Programming Assignment 5 – Basic Probability, Computing and Statistics 2016

Fall 2016, Master of Logic, University of Amsterdam

Instructors: Philip Schulz, Christian Schaffner and Bas Cornellisen

Submission deadline: Wednesday, December 7th, 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 here. 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 5 mixture components, so this is the number of components that you should use as well.

Recall that EM is sensitive to the starting point. 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 do not provide skeleton code this time. You will have to write all functions (and classes if you want to work with those) from scratch. You are encouraged to discuss your implementation in the forum. While you should not post any code, asking questions like *Do you think that this is the right data structure to use?* are is perfectly 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, you should compute and print 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 more than enough for this data set.

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 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.

1 point A starting point for EM can be provided to program.

1 point If no starting point is provided, the parameters are initialised randomly.

1 point The program prints the log-likelihood after each iteration. After iteration t it is fine to print the likelihood achieved with the parameter settings from iteration $t-1$. This simplifies the computation (figure out yourselves how).

3 points The log-likelihood increases monotonically when the algorithm is run.

3 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.