15/10

6.825
Techniques in Artificial Intelligence
2005/11/17

Dan Ports
drkp@mit.edu

Project 3 Exercise 1

We begin by considering the first four data points. After implementing the E-M algorithm[1], we run it for four iterations on the data points. The results are shown below.

t	p_{11}	p_{12}	p_{13}	p_{14}	p_1	μ_1	σ_1^2	p_2	μ_2	σ_2^2	$\log p(\mathfrak{X} \mid \Theta_t)$
0					0.5000	-1.0000	1.0000	0.5000	1.0000	1.0000	-9.0750
1	0.0030	0.3067	0.0853	0.0069	0.1005	0.6274	0.2013	0.8995	1.8683	0.9268	-5.4771
2	0.0000	0.4020	0.1242	0.0001	0.1316	0.5917	0.1097	0.8684	1.9181	0.8954	-5.3509
3	0.0000	0.5701	0.1043	0.0000	0.1686	0.5282	0.0793	0.8314	1.9901	0.8180	-5.2072
4	0.0000	0.7331	0.0592	0.0000	0.1981	0.4659	0.0419	0.8019	2.0592	0.7251	-4.9151

Extra credit Using all the data, we test how long the E-M procedure requires to converge. We define the procedure to have converged when p_i, mu_i, and σ_i^2 do not change by more than δ after an E-M step. For $\delta = 0.01$, 12 iterations are required; for $\delta = 0.001$, 66 iterations are required; for $\delta = 0.0001$, 543 iterations are required; and for $\delta = 0.00001$, 813 iterations are required. The results reached are shown below:

t	p_1	μ_1	σ_1^2	p_2	μ_2	σ_2^2	$\log p(\mathfrak{X} \mid \Theta_t)$
12	0.2287	-0.1549	0.3657	0.7713	2.5329	0.6593	-48.2469
66	0.2552	-0.0287	0.4846	0.7448	2.5853	0.5963	-48.2223
543	0.3000	0.1714	0.6681	0.7000	2.6668	0.5120	-48.2175
813	0.3026	0.1820	0.6778	0.6974	2.6713	0.5076	-48.2174

[1]Python source code for the implementation is available at `http://www.ambulatoryclam.net/svn/classes/6.825/proj3exercises/ex1`