A. Appendix

A.1. Convergence in Log-likelihood

We experiment with all the four algorithms (EM, Anti-annealing, BFGS, ECG) on the three datasets explained in the main paper (section 5.1). We run each algorithm 10 times on each dataset and observe how each of these algorithms converges in terms log-likelihood. The average log-likelihood values are plotted in Figure A.1. For closer examination, we also plot the zoomed in versions for each associated plots (bottom row). We observe that all the four algorithms converge fairly quickly to values close to the optimum log-likelihood, even though they are far away from true parameters in the parameter space. The reason for such behavior is that the log-likelihood values are dominated by the larger clusters, whose parameters are learned quickly. Although regular EM exhibits slow convergence for parameters of smaller clusters, these smaller clusters have relatively less impact on log-likelihood values. The zoomed-in plots show that the deterministic anti-annealing method achieves slightly better average log-likelihood than the other methods, because it can learn the parameters of smaller clusters fast and more accurately. The non-monotonic behavior of the log-likelihood values for the deterministic anti-annealing method is due to the change in temperatures, which essentially change the objective function being optimized.

We also plot the distribution of the final log-likelihood values over the 10 repeated runs (with random initialization) for all four algorithms. The results are shown in Figure A.1. While there are only minor differences in terms of the final log-likelihood values, we see that the deterministic anti-annealing method is more stable and consistently provides slightly better average log-likelihood values.

A.2. Gradients used by ECG and BFGS

The Conjugate Gradient and Quasi-Newton methods (BFGS) are known to outperform first order gradient-based methods for special cases when the objective function is elongated, and the conjugate direction is a better direction than the steepest gradient direction. These methods do not need to explicitly compute the Hessian, and can work by only computing gradient. We derive here the gradient functions that are used by both the conjugate gradient and BFGS methods.

Let Q denote the complete log-likelihood function:

$$Q = \sum_{t=1}^{N} \sum_{i=1}^{K} h_i(t) \log(\alpha_i p(\mathbf{x}_t | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)) \tag{1}$$

where $h_i(t)$ is the responsibility of the i^{th} Gaussian component for the data point $\mathbf{x}_t$. The EM algorithm automatically satisfies several constraints on the parameter space: $\alpha_i \geq 0$, $\sum_i \alpha_i = 1$, and $\boldsymbol{\Sigma} \succeq 0$. To ensure that the ECG algorithm satisfies the same set of constraints, Salakhutdinov et al. (2003) propose to re-parameterize the model parameters:

$$\alpha_j = \frac{e^{\lambda_j}}{\sum_l e^{\lambda_l}} \tag{2}$$

$$\boldsymbol{\Sigma}_j = \mathbf{L}_j \mathbf{L}_j^* \tag{3}$$

where $\mathbf{L}_j$ is the upper triangular matrix obtained by Cholesky decomposition of $\boldsymbol{\Sigma}_j$. Under this re-parameterization, the gradient values can be computed in a straight forward manner:

$$\frac{\partial Q}{\partial \lambda_j} = \left(\sum_{t=1}^{N} h_j(t) \right) - N\alpha_j \tag{4}$$

$$\frac{\partial Q}{\partial \boldsymbol{\mu}_j} = \sum_{t=1}^{N} h_j(t) \boldsymbol{\Sigma}_j^{-1} (\mathbf{x}_t - \mu_j) \tag{5}$$

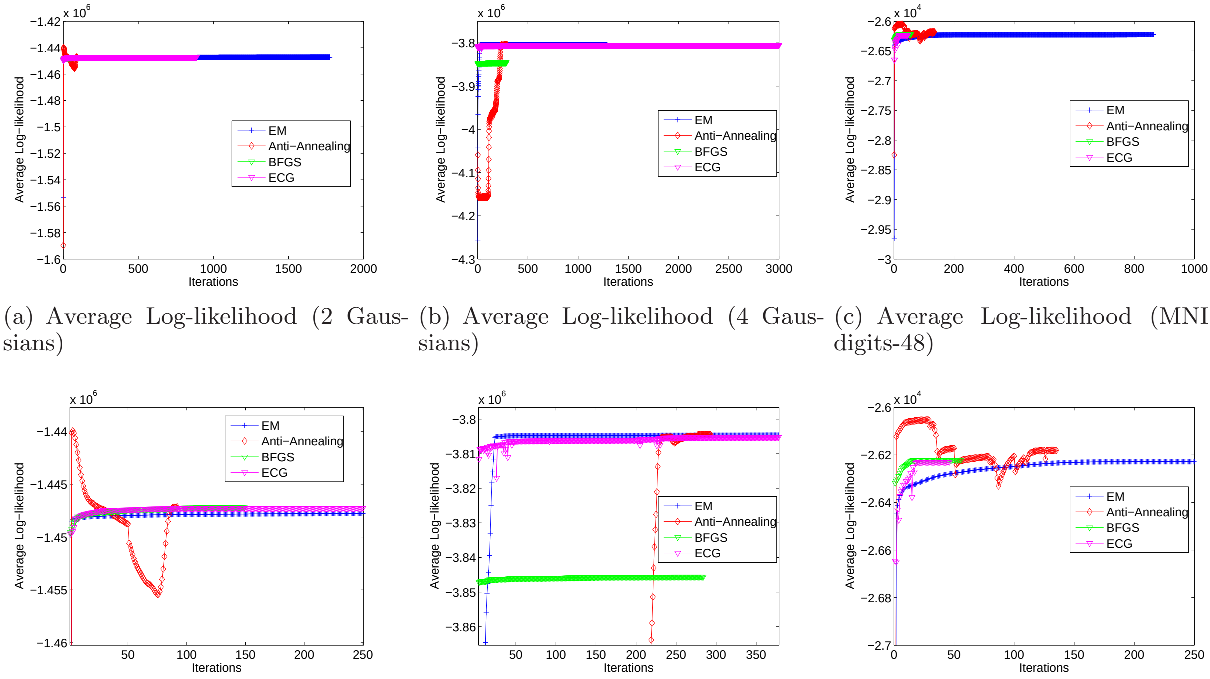
$$\begin{aligned}\frac{\partial Q}{\partial L_j} = &\sum_{t=1}^{N} h_j(t) \boldsymbol{\Sigma}_j^{-1} \\ &\left[(\mathbf{x}_t - \boldsymbol{\mu}_j)(\mathbf{x}_t - \boldsymbol{\mu}_j)^T - \boldsymbol{\Sigma}_j \right] \boldsymbol{\Sigma}_j^{-1} \mathbf{L}_j\end{aligned} \tag{6}$$

A.3. Application to Image Segmentation

We evaluate the performance of the deterministic anti-annealing algorithm on image segmentation task. We apply all four algorithms on the publicly available St. Paulia flower data (Fraley et al., 2005) (Figure 3), which has lots of variations in colors. Please note the tiny yellow flower centers that we expect to form a small cluster. [1]

We apply all four algorithms to cluster this dataset. The raw RGB values for each pixel are used as features, and no preprocessing was performed. The total number of pixels is 81472 (image resolution 304 × 268). We empirically set the number of clusters $K = 40$, so that clustering methods can detect subtle variations in colors of leaves and flower centers. For anti-annealing, we used the same annealing schedule as for the 4-Gaussian

[1] The dataset can be downloaded from www.cac.science.ru.nl/people/rwehrens/suppl/mbc.html



(a) Average Log-likelihood (2 Gaussians)

(b) Average Log-likelihood (4 Gaussians)

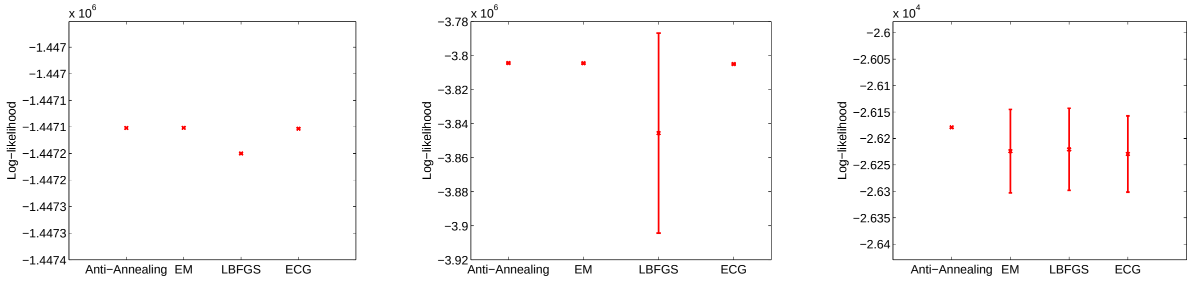
(c) Average Log-likelihood (MNIST digits-48)

(d) Zoomed-in version (2 Gaussians)

(e) Zoomed-in version (4 Gaussians)

(f) Zoomed-in version (MNIST digits-48)

Figure 1. The changes in average log-likelihood with iterations for all the four algorithms. Each plot in the bottom row presents a zoomed in version for the corresponding plot on the top row.



(a) Distribution of final log-likelihood (2 Gaussians)

(b) Distribution of final log-likelihood (4 Gaussians)

(c) Distribution of final log-likelihood (MNIST digits-48)

Figure 2. The distribution (mean and standard deviation) of final log-likelihood values for each of the four algorithms, over 10 repeated runs with random initialization.

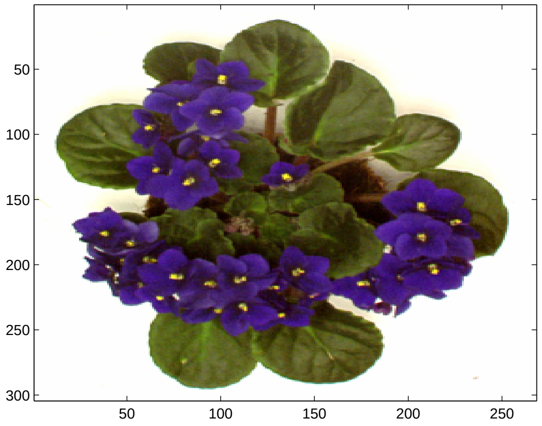


Figure 3. The original St. Polya flower image.

dataset: $\beta = [0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.0]$. For fair comparison among the algorithms, we set the maximum number of allowed iterations to 500, so that each algorithm terminates either if the convergence criteria is satisfied, or if the number of iterations reached 500.

Due to lack of ground truth model parameters, we first visually validate the clustering results. We represent each segment by the mean color of the associated cluster and reconstruct a summary image where each pixel's RGB values are replaced by that of its cluster mean. We also estimate the mean squared error between the original image ($\mathbf{I}_0$) and the segmented summary image ($\mathbf{I}_s$):

$$MSE(\mathbf{I}_0, \mathbf{I}_s) = \sqrt{\sum_i \sum_j [\mathbf{I}_0(i,j) - \mathbf{I}_s(i,j)]^2}$$

The results are shown in Figure 4. Please note the small yellow cluster centers. While our anti-annealing method was able to represent them by a cluster whose mean is yellow, all the other methods resulted in a greenish cluster. We also performed numerical evaluation by running the algorithms 10 times with random initialization and compare the average mean-squared error ($MSE(\mathbf{I}_0, \mathbf{I}_s)$) between the original image I_0 and the segmented summary image I_s. The results are also presented in Figure 4.

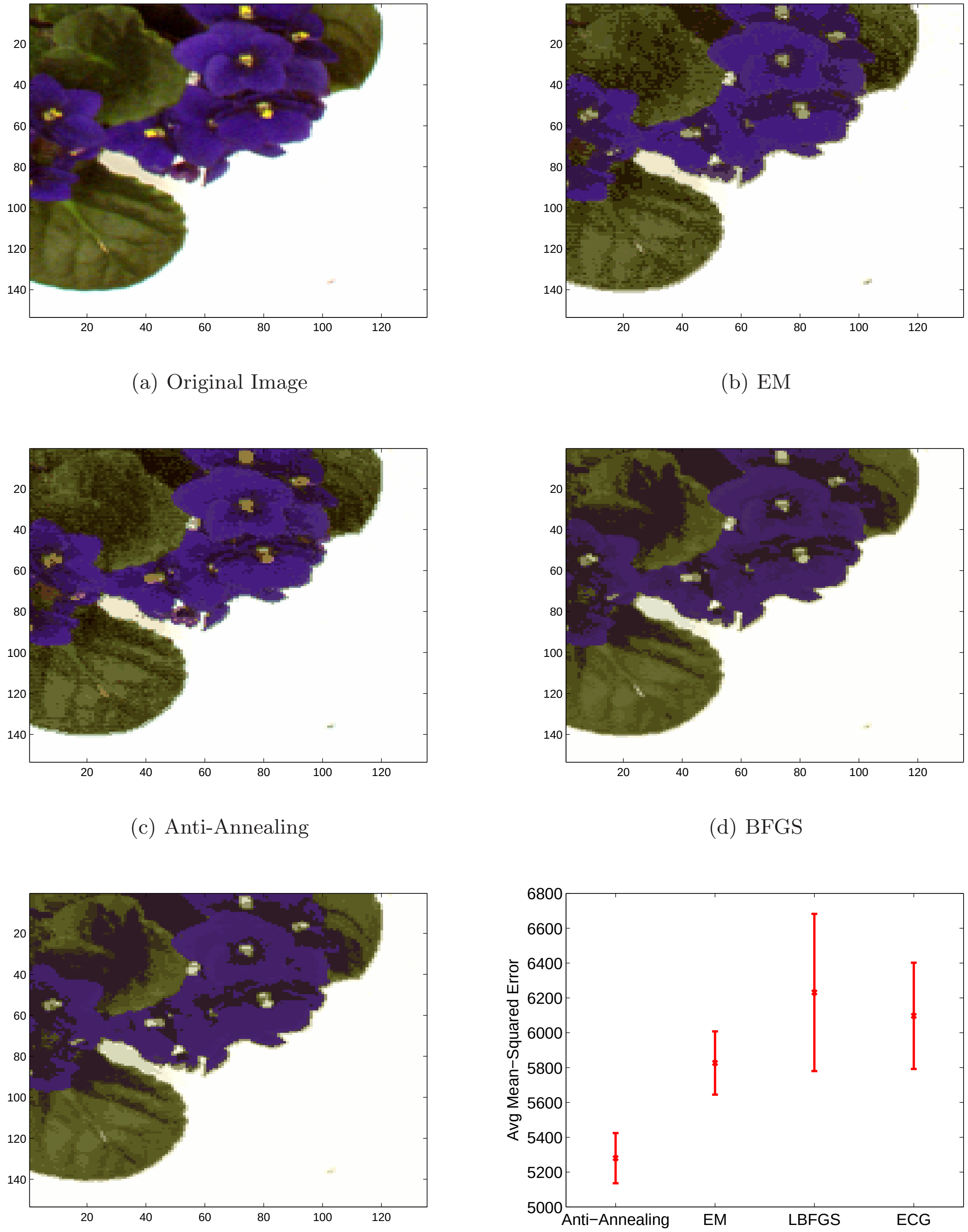
A.4. Complexity per Iteration

The complexity of each iteration of EM and Deterministic Anti-annealing EM is almost the same. Both requires $O(NKd^2)$ operations, where N is the number of data points, K is the number of clusters, and d is the number of dimensions. The Deterministic Anti-annealing method needs to estimate the additional β-exponents of posterior probabilities, but that does not change the asymptotic complexity. On the other hand, each iteration of BFGS and ECG requires line-search to ensure monotonic convergence. Therefore, each iteration of BFGS and ECG is little slower than that of EM and Deterministic Anti-annealing EM.

References

Fraley, C., Raftery, A., and Wehrens, R. Incremental model-based clustering for large datasets with small clusters. *Journal of Computational and Graphical Statistics*, 14(3):529–546, 2005.

Salakhutdinov, R., Roweis, S., and Ghahramani, Z. Optimization with EM and Expectation-Conjugate-Gradient. *in Proceedings of ICML*, 20:672–679, 2003.



(a) Original Image

(b) EM

(c) Anti-Annealing

(d) BFGS

(e) ECG

(f) Mean-squared Error

Figure 4. Clustering results on the St. Polya flower data, using all the four algorithms. The segmented images are summarized by replacing pixel RGB values by associated cluster mean. Please note the yellow cluster centers. The last sub-figure shows the mean-squared error, averaged over 10 repeated runs.