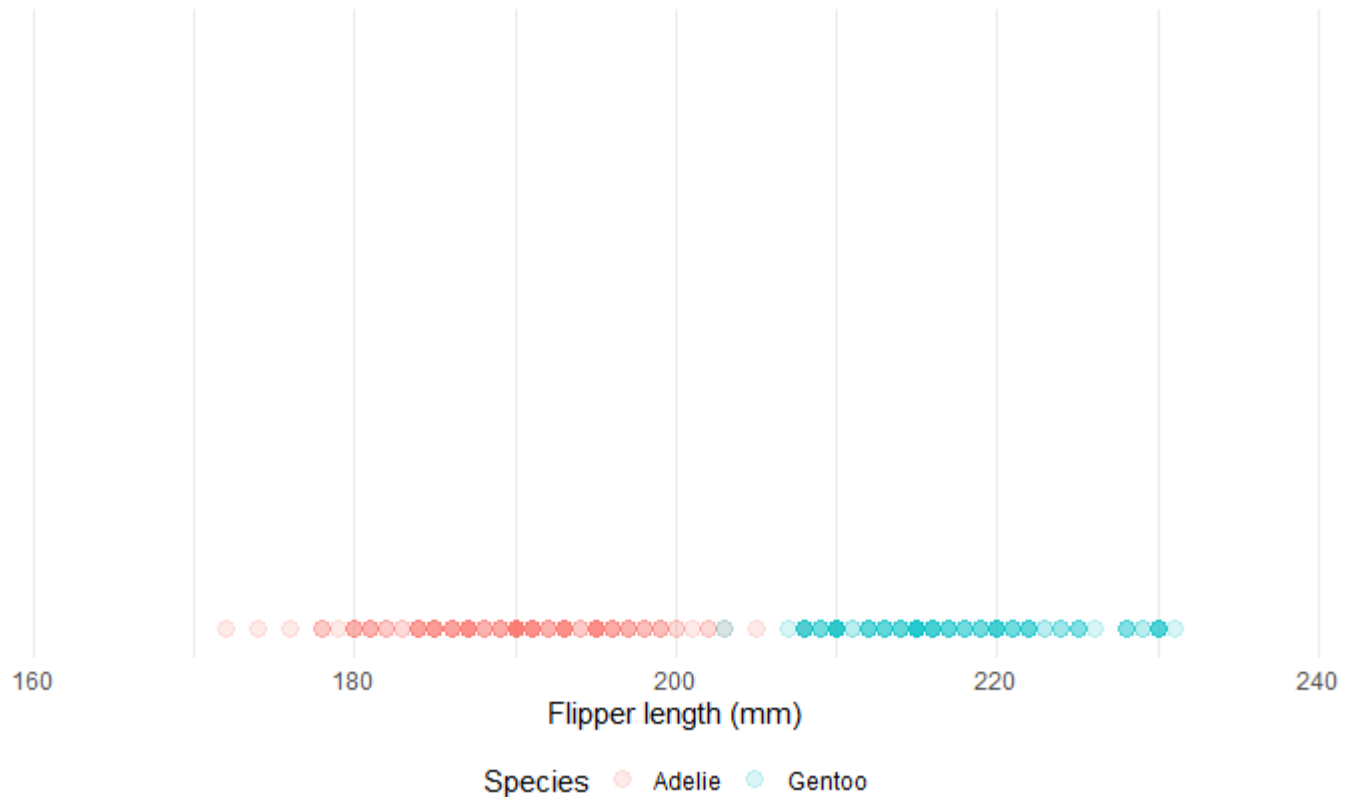
Penguin flipper length



Penguin flipper length

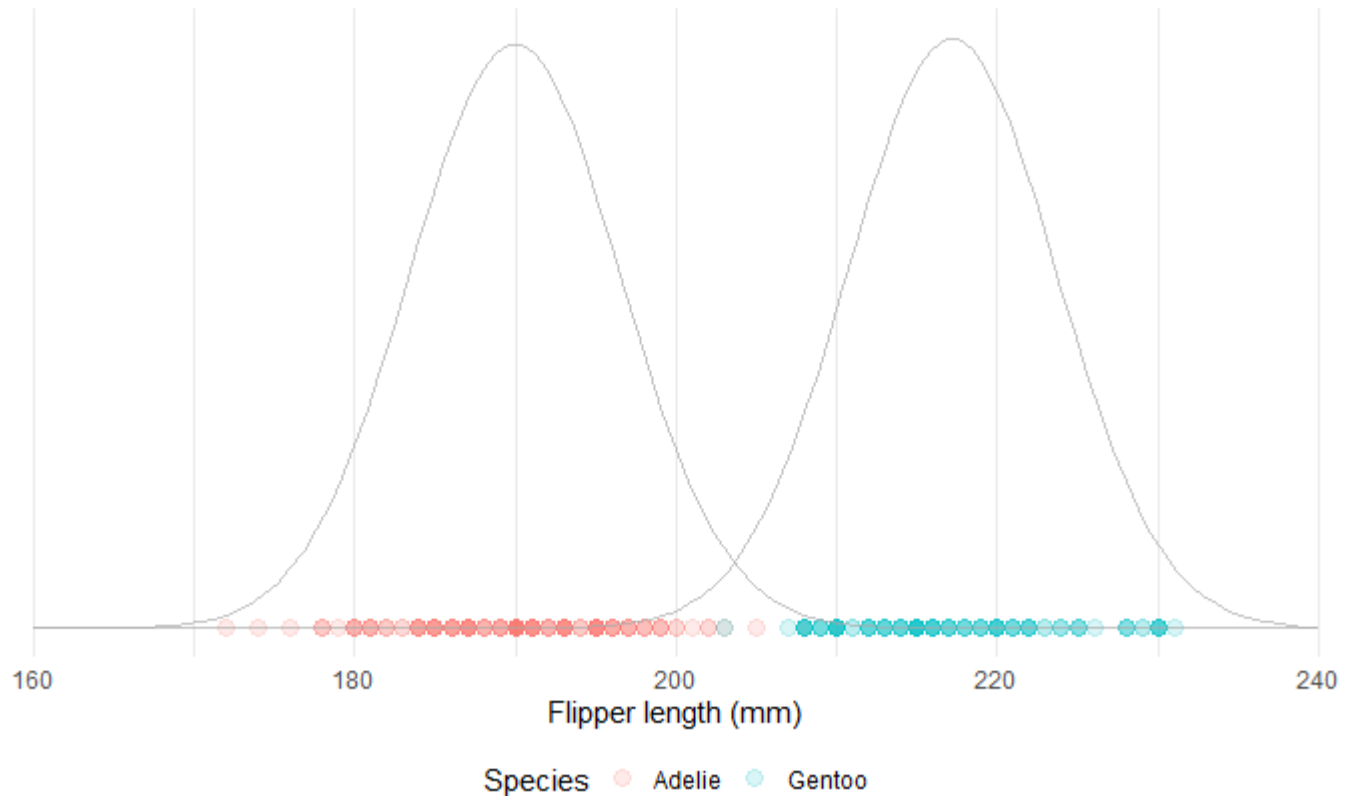


Penguin flipper length



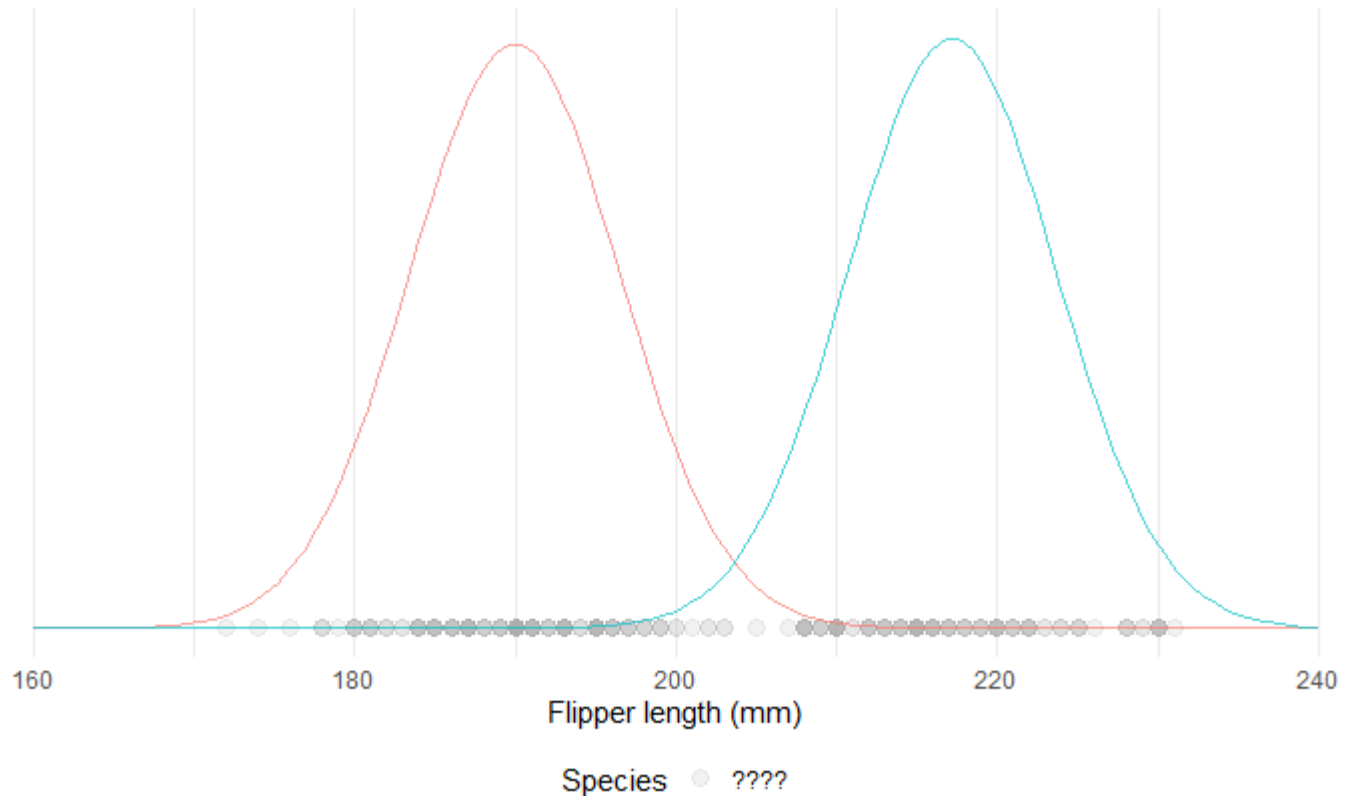
Let's say we have observations and know that they come from two separate normal distributions.

Penguin flipper length



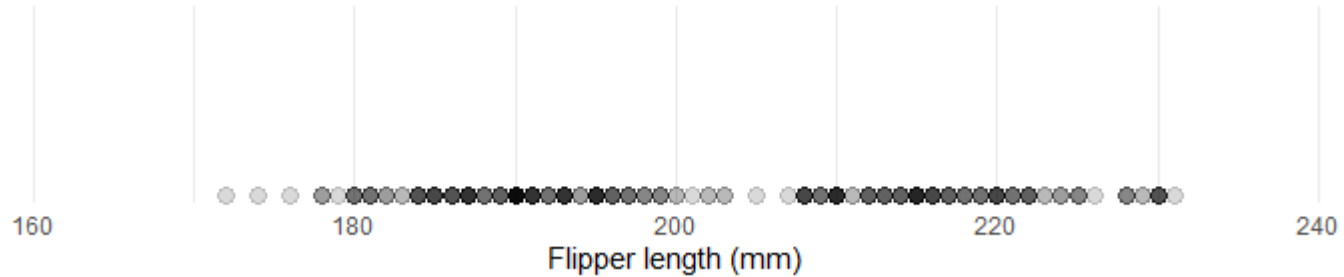
Since we know there are two sources and they're normally distributed, it's easy to estimate the mean and variance of the groups.

Penguin flipper length



On the other hand, if we know the parameters of the two distributions, we could estimate species-specific probabilities for each penguin.

Penguin flipper length



What if all we had was this? Could we guess the species labels and/or parameters of the two distributions?

Problem: If we knew which species each penguin was, we could estimate the means and variances. If we had the means and variances themselves, we could figure out which species each penguin was.

But we don't have either.

Penguin flipper length

We have the marginal mixture distribution:

$$f_X(x) = \pi_A \mathcal{N}(x|\mu_A, \sigma_A^2) + \pi_G \mathcal{N}(x|\mu_G, \sigma_G^2).$$

We'd like to maximize the log-likelihood:

$$\begin{aligned} & \log \mathcal{L}(\mu_A, \mu_G, \sigma_A^2, \sigma_G^2, \pi_A, \pi_G | \mathbf{X}) \\ &= \log \prod_{i=1}^n P(X_i | \mu_A, \mu_G, \sigma_A^2, \sigma_G^2, \pi_A, \pi_G) \\ &= \sum_{i=1}^n \log \{ \pi_A \mathcal{N}(x_i | \mu_A, \sigma_A^2) + \pi_G \mathcal{N}(x_i | \mu_G, \sigma_G^2) \} \end{aligned}$$

How can we maximize this log-likelihood?

Penguin flipper length

Unfortunately, there is no closed form solution (and numerical methods we've learned so far get very messy very fast).

Idea: let's introduce a **latent variable** such that when we write out the **complete data log-likelihood** that includes this latent variable, such a function is "easy" to maximize.

For our penguin example, define the random variable Z_i which takes on values A and G depending on what species penguin i is.

If we knew Z_1, \dots, Z_n (the true group assignments), then what would the complete data log-likelihood be?

Penguin flipper length

If we *knew* Z_1, \dots, Z_n , we would have

$$\begin{aligned} & \log \mathcal{L}(\mu_A, \mu_G, \sigma_A^2, \sigma_G^2, \pi_A, \pi_G | \mathbf{X}, Z) \\ &= \sum_{i=1}^n I(Z_i = A) \log \pi_A + I(Z_i = A) \log \mathcal{N}(x_i | \mu_A, \sigma_A^2) + \\ & \quad \sum_{i=1}^n I(Z_i = G) \log \pi_G + I(Z_i = G) \log \mathcal{N}(x_i | \mu_G, \sigma_G^2). \end{aligned}$$

We would like to maximize this complete data log-likelihood, but unfortunately we never actually observe Z .

How can we get information about Z in order to try and maximize the complete data log-likelihood?

Penguin flipper length

All the information we have about Z should be given by its posterior distribution conditional on the observed data and model parameters!

Thus, let's instead maximize the *posterior expectation* of the complete data log-likelihood with respect to Z , given the observed data and model parameters. That is:

$$\begin{aligned} & E_{Z|\mathbf{X}} [\log \mathcal{L}(\mu_A, \mu_G, \sigma_A^2, \sigma_G^2, \pi_A, \pi_G | \mathbf{X}, Z)] \\ &= E_{Z|\mathbf{X}} [P(\mathbf{X}, Z | \mu_A, \mu_G, \sigma_A^2, \sigma_G^2, \pi_A, \pi_G)] \end{aligned}$$

What is this expectation?

Penguin flipper length

$$\begin{aligned} & E_{Z|\mathbf{X}} [\log \mathcal{L}(\mu_A, \mu_G, \sigma_A^2, \sigma_G^2, \pi_A, \pi_G | \mathbf{X}, Z)] \\ &= \sum_{i=1}^n E_{Z|\mathbf{X}} [I(Z_i = A) \{\log \pi_A + \log \mathcal{N}(x_i | \mu_A, \sigma_A^2)\}] + \\ & \quad \sum_{i=1}^n E_{Z|\mathbf{X}} [I(Z_i = G) \{\log \pi_G + \log \mathcal{N}(x_i | \mu_G, \sigma_G^2)\}] \\ &= \sum_{i=1}^n P(Z_i = A | \mathbf{X}) \{\log \pi_A + \log \mathcal{N}(x_i | \mu_A, \sigma_A^2)\} + \\ & \quad \sum_{i=1}^n P(Z_i = G | \mathbf{X}) \{\log \pi_G + \log \mathcal{N}(x_i | \mu_G, \sigma_G^2)\} \end{aligned}$$

What are MLEs if we knew $P(Z_i = A | \mathbf{X})$ and $P(Z_i = G | \mathbf{X})$?

Penguin flipper length

Assuming we know $P(Z_i = A|\mathbf{X})$ and $P(Z_i = G|\mathbf{X})$,

$$\hat{\mu}_A = \frac{1}{\sum_{i=1}^n P(Z_i = A|\mathbf{X})} \sum_{i=1}^n P(Z_i = A|\mathbf{X}) x_i$$

$$\hat{\sigma}_A^2 = \frac{1}{\sum_{i=1}^n P(Z_i = A|\mathbf{X})} \sum_{i=1}^n P(Z_i = A|\mathbf{X}) (x_i - \mu_A)^2$$

$$\hat{\pi}_A = \frac{1}{n} \sum_{i=1}^n P(Z_i = A|\mathbf{X})$$

and similarly for $\hat{\mu}_G$, $\hat{\sigma}_G^2$, and $\hat{\pi}_G$ (work not shown; please verify!).

On the other hand, if we knew μ_A , μ_G , σ_A^2 , σ_G^2 , π_A , and π_G , how could we calculate $P(Z_i = A|\mathbf{X})$ and $P(Z_i = G|\mathbf{X})$?

Penguin flipper length

By Bayes' rule we have:

$$\begin{aligned} & P(Z_i = A | \mathbf{X}) \\ &= \frac{P(Z_i = A)P(\mathbf{X} | Z_i = A)}{P(Z_i = A)P(\mathbf{X} | Z_i = A) + P(Z_i = G)P(\mathbf{X} | Z_i = G)} \\ &= \frac{\pi_A \mathcal{N}(x_i | \mu_A, \sigma_A^2)}{\pi_A \mathcal{N}(x_i | \mu_A, \sigma_A^2) + \pi_G \mathcal{N}(x_i | \mu_G, \sigma_G^2)} \end{aligned}$$

and similarly for $P(Z_i = G | \mathbf{X})$.

The EM algorithm

We've now formalized the question given before, and found that we have the same chicken-and-egg problem:

- If we knew the parameters, we could easily estimate posterior probabilities $P(Z_i = z|\mathbf{X})$ (for $z \in \{\text{Adelie, Gentoo}\}$)
- If we knew the posterior probabilities $P(Z_i = z|\mathbf{X})$, we could easily estimate the parameters


The EM algorithm

The **EM algorithm** (expectation-maximization) is an iterative procedure that numerically solves this problem:

1. Initialize parameter values
2. **E-step**: construct expected log-likelihood function, where expectation takes advantage of latent variable formulation and is taken using parameter estimates from current M-step
3. **M-step**: calculate maximum likelihood estimates from expected log-likelihood function constructed from current E-step to inform distribution of latent variables
4. Repeat until convergence criterion is satisfied

We "guess" the latent variables and use them to maximize an "easier" likelihood. The EM algorithm hinges on defining a useful complete data log-likelihood.

The EM algorithm



Articles About 97,500 results (0.04 sec)

Any time

Since 2021

Since 2020

Since 2017

Custom range...

Maximum Likelihood from Incomplete Data Via the *EM* Algorithm

[AP Dempster, NM Laird...](#) - Journal of the Royal ..., 1977 - Wiley Online Library

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 ...

☆ 99 **Cited by 63152** [Related articles](#) [All 73 versions](#)

Maximum Likelihood from Incomplete Data via the *EM* Algorithm

By A. P. DEMPSTER, N. M. LAIRD and D. B. RUBIN

Harvard University and Educational Testing Service

[Read before the ROYAL STATISTICAL SOCIETY at a meeting organized by the RESEARCH SECTION on Wednesday, December 8th, 1976, Professor S. D. SILVEY in the Chair]

SUMMARY

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, including missing value situations, applications to grouped, censored or truncated data, finite mixture models, variance component estimation, hyperparameter estimation, iteratively reweighted least squares and factor analysis.

Keywords: MAXIMUM LIKELIHOOD; INCOMPLETE DATA; EM ALGORITHM; POSTERIOR MODE

fun fact: there's an error on page 8 in one of their proofs!

A few caveats

The M-step doesn't maximize the *observed* log-likelihood, but rather a surrogate function given by the conditional expectation of the complete data log-likelihood.

The good news is that at each iteration the value of the observed log-likelihood will never decrease. However, if initial values are chosen poorly, the EM algorithm may get stuck in a local maximum or stationary value.

In the previous example we found a closed form solution to the M-step (these are simply MLE estimates from normal distributions). However, sometimes no closed form exists and we must rely on numerical methods to obtain solutions (e.g., EM algorithm with one-step Newton-Raphson in the M step).

The EM algorithm

In the context of our estimation problem, we could have:

1. Initialize values $\mu_A, \mu_G, \sigma_A^2, \sigma_G^2$, and π_A (note that $\pi_G = 1 - \pi_A$)
2. Estimate the posterior probabilities $P(Z_i = A|\mathbf{X})$ and $P(Z_i = G|\mathbf{X})$ using current parameter estimates from M-step
3. Update MLEs using current parameter estimates and posterior probabilities from E-step
4. Repeat until convergence criterion is satisfied

The EM algorithm

Note that the E and M steps "cycle," and so we don't have to start with one or the other. In this case it might be tough to come up with initial parameter values. So, it may be easier to assign penguins to groups as the initialization step and start "in the middle," so to speak.

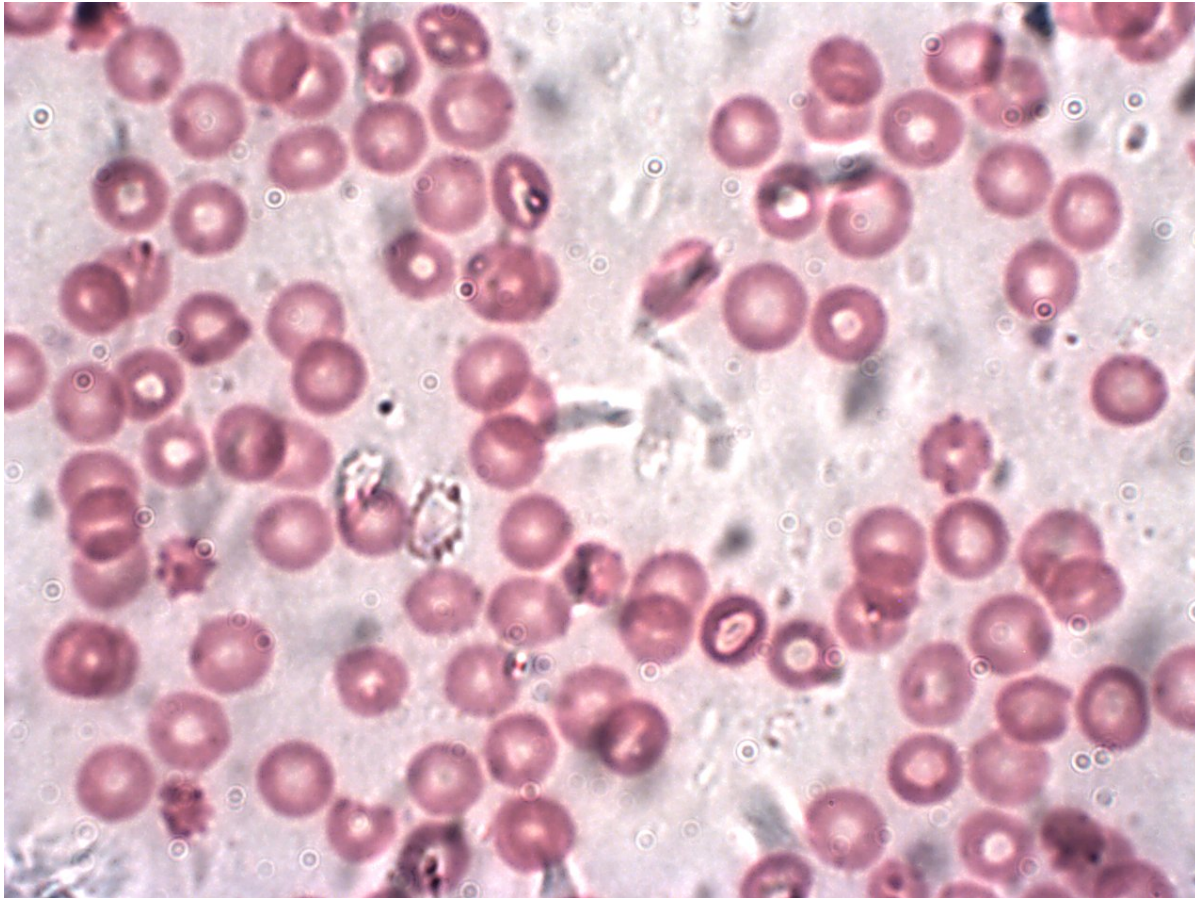
In practice, the convergence criterion is often based on changes in the log-likelihood function evaluated at parameter estimates at each iteration.

Try implementing the EM algorithm for the penguin data. What initial parameter values did you use? What were your final parameter estimates and group probabilities for each penguin?

The EM algorithm

Live demonstration: visualizing steps.

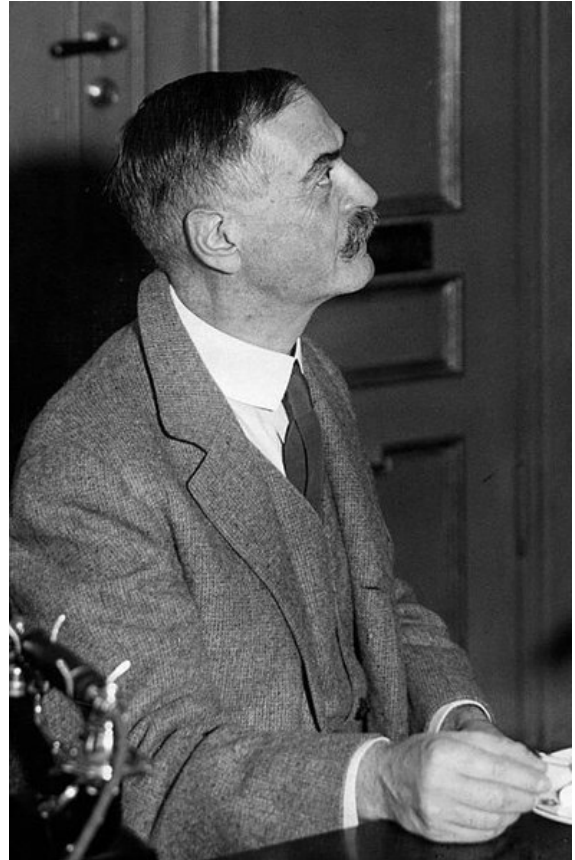
The ABO blood group system



The ABO blood group system

Human blood group refers to a classification of blood based on presence or absence of certain surface antigens on red blood cells.

One of the most important blood group systems is the ABO system; during blood transfusion, if mismatches occur in the ABO blood group system, potentially fatal reactions may occur.



The ABO blood group system

	A	B	O
A	AA	AB	AO
B	AB	BB	BO
O	AO	BO	OO

- Three alleles: A, B, and O
- One allele inherited from each parent
- A and B alleles are co-dominant; O is recessive
- Six unique genotypes given in table to the left
- Four phenotypes possible given these genotypes: A-type blood, B-type blood, AB-type blood, and O-type blood

The ABO blood group system

	A	B	O
A	AA	AB	AO
B	AB	BB	BO
O	AO	BO	OO

Let p_A , p_B , and p_O be **allele frequencies** of underlying alleles.
Then **genotype frequencies** are:

- AA: p_A^2
- BB: p_B^2
- AB: $2p_Ap_B$
- AO: $2p_Ap_O$
- BO: $2p_Bp_O$
- OO: p_O^2

Allele frequency estimation

Assume genotype counts are multinomially distributed:

$$P(N_{AA} = n_{AA}, \dots, N_{OO} = n_{OO}) = \frac{n!}{n_{AA} \dots n_{OO}!} p_{AA}^{n_{AA}} \dots p_{OO}^{n_{OO}},$$

for $\mathcal{B} = \{AA, BB, AB, AO, BO, OO\}$, and where $\sum_{i \in \mathcal{B}} n_i = n$.

We observe the phenotype counts N_A, N_B, N_{AB} and N_O only.

Using only *observed phenotype counts*, can we estimate **allele frequencies** in the underlying population? What is the observed data likelihood?

Allele frequency estimation

Using the observed *phenotype* frequencies, we have:

$$\begin{aligned}\mathcal{L}(p_A, p_B, p_O) &\propto (p_A^2 + 2p_A p_O)^{n_A} (p_B^2 + 2p_B p_O)^{n_B} \times \\ &\quad (2p_A p_B)^{n_{AB}} (p_O^2)^{n_O} \\ \log \mathcal{L}(p_A, p_B, p_O) &= n_A \log(p_A^2 + 2p_A p_O) + \\ &\quad n_A \log(p_A^2 + 2p_A p_O) + \\ &\quad n_{AB} \log(2p_A p_B) + 2n_O \log(p_O) + \\ &\quad \text{const.}\end{aligned}$$

Setting partial derivatives equal to zero and solving is a disaster.

How would you use a latent variable to write the complete data log-likelihood using the *genotype* frequencies?

The ABO blood group system

The complete data log-likelihood is

$$\begin{aligned}\log \mathcal{L}(p_A, p_B, p_O) = & n_{AA} \log(p_A^2) + n_{BB} \log(p_B^2) + \\ & n_{AO} \log(2p_A p_O) + n_{BO} \log(2p_B p_O) + \\ & n_{AB} \log(2p_A p_B) + n_{OO} \log(p_O^2) + \text{const.},\end{aligned}$$

which we notice is much easier to maximize (no more sums of different parameters inside a log).

How could you maximize this log-likelihood function with respect to the parameters of interest (watch out for a constraint)?

The ABO blood group system

Let's calculate MLEs for p_A , p_B , and p_O using the complete data log-likelihood. We can use Lagrange multipliers to solve the optimization problem under the constraint $p_A + p_B + p_O = 1$ (again, no work shown; please verify!):

$$\begin{aligned}\hat{p}_A &= \frac{2n_{AA} + n_{AO} + n_{AB}}{2n} \\ \hat{p}_B &= \frac{2n_{BB} + n_{BO} + n_{AB}}{2n} \\ \hat{p}_O &= \frac{2n_{OO} + n_{AO} + n_{BO}}{2n}.\end{aligned}$$

Now how about the E-step?

The ABO blood group system

Once again, we never observe the latent variables (in this case the genotype frequencies). However we can again get useful information by maximizing conditional expectations of the log-likelihood for the latent data, given the observed data and model parameters.

For observed blood types AB and O which can only happen given a single genotype combination we have

$$E(n_{AB} | \mathbf{X}, p_A, p_B, p_O) = n_{AB}$$

$$E(n_{OO} | \mathbf{X}, p_A, p_B, p_O) = n_O$$

How can we get information about genotype frequencies n_{AA} and n_{AO} given our observed phenotype frequencies?

The ABO blood group system

Note that $n_{AA} + n_{AO} = n_A$. Thus, we know

$$n_{AA}|n_A \sim \text{Binomial} \left(n_A, \frac{p_A^2}{p_A^2 + 2p_Ap_O} \right)$$

and so we have

$$E(n_{AA}|\mathbf{X}, p_A, p_B, p_O) = \frac{n_A p_A^2}{p_A^2 + 2p_Ap_O}$$

The ABO blood group system

Similarly,

$$E(n_{BB}|\mathbf{X}, p_A, p_B, p_O) = \frac{n_B p_B^2}{p_B^2 + 2p_B p_O}$$

$$E(n_{AO}|\mathbf{X}, p_A, p_B, p_O) = \frac{2n_A p_A p_O}{p_A^2 + 2p_A p_O}$$

$$E(n_{BO}|\mathbf{X}, p_A, p_B, p_O) = \frac{2n_B p_A p_O}{p_B^2 + 2p_B p_O}$$

We can thus plug in these conditional expectations to the expressions for the MLE from the M-step, and use MLE estimates of p_A , p_B , and p_O from the M-step to obtain estimates for these conditional expectations in the E-step, and so on.

The ABO blood group system

The EM algorithm for estimating allele frequencies is thus given by

1. Initialize values p_A , p_B , and p_O (note that $p_A + p_B + p_O = 1$)
2. Estimate genotype counts using current parameter estimates from M-step
3. Update MLEs using parameter estimates and genotype counts from E-step
4. Repeat until convergence criterion is satisfied

Suppose in a group of 300 individuals we observe 135 with blood type A, 39 with blood type B, 108 with blood type O, and 18 with blood type AB. Try implementing the EM algorithm. What initial parameter values did you use? What were your final allele frequencies?