Estimating Gaussian Mixture Models using the Expectation Maximization Algorithm

by Mahmoud Abdelkhalek

Abstract: This report outlines the author's attempt at estimating the parameters of a Gaussian Mixture Model (GMM) using the Expectation Maximization (EM) algorithm. A GMM was initialized and then sampled from to obtain the data used in this experiment. Next, using this data, estimates of the parameters of the GMM were obtained using the EM algorithm. The effect of different initial estimates of these parameters on the EM algorithm was evaluated. Also, the original parameters were varied to test the clustering capability of the EM algorithm. Experiments showed that while the EM algorithm generally converges quickly, it is not always accurate and can converge to inaccurate parameter estimates given initial parameters estimates that are very different to the original parameters. Nonetheless, in different contexts, the EM algorithm may be accurate enough for estimating GMM parameters.

1. Introduction

A mixture model is a linear combination of probability density (or mass) functions in the form of [1]:

$$p(\boldsymbol{x}) = \sum_{j=1}^{J} p(\boldsymbol{x}|j) \cdot P_j \tag{1}$$

Where P_j are the weights and:

$$\sum_{j=1}^{J} P_j = 1, \qquad \int_{\boldsymbol{x}} p(\boldsymbol{x}|j)\, d\boldsymbol{x} = 1$$

This means that sampling a random vector $\boldsymbol{x}$ from this mixture model involves sampling from any of the J distributions with probability P_j. In the case of GMM's, the conditional distribution $p(\boldsymbol{x}|j)$ is a Gaussian distribution with parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$:

$$p(\boldsymbol{x}) = \sum_{j=1}^{J} p(\boldsymbol{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j | j) \cdot P_j \tag{2}$$

Where for a d-dimensional random vector $\boldsymbol{x}$:

$$p(\boldsymbol{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j | j) = \frac{1}{(2\pi)^{\frac{d}{2}} |\boldsymbol{\Sigma}_j|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j)\right) \tag{3}$$

Now suppose that you are given a GMM and you only know its number of components J while its parameters $\boldsymbol{\mu}_j$, $\boldsymbol{\Sigma}_j$, and P_j are unknown and need to be estimated. This can be done using maximum likelihood estimation. Suppose that N independent and identically distributed vectors are sampled from this GMM: $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$. Then, the log-likelihood function associated with this data is:

$$\ln p(X;\boldsymbol{\mu},\boldsymbol{\Sigma}) = \ln\left(\prod_{i=1}^{N} p(\boldsymbol{x}_i)\right)$$

$$= \ln\left(\prod_{i=1}^{N}\left[\sum_{j=1}^{J} p\left(\boldsymbol{x}_i;\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j\middle|j\right)\cdot P_j\right]\right) = \sum_{i=1}^{N}\ln\left(\sum_{j=1}^{J} p\left(\boldsymbol{x}_i;\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j\middle|j\right)\cdot P_j\right) \tag{4}$$

Where each of the $p\left(\boldsymbol{x}_i;\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j\middle|j\right)$ terms are given in equation 3. Differentiating equation 4 with respect to the k^{th} mean vector $\boldsymbol{\mu}_k$ and equating to 0 yields [2]:

$$\frac{\partial(\ln p(X;\boldsymbol{\mu},\boldsymbol{\Sigma}))}{\partial\boldsymbol{\mu}_k} = \sum_{i=1}^{N}\left[\frac{P_k\cdot p(\boldsymbol{x}_i;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k|k)}{\sum_{j=1}^{J} P_j\cdot p\left(\boldsymbol{x}_i;\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j\middle|j\right)}\cdot\boldsymbol{\Sigma}_k(\boldsymbol{x}_i-\boldsymbol{\mu}_k)\right] = \boldsymbol{0} \tag{5}$$

Re-arranging equation 5 to solve for $\boldsymbol{\mu}_k$:

$$\widehat{\boldsymbol{\mu}}_k = \frac{1}{N_k}\sum_{i=1}^{N}\gamma_{ik}\cdot\boldsymbol{x}_i \tag{6}$$

Where:

$$\gamma_{ik} = \frac{P_k\cdot p(\boldsymbol{x}_i;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k|k)}{\sum_{j=1}^{J} P_j\cdot p\left(\boldsymbol{x}_i;\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j\middle|j\right)} = \frac{P_k\cdot p(\boldsymbol{x}_i;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k|k)}{p(\boldsymbol{x}_i)} = \frac{p(\boldsymbol{x}_i,k)}{p(\boldsymbol{x}_i)} \tag{7}$$

$$N_k = \sum_{i=1}^{N}\gamma_{ik} \tag{8}$$

Notice that the term γ_{ik} is essentially the probability of a vector in the k^{th} mixture component weighted by the probability of the same vector in the GMM and N_k is the number of vectors sampled from the k^{th} component. Using the same process, the estimates for the k^{th} covariance matrix $\boldsymbol{\Sigma}_k$ and the k^{th} component probability P_k can also be computed [2]:

$$\widehat{\boldsymbol{\Sigma}}_k = \frac{1}{N_k}\sum_{i=1}^{N}\gamma_{ik}\cdot(\boldsymbol{x}_i-\boldsymbol{\mu}_k)(\boldsymbol{x}_i-\boldsymbol{\mu}_k)^T \tag{9}$$

$$\widehat{P}_k = \frac{N_k}{N} \tag{10}$$

Notice, however, that $\widehat{\boldsymbol{\mu}}_k$ and $\widehat{\boldsymbol{\Sigma}}_k$ in equations 6 and 9 are directly dependent on γ_{ik}, while γ_{ik}, from equation 7, is dependent on $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ for $j = 1,2,\dots,J$. This vicious dependency cycle can be resolved by first setting initial guesses for $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ for $j = 1,2,\dots,J$, then computing γ_{ik}, then computing the new estimates $\widehat{\boldsymbol{\mu}}_k$, $\widehat{\boldsymbol{\Sigma}}_k$, and $\widehat{P}_k$ for $k = 1,2,\dots,J$, and then substituting these new estimates back into equation 7 to compute a new value for γ_{ik}, and finally repeating this until the values of $\widehat{\boldsymbol{\mu}}_k$, $\widehat{\boldsymbol{\Sigma}}_k$, and $\widehat{P}_k$ for $k = 1,2,\dots,J$ converge. The E step in the EM algorithm corresponds to computing γ_{ik} for all values of i and k and the M step in the EM algorithm corresponds to computing the estimates $\widehat{\boldsymbol{\mu}}_k$, $\widehat{\boldsymbol{\Sigma}}_k$, and $\widehat{P}_k$ for $k = 1,2,\dots,J$.

Therefore, the goal of this project is to first sample vectors from the GMM given in equation 2 using known parameters, then to use equations 6,7,8,9, and 10 to estimate these known parameters using only the sampled vectors. Reference [2] helped me to understand GMM's better and also contained derivations for equations 6,7,8,9, and 10. Reference [3] helped me to understand the derivations in [2]. Section 2 of this report illustrates the methodology used to estimate the GMM parameters under different conditions. Section 3 discusses the results of these experiments using contour plots. Section 4 concludes the report and provides some advice to the reader, while section 5 contains references.

2. Experimental set-up

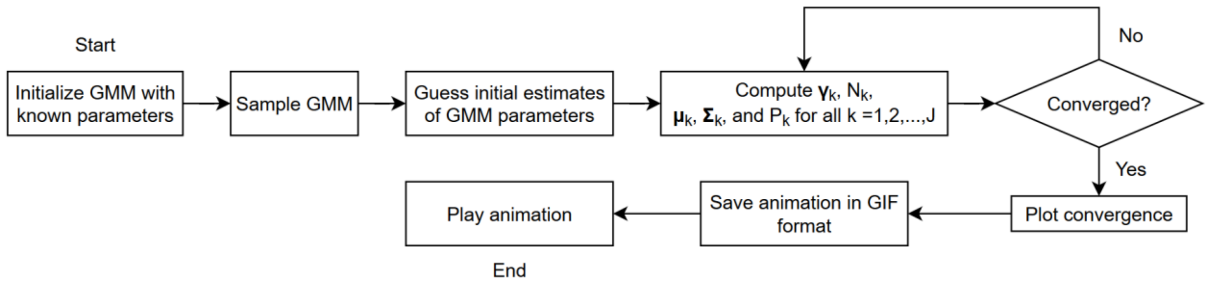
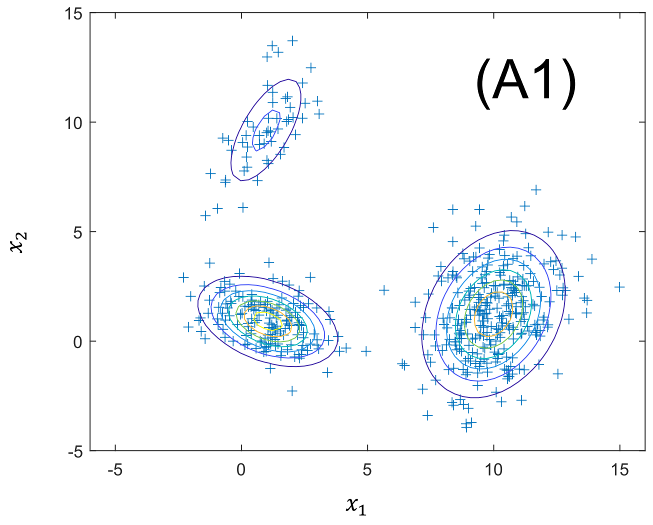

Start
Initialize GMM with known parameters
Sample GMM
Guess initial estimates of GMM parameters
Compute $\boldsymbol{\gamma}_k$, N_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, and P_k for all k =1,2,...,J
Converged?
No
Yes
Plot convergence
Save animation in GIF format
Play animation
End

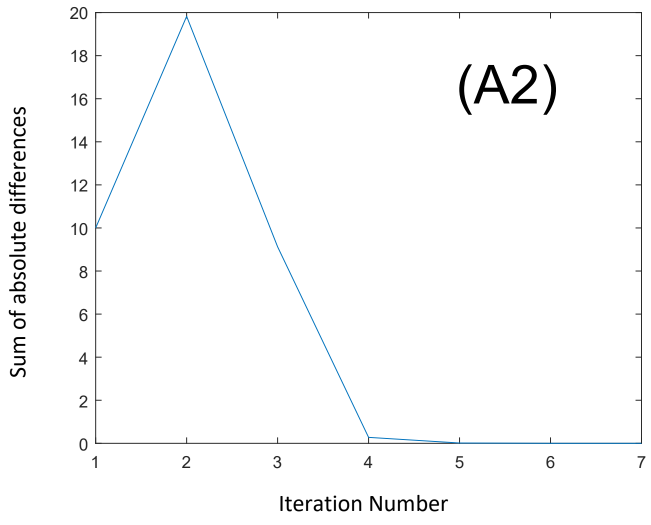
Figure 1 – Program Workflow

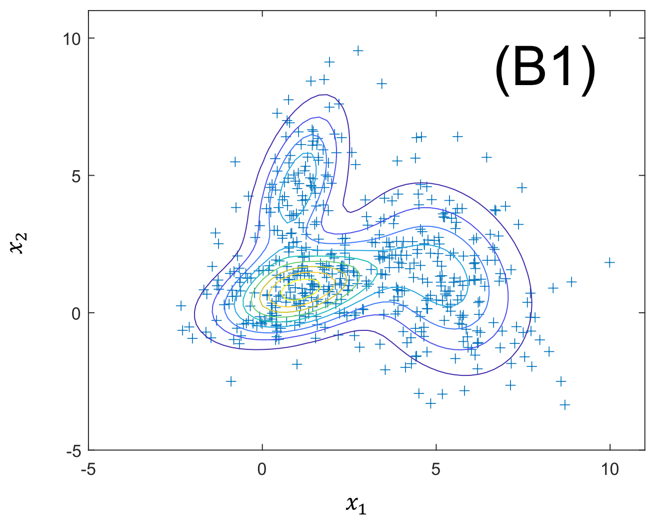
Figure 1 shows the overall workflow that was implemented in the `main.m` file in MATLAB. An original set of parameters $\{N\} \cup \{P_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}_{j=1}^{J}$ is first used to create and sample $N = 500$ points from a GMM. Next, initial guesses of the parameters are used to compute the vector $\boldsymbol{\gamma}_k$, which contains γ_{ik} values for $i = 1,2,\dots,N$, and the samples from the GMM are then used to compute the other parameters $N_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$, and P_k for all GMM components.

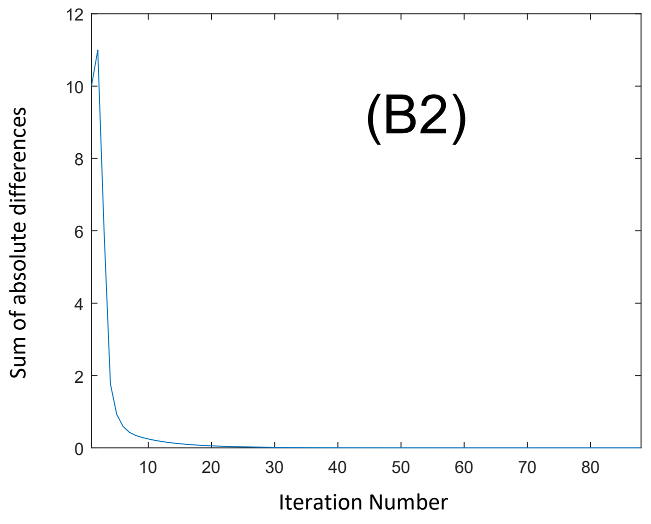
Once these new parameters are computed, the sum of absolute differences between these new parameters and old parameters is computed and compared to a threshold to check if these parameters are converging to specific values. Note that these specific values are not necessarily correct. If they converge, then a convergence plot, which shows the variation of the sum of absolute differences of the parameters between consecutive iterations, is displayed. Finally, a collection of contour plots for each set of parameters at each iteration step is animated, and the resulting animation is saved in a local file called "GMM.gif" in the working directory, and the animation is displayed in a separate figure. As previously mentioned, the data that is used to estimate the GMM parameters was manually generated. Additionally, different initial guesses were used to measure their effect on the convergence and accuracy of the EM algorithm.

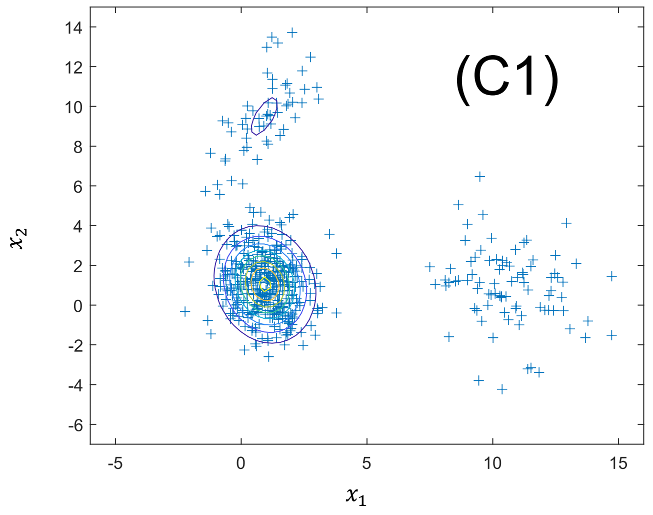
3. Results and discussion

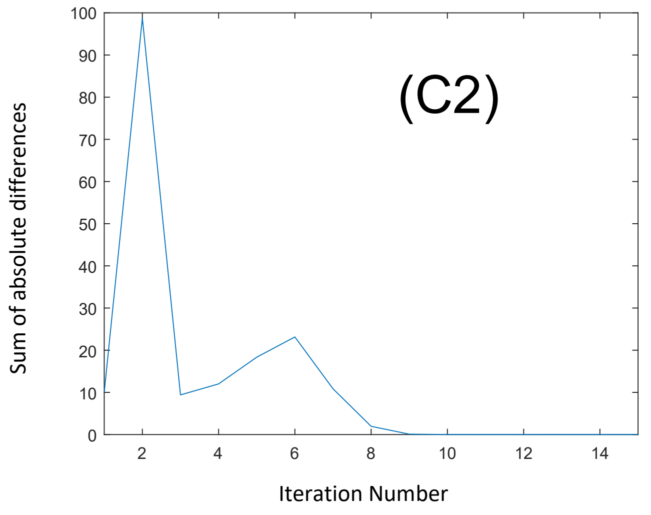












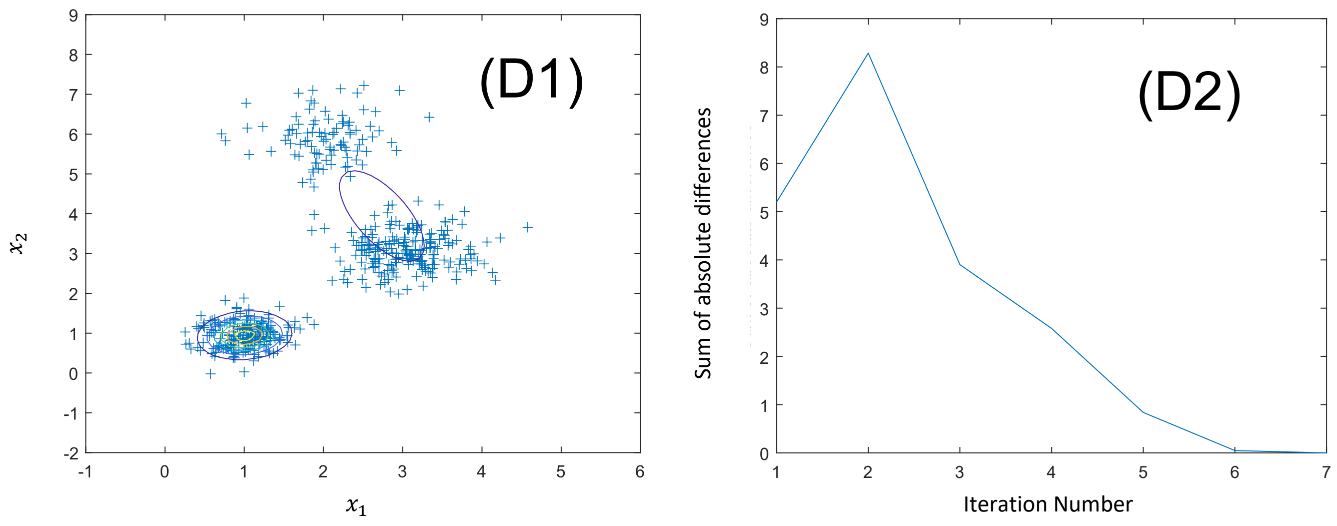


Figure 2 – Contour plots of estimated GMMs with sampled vectors overlaid and their associated convergence plots for the EM algorithm

The original and estimated GMM parameters associated with each plot in Figure 2 are shown in Table 1 in MATLAB notation. Note that values are rounded to two decimal places.

Table 1 – Original parameters and GMM parameters estimated using the EM algorithm

Plot Label		P_1	P_2	P_3	$\boldsymbol{\mu}_1$	$\boldsymbol{\mu}_2$	$\boldsymbol{\mu}_3$	$\boldsymbol{\Sigma}_1$	$\boldsymbol{\Sigma}_2$	$\boldsymbol{\Sigma}_3$	SAE
A1	Orig.	0.3	0.6	0.1	[1,1]	[10,1]	[1,10]	[2,-0.5; -0.5,1]	[2,0.8; 0.8,4]	[1,0.9; 0.9,3]	2.57
	Est.	0.31	0.59	0.1	[1.05,0.9]	[10,1.23]	[0.97,9.64]	[1.88,-0.54; -0.54,1.03]	[2.08,0.75; 0.75,3.76]	[1.15,1.28; 1.28,3.22]	
B1	Orig.	0.4	0.4	0.2	[1,1]	[5,1]	[1,5]	[2,0.5; 0.5,1]	[2,-0.8; -0.8,4]	[1,0.9; 0.9,3]	2.91
	Est.	0.4	0.4	0.2	[1.14,0.82]	[5.21,1.22]	[1.02,4.62]	[2.13,0.62; 0.62,1.03]	[2.15,-0.74; -0.74,4.09]	[0.66,0.79; 0.79,3.44]	
C1	Orig.	0.7	0.2	0.1	[1,1]	[10,1]	[1,10]	[1,-0.25; -0.25,2]	[3,-0.8; -0.8,5]	[1,0.9; 0.9,3]	52.3
	Est.	0.17	0.11	0.72	[10.67,0.9]	[0.92,9.5]	[0.94,1.04]	[2.53,-0.65; -0.65,3.54]	[1.19,1.45; 1.45,3.66]	[0.95,-0.19; -0.19,2.06]	
D1	Orig.	0.4	0.4	0.2	[1,1]	[3,3]	[2,6]	[0.1,0; 0,0.1]	[0.2,0; 0,0.2]	[0.3,0; 0,0.3]	2.79
	Est.	0.4	0.6	-	[1,0.94]	[2.74,3.94]	-	[0.09,0.01; 0.01,0.1]	[0.43,-0.61; -0.61,1.95]	-	

In plot A1 in Figure 2, initial parameter estimates were close to the original parameters. Because of this, the EM algorithm converged quickly in 7 steps. Also, as shown in Table 1, most estimated values are accurate to 1 significant figure, with a relatively small Sum of Absolute Errors (SAE).

In plot B1 in Figure 2, clusters were intentionally put close together to assess the EM algorithm's ability to discriminate. Because the clusters were close together, it took 95 steps (longer) for the EM algorithm to converge. Again, most estimated parameters were accurate to 1 significant figure, with a slightly larger SAE.

In plot C1 in Figure 2, initial parameter estimates were very different to the original parameters to test the effect that the initial estimates have on the final estimates. While the EM algorithm converged in only 15 steps, the SAE is large. However, looking closer at the parameter estimates, it seems that the parameters converged to the "wrong" order of parameters. For example, the estimate of P_1 converged to the true value of P_3, the estimate of $\boldsymbol{\mu}_2$ converged to the true value of $\boldsymbol{\mu}_3$, and so on. In other words, parameter estimates were mixed up. This could be attributed to the fact that the EM algorithm converges to local and not global maxima.

In plot D1 in Figure 2, it was intentionally assumed that there were only 2 mixing components instead of 3 in the GMM to test the effect that this has on the EM algorithm. The resulting plot and results in Table 1 show that the EM algorithm tries to “fit” two clusters using a single component to compensate. It can be seen that the estimated value of $\boldsymbol{\mu}_2$ is approximately the average of the true values of $\boldsymbol{\mu}_2$ and $\boldsymbol{\mu}_3$.

4. Conclusion

In this report, the EM algorithm was used to estimate the parameters of a GMM given only samples from the GMM. Through experiments, it was shown that most parameter estimates resulting from the EM algorithm are accurate to 1 significant figure. However, it was also shown that the EM algorithm could converge to the wrong order of parameter estimates. Therefore, the EM algorithm is only as good as the initial parameter estimates it is given. Additionally, it was shown that assuming the wrong number of components in the GMM could lead to fitting multiple clusters using a single component and vice versa.

Note however that due to page limits, not enough experiments were conducted to fully characterize the EM algorithm’s dependency on initial parameter estimates, cluster proximity, and other hyper-parameters. Therefore, the reader should conduct more experiments to explore these relationships further. An important concept that I learned from the derivations in section 1 of this report is that vicious dependency cycles could be resolved using a successive approximation method like the EM algorithm.

5. References

[1] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, Fourth Edition*. Burlington, MA: Academic Press (Elsevier), 2009.

[2] C. M. Bishop, "Pattern Recognition and Machine Learning," in *Pattern Recognition and Machine Learning*, Springer, 2011, pp. 430 - 455.

[3] M. I. Jordan, "Stat 260/CS 294 Bayesian Modelling and Inference," Spring 2010. [Online]. Available: https://people.eecs.berkeley.edu/~jordan/courses/260-spring10/other-readings/chapter13.pdf. [Accessed 14 March 2020].