MT 4113: Computing in Statistics, Assignment 2

Setter: Eiren Jacobson

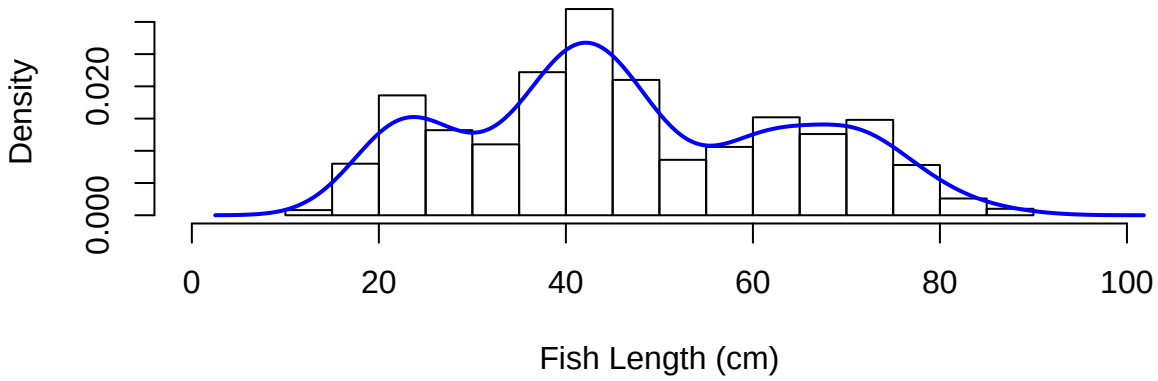
Updated: 13 October 2018. Due: 19 October 2018

Motivation

In the field of fisheries stock assessment, biologists model fish populations to estimate sustainable levels of fishing mortality. One important component of these models is the age structure of the fish population, which can be estimated by taking a sample of fish from the population and counting growth layers on a small ear bone called the otolith. This is similar to ageing a tree by counting growth rings.

This is a time consuming process and thus relatively expensive. It is easier to measure fish lengths; however, due to individual variation in growth rates, fish length does not correspond perfectly to fish age. We can consider observed fish lengths as resulting from a mixture of length distributions that correspond to age cohorts.

In this assignment, you will be given 1000 fish length measurements, 100 of which are of known age. Your task is to determine the mean and standard deviation of the expected length-at-age for each age cohort in the sample and estimate the proportion of the sample that belongs to each age cohort.



Problem

Given a set of observations of fish lengths spanning three age cohorts, estimate the parameters of the Gaussian (normal) distributions of length-at-age, when the age of each length observation is not known.

The probability density function of this mixture distribution is

$$f(x) = \lambda_1\phi(x|\mu_1, \sigma_1^2) + \lambda_2\phi(x|\mu_2, \sigma_2^2) + \lambda_3\phi(x|\mu_3, \sigma_3^2),$$

with λ_k being the probability of an observation coming from component k and ϕ is a normal distribution with mean μ_k and variance σ_k^2.

If we knew the component that each observation came from, we would use maximum likelihood to estimate mean length $\hat{\mu}_k$ and standard deviation $\hat{\sigma}_k$ of each of the k ages. We would also know the proportion of

observations coming from each age cohort λ_k. However, we do not know which component of the distribution generated each observation (the generation process is "hidden" or "latent"). The key to the estimation problem is therefore to **assign** labels (ages) to observations by some process, and subsequently perform maximum likelihood estimation of $\hat{\mu}_k$ and $\hat{\sigma}_k$. This assignment and estimation is performed repeatedly until the parameter estimates cease changing.

Algorithm to perform the estimation

The expectation maximization (EM) algorithm is the tool we will use for this task. It consists of initialization followed by iteration over expectation and maximization, with test for convergence determining when iteration can stop.

Background reading

Do, C.B. and S. Batzoglou. 2008. What is the expectation maximization algorithm? Nature Biotechnology 26:897-899.

This short paper is also available in the Assignment 2 folder on Moodle.

Initialization

1. Using data from the fish of known age as a starting point, assign labels (ages) to each of N length observations in the dataset that come from fish of unknown age.
2. Given these initial assignments, compute initial values of $\hat{\mu_k}$ and $\hat{\sigma}_k$ for each age class $k = 1, 2, 3$.
3. Compute initial values of $\hat{\lambda}_k$ as n_k/N.

Expectation

Calculate the probability that length observation x_i belongs to component k using Bayes Rule:

$$P(x_i \in k|x_i) = \frac{P(x_i|x_i \in k)P(k)}{P(x_i)}.$$

This *posterior probability* is computed for each observation x_i based upon our initial guesses about the model parameters. $P(\cdot)$ represents the probability of the event described inside the parenthesis and $x_i \in k$ is the event that observation x_i comes from component k.

How can we calculate the probability that observation x_i belongs to age class k? If we assume each component of the mixture is Gaussian, then these probabilities are derived from the normal distribution with parameters $\hat{\mu}_k$ and $\hat{\sigma}_k$. We compute these probabilities for each observation for each possible component that may have generated each x_i.

To ensure $P(x_i \in k|x_i)$ is a probability, i.e., $0 \leq P(x_i \in k|x_i) \leq 1$, standardize $P(x_i \in k|x_i)$ by dividing by the sum of probabilities across all three components:

$$P(x_i) = \sum_{k=1}^{3} P(x_i|x_i \in k)P(k)$$

Hint: at this point you should have a matrix with rows for each length observation and columns for each age class, where the matrix components are the probability of membership in each age class.

Maximization

Given the probabilities of membership estimated above, compute another iteration of the estimate $\hat{\mu_k}$ as

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} P(x_i \in k|x_i)x_i}{\sum_{i=1}^{N} P(x_i \in k|x_i)}.$$

Note that this is a weighted mean, with the weights being the probability of an observation belonging to each age class for each observation.

The new estimate of $\hat{\sigma}_k$ is based on $\hat{\mu}_k$:

$$\hat{\sigma}_k = \sqrt{\frac{\sum_{i=1}^{N} P(x_i \in k|x_i)\,(x_i - \hat{\mu}_k)^2}{\sum_{i=1}^{N} P(x_i \in k|x_i)}}.$$

Finally,

$$\hat{\lambda}_k = \frac{1}{N}\sum_{i=1}^{N} P(x_i \in k|x_i).$$

Likelihood evaluation

We can now calculate the likelihood of our data given these new values of $\hat{\lambda}$, $\hat{\mu}$, and $\hat{\sigma}$. The likelihood is

$$P(X|\mu, \sigma, \lambda) = \sum_{k=1}^{K} \lambda_k \mathcal{N}(X|\mu_k, \sigma_k^2)$$

note $\mathcal{N}()$ refers to the normal distribution and σ_k^2 refers to the variance (rather than standard deviation) of the k^{th} component of the mixture.

This likelihood is calculated at each iteration.

Iteration and the stopping rule

The process of expectation and maximisation is repeated until the likelihood of data given the estimated parameters in the model changes negligibly. The magnitude of the change in successive likelihoods forms our criterion to determine when iteration should cease. Because likelihoods can become vanishingly small (and lead to underflow problems), it is easier to work with log-likelihoods:

$$\ln(\mathcal{L}) = \ln(P(X|\mu, \sigma, \lambda)) = \sum_{n=1}^{N} \ln\left[\sum_{k=1}^{K} \lambda_k \mathcal{N}(x_n|\mu_k, \sigma_k^2)\right]$$

Convergence is reached when

$$\ln(\mathcal{L}^{i+1}) - \ln(\mathcal{L}^{i}) < \epsilon$$

where $\ln(\mathcal{L}^{)+\infty})$ is the log-likelihood from the i^{th} iteration of the EM algorithm and ϵ is a desired tolerance.

Your assignment

You will be working in teams of four and collaborating via the version control software Git and the web-based hosting service GitHub. We will set up skeleton repositories for you and show you how to use them in Practical 3. It will be your responsibility to design your code (how many functions, how is information passed between functions) and delegate code writing, documentation, and testing duties among team members. We suggest implementing the algorithm can be conveniently broken into three or four functions. While we have asked you to apply the EM algorithm to a specific dataset, your code should be generalizeable to other datasets. The README file of your GitHub team repository is a markdown document and will serve as your project report. We will demonstrate basic markdown syntax in Practical 3. This file should include the text, figures, and tables described in the tasks below.

Task 1: Explore the data

- Create two plots of the data: one with age on the x-axis and length on the y-axis, and one with length on the x-axis and frequency on the y-axis. Add axis labels and any additional information you think is relevant (e.g., average length-at-age).
- Comment on the data and plots. Describe the properties of the data (max 250 words).

Task 2: Describe the methods

- In your own words, describe how you will apply the EM algorithm to this dataset (max 250 words).
- Include a diagram (e.g., a flowchart) of your team's approach to the problem. This can be digital or hand-drawn and scanned in as a PDF.

Task 3: Implement the EM algorithm

Create modular code consisting of a set of documented and tested functions implementing the EM algorithm. Your main function, called by the user, will be called `teamEM` and it will take three arguments:

- a dataframe (argument name `data`, no default) of observations with columns for FishID, Length, and Age (when known)
- a tolerance value ϵ (argument name `epsilon`, default 1e-08),
- a maximum number of iterations (argument name `maxit`, default value 1000)

The function `teamEM` will return a named list containing the following:

- **estimates**: a dataframe of estimates with rows `Age1`, `Age2`, and `Age3`, and columns for `mu` ($\hat{\mu}_k$), `sigma` ($\hat{\sigma}_k$, standard deviation, not variance), and `lambda` ($\hat{\lambda}_k$). You can use the function `rownames` to name the rows and `colnames` to name the columns.
- **inits**: a dataframe of initial values with rows `Age1`, `Age2`, and `Age 3` and columns for `mu`, `sigma`, and `lambda`. Again, use `rownames` and `colnames` to name the rows and columns.
- **converged**: TRUE or FALSE (Boolean) indicating whether algorithm converged
- **posterior**: N by k dataframe of posterior probabilities: dimensions–'number of observations' rows by 'number of components' columns
- **likelihood**: vector of log-likelihood values, one for each iteration of the algorithm; vector length is number of iterations of the algorithm

Task 4: Test your function

- Write a function to create simulated datasets with similar properties to the "true" data
- Show that your implementation of the EM algorithm returns correct values for simulated datasets

Task 5: Report results

- Include a table of the estimates returned by `teamEM`
- Plot the original data with the densities of the mixture components superimposed

Task 6: Describe the work of each team member

- Statement of original work
- In one sentence per team member, describe who did what (e.g., Matilda wrote function x, tested function y, documented function z).

What to hand in

- submit to MMS a *release* from the Github team repository (all members of the team will submit the same .zip file to MMS)
- in the release file and team repository will be all the functions needed to carry out the assignment
- The README of the team repository will serve as your project report

Marking scheme

This assignment will be marked with computer assistance, just like Assignment 1. The team's code will be sourced, the output from the `teamEM` function will be checked against results of Gaussian mixture model fitting produced by the function `normalmixEM` in the package `mixtools`.

For example, testing code would evaluate operation of your submission like this:

```
mod <- teamEM(data, epsilon = 1e-08, maxit = 1000)
mixmod <- normalmixEM(data$Length, mu = mod$inits$mu,
                      sigsqrd = mod$inits$sigma, lambda = mod$inits$lambda,
                      epsilon = 1e-08, maxit = 1000)
mark <- 0
mark <- mark + (max(abs(mod$estimates$mu - mixmod$mu)) < 0.2)
mark <- mark + (nrow(mod$posterior) == length(data$Length))
mark <- mark + ((abs(mod$likelihood[length(mod$likelihood)] -
          mixmod$loglik)) < 0.01)
```

Fifty marks in total are available:

- ten marks will be awarded based on the automated tests that we will perform on your code,
- ten marks for evidence of code design and testing,
- ten marks for programming style,
- ten marks for documentation,
- ten marks for any enhancements your team might wish to add to increase functionality of the code.

All members of the team will receive the same mark.

Because this assignment counts towards your final grade, it is important that you do not collaborate with individuals outside your team in completing the work. You should be comfortable with the following statement, which you should include in the work attribution section of your README file.

```
# I confirm that this repository is the work of our team, except where clearly indicated
in the text.
```

Marks will be deducted if this statement is not present.

For more information, see the "Academic misconduct' section of the university web site [http://www.st-andrews.ac.uk/students/rules/academicpractice/]. This holds for all assessed project work on this course. Plagiarism cannot be tolerated on this or any other assignment for this module.