Programming Assignment 5 – Basic Probability, Computing and Statistics 2016

Fall 2016, Master of Logic, University of Amsterdam

Instructors: Christian Schaffner and Bas Cornelissen

Submission deadline: Friday, December 9th, 8 p.m.

Note: if the assignment is unclear to you or if you get stuck, do not hesitate to ask in the forum.

1 Assignment

This week we are finally going to implement the expectation maximisation algorithm (EM). You can look up explanations of the algorithm here and see an example on these slides. Our data (download here) have been generated from a mixture model. The mixture components are geometric distributions. Your tasks is to find the mixture weights and parameters of the mixture components. The data were generated from 3 mixture components, so this is the number of components that you should use as well.

Recall that EM is sensitive to the starting point (i.e. the initial parameter estimate from which we start optimising). During peer review, we will provide you with a starting point whose resulting parameter estimates we have already computed. You will be assessed according to how well your algorithm does on this particular starting point. This implies that you will have to provide a place in your code (ideally at the beginning) where the reviewers can plug in that fixed starting point.

2 Implementation

We have provided a partial implementation for you here. You have to implement the E-step and the M-step. The positions where you have to fill in your code are marked with TODO. Before you start implementing anything, make yourself familiar with the code and ensure that you understand all the data structures it uses. You can also discuss the basic code on the forum. As usual, feel free to add any variables or functions/methods that you deem necessary. If you want to remove any of the data structures or methods that we provide, that's also fine.

Notice that we are again in a setting were we have to work with logprobs. In order to compute the log-likelihood (see next section), we have to add those logprobs. This is where the logarithm functions that we implemented in week 3 come in handy. Please use them whenever needed. We provide a correct implementation of them here.

3 Debugging

Recall that EM is guaranteed to always increase the log-likelihood of the data. Therefore, the code prints the log-likeihood after each iteration. If it goes up after each iteration or stays unchanged, your implementations is probably correct (no guarantees, though). Please make sure to be very strict about this. Even if an EM implementation is wrong, the log-likelihood often still increases during the initial iterations. Pay close attention to later iterations. If they are volatile, you have a bug! EM usually does not need all too many iterations before it converges. Always run 20 iterations, which should be enough for this data set.

Hint about the log-likelihood: In order to compute the log-likelihood for the data set, you need to add the log-likelihoods of all data points (this is ok because we assume that the data points are independent given the mixture components). In order to compute the log-likelihood of the a data point x, you need to sum over all possible mixture assignments.

$$\log(P(X = x)) = \log\left(\sum_{c} P(X = x|Y = c)P(Y = c)\right) \tag{1}$$

Observe that you already compute this sum as part of computing the posterior $P(Y = c|X = x)$. You can hence reuse the result you get there in order to compute the log-likelihood. This implies that the log-likelihood that you print out at after iteration t is the log-likelihood obtained under the parameters of iteration $t - 1$ but that is ok.

4 Grading

You will be graded on your performance from a specific starting point, i.e. a specific setting of initial mixture weights and parameters of mixture components. Make sure that these can be provided to your code. This can be done either by defining list variables at the beginning of your python file (one list of the mixture weights, one list for the component parameters) or, more elegantly, by asking the user to provide a text file in a specific format from which your program reads the initial parameter settings.

1 point All additional classes and functions/methods have docstrings. Award 0 points here if there are one or more classes/functions/methods that do not have a docstring. If the student didn't add any classes or functions, award 1 point by default.

2 points If `initial_mixture_weights` and `initial_geometric_parameters` are provided, they are correctly taken into account in the first step. If these parameters are not provided, random initialisation values are used.

1 point The e-step is correctly implemented. Inspect the code to check this.

1 point The m-step is correctly implemented. Inspect the code to check this.

2 points The log-likelihood increases monotonically when the algorithm is run. It's ok if after a couple of iterations the log-likelihood does not change anymore, as long as it does not go down. (Be strict here!)

2 points Your parameter estimate after 20 iterations for the fixed initial parameter settings is equal to the parameter estimate that we will provide during the peer review period.