Local Gradient Descent Methods for GMM Simplification

Abstract

Gaussian mixture model simplification is a powerful technique for reducing the number of components of an existing mixture model without having to re-cluster the original data set. Instead, a simplified GMM with fewer components is computed by minimizing some distance metric between the two models. In this paper, we derive an analytical expression for the difference between the probability density functions of two GMMs along with its gradient information. We minimize the objective function using gradient descent and K-means. Both synthetic and non-synthetic test cases are used in the experiments.

1. Introduction

Gaussian mixture models (GMMs) are a powerful tool for estimating the probability density function of a random variable x. Mixture models have found a wide range of applications in different domains such as speaker recognition, image processing, finances, etc.

The density of the known mixture model f at a point $x \in \mathbb{R}^d$ is

$$\begin{aligned} f(x) &= \sum_{k=1}^{K_f} \pi_{k,f} \eta(x, \mu_{k,f}, \Sigma_{k,f}), \\ \sum_{k=1}^{K_f} \pi_{k,f} &= 1, \quad \pi_{k,f} \geq 0, \end{aligned} \tag{1}$$

where η is the normal distribution centered around μ with a symmetric positive-definite covariance matrix Σ

$$\eta(x, \mu, \Sigma) = |(2\pi)^d \Sigma|^{-\frac{1}{2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}. \tag{2}$$

Learning the mixture model is a clustering problem often done by estimating the number of components K_f and iteratively maximizing a log-likelihood quantity

$$\ln f(X) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K_f} \pi_{k,f} \eta(x_n, \mu_{k,f}, \Sigma_{k,f}) \right\} \tag{3}$$

over a set of N independent and identically distributed features. The parameters are estimated to a local optimum using the expectation-maximization (EM) algorithm (Dempster et al., 1977).

Since it is difficult to specify the number of components a priori, a GMM may *over-fit* the underlying distribution. That is, the number of parameters is too large and so we would like to reduce or simplify the model by decreasing the number of components K_f. One can recompute the model f with fewer components using the standard EM algorithm but this is costly when the data set is large. Instead, we use Gaussian simplification to obtain new parameters for a mixture model g that approximates f without accessing the original feature space.

Gaussian simplification is a process that finds a target mixture model g with $K_g < K_f$ components that is *similar* to mixture model f. The density of the target mixture model g at a point $x \in \mathbb{R}^d$ is

$$\begin{aligned} g(x) &= \sum_{k=1}^{K_g} \pi_{k,g} \eta(x, \mu_{k,g}, \Sigma_{k,g}), \\ \sum_{k=1}^{K_g} \pi_{k,g} &= 1, \quad \pi_{k,g} \geq 0. \end{aligned} \tag{4}$$

In prior works, the simplification problem is often posed in terms of relative entropy between the mixture models g and f. The Kullback-Leibler(KL) divergence compares multiple distributions that may be optimized by Bregman K-means in (Nielsen et al., 2009), (Garcia et al.). A closed form of the Jensen-Rényi divergence is minimized in (Hamza & Krim, 2003), (Wang et al., 2009). The Unscented Transform Approximation (UTA) criterion approximates the KL divergence between GMMs which can be maximized via an EM-like algorithm (Goldberger et al., 2008).

We define similarity as the χ^2 distance between the probability distribution functions (PDFs) of mixture

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

models f and g. In section 2, we derive the approximation error between mixture models based on this similarity measurement. In section 3, we look at methods for minimizing the approximation error using gradient information. In section 4, we run the methods on both synthetic data generated by pre-defined distributions and real-world features extracted from speech data.

2. χ^2 Distance

The χ^2 distance is the approximation error or the squared difference between the PDFs of the mixture models f and g sampled across the entire domain (Hall & Hicks, 2004). The integral over the squared difference is our objective function, and has the form

$$\begin{aligned} F(\theta) &= \int_{-\infty}^{\infty} (f(x) - g(x,\theta))^2 \mathrm{d}x \\ &= \int_{-\infty}^{\infty} f(x)^2 - 2f(x)g(x,\theta) + g(x,\theta)^2 \mathrm{d}x. \end{aligned} \tag{5}$$

Eqn. 5 leads to a computable form as the products of Gaussian components are unnormalized Gaussians (Appendix 6.1). By integrating each term over the entire domain, we are left with the weighted summation of unnormalized coefficients which are themselves Gaussians

$$\begin{aligned} \int_{-\infty}^{\infty} f(x)^2 \mathrm{d}x &= \sum_i^{K_f} \sum_j^{K_f} \pi_{i,f}\pi_{j,f} z_{i,j}^f, \\ z_{i,j}^f &= \eta(\mu_{i,f}, \mu_{j,f}, \Sigma_{i,f} + \Sigma_{j,f}), \\ \int_{-\infty}^{\infty} -2f(x)g(x,\theta)\mathrm{d}x &= -2\sum_i^{K_f} \sum_j^{K_g} \pi_{i,f}\pi_{j,g} z_{i,j}^{fg}, \\ z_{i,j}^{fg} &= \eta(\mu_{i,f}, \mu_{j,g}, \Sigma_{i,f} + \Sigma_{j,g}), \\ \int_{-\infty}^{\infty} g(x)^2 \mathrm{d}x &= \sum_i^{K_g} \sum_j^{K_g} \pi_{i,g}\pi_{j,g} z_{i,j}^g, \\ z_{i,j}^g &= \eta(\mu_{i,g}, \mu_{j,g}, \Sigma_{i,g} + \Sigma_{j,g}). \end{aligned} \tag{6}$$

We may simplify the notation by writing in matrix-vector form. The set of weights for mixtures f and g are treated as vectors. The unnormalized coefficients populate the matrices Z. Note that the first term remains constant as it consist only of elements from mixture f. The objective function is equivalent to

$$\begin{aligned} &F(\theta) = v_f^T Z^f v_f - 2v_f^T Z^{fg} v_g + v_g^T Z^g v_g, \\ &\sum_{k=1}^{K_g} v_{k,g} = 1, \quad v_{k,g} \geq 0, \\ &v_f = [\pi_{1,f}, \dots, \pi_{K_f,f}]^T, \quad v_g = [\pi_{1,g}, \dots, \pi_{K_g,g}]^T, \\ &Z_{ij}^f = \eta(\mu_{i,f}, \mu_{j,f}, \Sigma_{i,f} + \Sigma_{j,f}), \\ &Z_{ij}^{fg} = \eta(\mu_{i,f}, \mu_{j,g}, \Sigma_{i,f} + \Sigma_{j,g}), \\ &Z_{ij}^g = \eta(\mu_{i,g}, \mu_{j,g}, \Sigma_{i,g} + \Sigma_{j,g}). \end{aligned} \tag{7}$$

3. Minimizing $F(\theta)$

Directly minimizing the approximation error $F(\theta)$ in Eqn. 7 leads to a non-linear system that is difficult to solve. However, a first-order iterative method such as gradient descent is possible. Recall that gradient descent finds a local minimum by moving in the negative gradient direction

$$\theta^{t+1} = \theta^t - \gamma \nabla F(\theta). \tag{8}$$

To compute gradients, we differentiate $F(\theta)$ w.r.t. each parameter. For convenience, we use vector notation to represent the entire set of weight parameters v_g in the mixture model. For a component l, its mean and covariance parameters are represented by the vector $\mu_{l,g}$ and the symmetric positive definite matrix $\Sigma_{l,g}$. Note that $F(\theta)$ is quadratic in terms of the weight parameters v_g and so its partial derivative is linear. Thus, we can normalize the weights to sum to 1 at the end of each step without changing the sign of the gradient. The partial derivatives (Appendix 6.3) are

$$\begin{aligned} &\frac{\partial F}{\partial v_g} = -2(v_f^T Z^{fg} - v_g^T Z^g), \\ &\frac{\partial F}{\partial \mu_{l,g}} = \pi_{l,g} \left(\sum_i^{k_f} \pi_{i,f}\eta_{i,l}\mu_{l,i}^{gf}\Sigma_{i,l}^{fg} - \sum_j^{k_g} \pi_{j,g}\eta_{l,j}\mu_{l,j}^{gg}\Sigma_{l,j}^{gg} \right), \\ &\frac{\partial F}{\partial \Sigma_{l,g}} = \pi_{l,g} \left(\sum_i^{k_f} \pi_{i,f}\eta_{i,l}\Sigma_{i,l}^{fg} \left(I - (\mu_{i,l}^{fg})^T \mu_{i,l}^{fg} \Sigma_{i,l}^{fg} \right) \right. \\ &\left. - \sum_{j \neq l}^{k_g} \pi_{j,g}\eta_{l,j}\Sigma_{l,j}^{gg} \left(I - (\mu_{l,j}^{gg})^T \mu_{l,j}^{gg} \Sigma_{l,j}^{gg} \right) \right) - \frac{\pi_{l,g}^2 \Sigma_{l,g}^{-1}}{2\sqrt{(2\pi)^d |2\Sigma_{l,g}|}}, \\ &\mu_{l,i}^{gf} = (\mu_{l,g} - \mu_{i,f})^T, \quad \Sigma_{i,l}^{fg} = (\Sigma_{i,f} + \Sigma_{l,g})^{-1}, \\ &\eta_{i,l} = \eta(\mu_{i,f}, \mu_{l,g}, \Sigma_{i,f} + \Sigma_{l,g}), \\ &\eta_{l,j} = \eta(\mu_{l,g}, \mu_{j,g}, \Sigma_{l,g} + \Sigma_{j,g}). \end{aligned} \tag{9}$$

To find a suitable γ coefficient, we minimize the functional $F(\theta + \gamma v)$ where v is the line search direction. The first derivative with respect to γ can be approximated by a truncated Taylor expansion

$$
\begin{aligned}
F(\theta+\gamma v) &\approx F(\theta) + \gamma \frac{\partial}{\partial\theta} F(\theta)^T v + \frac{\gamma^2}{2} \frac{\partial^2}{\partial\theta^2} F(\theta)^T v, \\
\frac{\partial}{\partial\gamma} F(\theta+\gamma v) &\approx \frac{\partial}{\partial\theta} F(\theta)^T v + \gamma \frac{\partial^2}{\partial\theta^2} F(\theta)^T v.
\end{aligned} \tag{10}
$$

Explicitly computing the second derivative Hessian matrix is expensive. Instead, we use a secant method for approximating the second derivative from a general line search step in non-linear conjugate gradients optimization (Shewchuk, 1994). Setting the first partial derivative to 0, we solve for the γ coefficient

$$
\begin{aligned}
\frac{\partial^2}{\partial\theta^2} F(\theta) &\approx \frac{\frac{\partial F(\theta+\sigma v)}{\partial\theta} - \frac{\partial F(\theta)}{\partial\theta}}{\sigma} \quad \text{for small } \sigma, \\
0 &= \frac{\partial}{\partial\theta} F(\theta)^T v + \gamma \frac{\partial^2}{\partial\theta^2} F(\theta)^T v, \\
\gamma &= -\sigma \frac{\nabla F(\theta)^T v}{\nabla F(\theta+\sigma v) - \nabla F(\theta)^T v}, \quad v = \nabla F(\theta).
\end{aligned}
$$

The step size σ is initially an arbitrarily small value and is set to the previous $|\gamma|$ in subsequent iterations. In practice, this secant approximation for line searching is only used during the first few iterations to quickly move towards a local minima. We revert back to normal gradient descent with a fixed $\gamma = 10^{-(d/3)}$ for v_g, μ_g, and $\gamma = 10^{-(d/3)-1}$ for Σ_g parameters.

It is also possible to perform local gradient descent on similar components in order to decrease run-time but with greater approximation error. The components of mixture model f into K_g can be partitioned into disjoint sets and local fitted for each component in mixture model g (Zhang & Kwok, 2007). We consider a similar approach that modifies the well known K-means algorithm to run over mixture model g's parameter space. This K-means algorithm alternates between an assignment followed by one or more update steps.

1. **Assignment step:** Assign each of the K_f components in the mixture model f to the most similar components in the mixture model g.

$$
\begin{aligned}
S_i^{(t)} = \Big\{ & k_{j,f} : D(k_{j,f}, k_{i,g}^{(t)}) \le D(k_{j,f}, k_{i^*,g}^{(t)}) \\
& \text{for all } i^* = 1, \dots, K_g.
\end{aligned} \tag{11}
$$

   The distance D between mixture components may take on alternative forms such as the average Kullback-Leibler, Bhattacharyya, and generalized Rényi divergences. In this paper, we use the similar χ^2 distance formulation between pair-wise components.

$$
\begin{aligned}
D(k_{j,f}, k_{i,g}^{(t)}) &= \int_{-\infty}^{\infty} (k_{j,f} - k_{i,g}^{(t)})^2 \mathrm{d}x \\
&= \pi_{j,f}^2 z_{j,j}^f - 2\pi_{j,f}\pi_{i,g} z_{j,i}^{fg} + \pi_{i,g}^2 z_{i,i}^g.
\end{aligned} \tag{12}
$$

2. **Update step:** Modify the components of mixture model g by performing local gradient descent.

$$
\begin{aligned}
\theta_i^t &= \theta_i^{t-1} - \gamma \nabla F_i^*(\theta) \\
\nabla F_i^*(\theta) &= \nabla F(\theta, \pi_f^i, \pi_g^i), \quad
\begin{aligned}
\pi_{j,f}^i &= \begin{cases} 0, & j \notin S_i^{(t)} \\ \pi_{j,f}, & j \in S_i^{(t)} \end{cases} \\
\pi_{j,g}^i &= \begin{cases} 0, & j \neq i \\ \pi_{j,g}, & j = i \end{cases}
\end{aligned}
\end{aligned} \tag{13}
$$

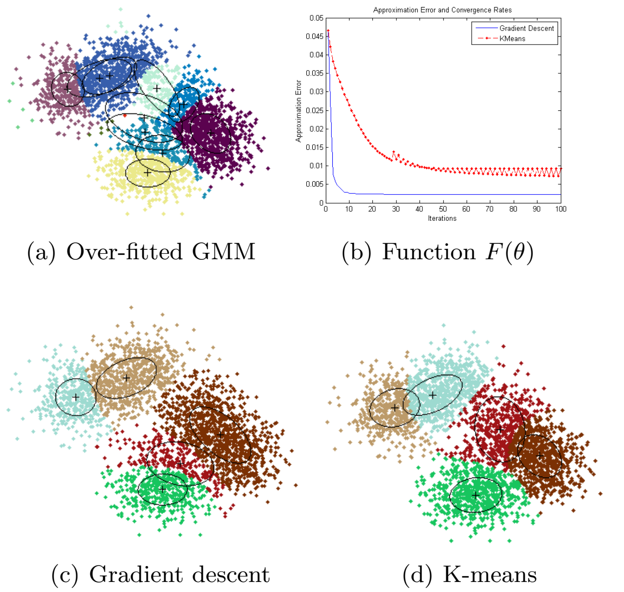
   The gradient $\nabla F_i^*(\theta)$ is now unique as each component $k_{i,g}$ is mutually independent and can only see the assigned components $S_i^{(t)}$ in the mixture model f.

The algorithm terminates when no new assignments are made and the local components have converged.

4. Experiments

To obtain the source mixture model f, we perform EM on both synthetic and real-world data. In the synthetic case, we generate a random set of $\frac{K_f}{2}$ weighted Gaussian distributions and then randomly sample N points from the distributions. This suggests that running EM on the source data for K_f cluster will produce an overfitted model that we can simplify. For the initial conditions of mixture model g, we suggest the K_g highest weighted components from mixture model f. This allows both gradient descent and K-means to start with a configuration that is likely to be close to the global minimum.

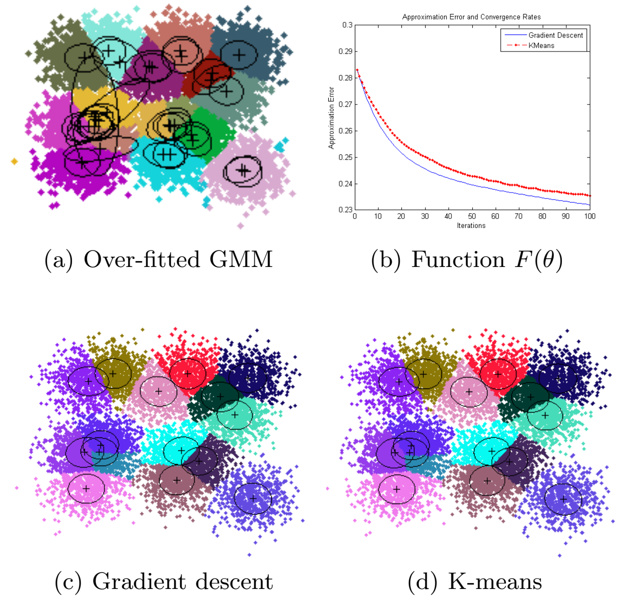
In Fig. 1, the local updates for K-means may cause the the assignment step to oscillate between two or more components. Gradient descent achieves the expected smaller approximation error than the K-means method. In Fig. 2 for higher dimensional data where the GMM components are more separated, the approximation error is less pronounced. The K-means routine performs 4 update steps for every assignment step and terminates 3 times faster than the gradient descent method.



(a) Over-fitted GMM (b) Function $F(\theta)$

(c) Gradient descent (d) K-means

Figure 1. Comparison of gradient descent and K-means on a 10 component mixture model simplified to 5. Original GMM generated from 3000 points sampled across 5 normal distributions of equal diagonal covariance, random mean, random weight, 2 dimensions.



(a) Over-fitted GMM (b) Function $F(\theta)$

(c) Gradient descent (d) K-means

Figure 2. Comparison of gradient descent and K-means on a 30 component mixture model simplified to 15. Original GMM generated from 10000 points sampled across 15 normal distributions of equal diagonal covariance, random mean, random weight, 10 dimensions. Graphs show a two dimensional slice of the GMMs and data.

For non-synthetic inputs, we work with speech data obtained from the NIST Speaker Recognition Evaluation (SRE) 2008 collection. The raw data has been transformed into 38 dimensional Mel-frequency cepstrum coefficients, extracted from 30ms frames with overlaps. These coefficients or speech features represent the short-term power spectrum of a sound and are shown to approximate the human auditory system's response (Ganchev et al., 2005). In speaker recognition, a common first step is to learn a Universal Background Model (UBM) that represents general, person-independent feature characteristics (Reynolds & Rose, 1995). This UBM is identical to a GMM that is trained over a large set of speaker features. In Fig. 3, the initial components have closely related means but with varying covariances. The K-means method oscillates wildly during certain assignment steps. In the cases of poor component assignment, performing local gradient descent may actually increase the approximation error.

5. Conclusions

We have shown that the analytical form of the difference between two PDFs of Gaussian mixture models can be directly used for model simplification. The partial derivatives derived from the analytical form can be applied to such techniques as gradient descent and K-means for minimization. The experimental results for synthetic data show that both techniques converge to local minimums for components that are well separated along the means. The experimental results for sound data where the components have locally close means are less conclusive for the K-means approach.

References

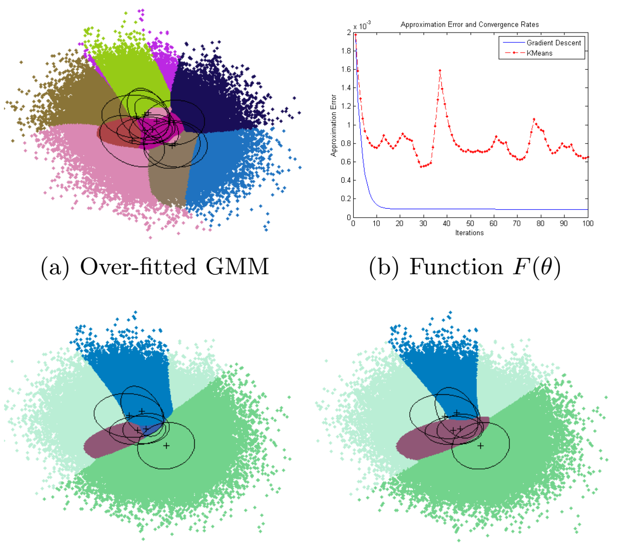
Dempster, A.P., Laird, N.M., and Rubin, D.B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B*, 39:1–38, 1977.

Ganchev, T., Fakotakis, N., and Kokkinakis, G. Comparative evaluation of various mfcc implementations on the speaker verification task. In *10th International Conference on Speech and Computer (SPECOM 2005)*, volume 1, pp. 191–194, 2005.

Garcia, V., Nielsen, F., and Nock, R. Hierarchical gaussian mixture models. In *ICASSP 2010*.

Goldberger, J., Greenspan, H., and Dreyfuss, J. Simplifying mixture models using the unscented transform. *IEEE Transactions Pattern Analysis Machine Intelligence*, 30:1496–1502, 2008.

Hall, P. M. and Hicks, Y. A method to add gaussian



(a) Over-fitted GMM (b) Function $F(\theta)$

(c) Gradient descent (d) K-means

Figure 3. Comparison of gradient descent and K-means on a 10 component mixture model simplified to 5. Original GMM generated from NIST SRE 2008 data, 146556 points across the first two dimensions.

mixture models. Technical report, Department of Computer Science, University of Bath, 2004.

Hamza, A.B. and Krim, H. Jensen-renyi divergence measure: theoretical and computational perspectives. In *IEEE International Symposium on Information*, pp. 257–257, 2003.

Nielsen, F., Garcia, V., and Nock, R. Simplifying gaussian mixture models via entropic quantization. In *17th European Conference on Signal Processing (EUSIPCO)*, 2009.

Reynolds, D. and Rose, R. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Trans. Speech Audio Processing*, 3:72–83, 1995.

Shewchuk, J. R. An introduction to the conjugate gradient method without the agonizing pain. Technical report, chool of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1994.

Wang, F., Syeda-Mahmood, T.F., Vemuri, B.C., Beymer, D., and Rangarajan, A. Closed-form jensen-renyi divergence for mixture of gaussians and applications to group-wise shape registration. In *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 648–655, 2009.

Zhang, K. and Kwok, J. T. Simplifying mixture models through function approximation. *Advances in Neural Information Processing Systems*, 17:1577–1584, 2007.

6. Appendix

6.1. Product of Multivariate-Gaussians

Theorem 6.1.1. *The product of two Gaussian $\eta(x, \mu_f, \Sigma_f)\eta(x, \mu_g, \Sigma_g)$ given the same random variable x is an unnormalized Gaussian. We assume that the covariance matrices are invertible and symmetric. A constructive proof is presented below.*

The product of Gaussians is derived from

$$\eta(x, \mu_f, \Sigma_f)\eta(x, \mu_g, \Sigma_g) = (2\pi)^{-d}|\Sigma_f \Sigma_g|^{-\frac{1}{2}} e^{\alpha},$$
$$\alpha = -\frac{1}{2}\left((x-\mu_f)^T \Sigma_f^{-1}(x-\mu_f) + (x-\mu_g)^T \Sigma_g^{-1}(x-\mu_g)\right). \tag{14}$$

The general form inside the exponential is

$$(x-y)^T C^{-1}(x-y) = x^T C^{-1} x - 2x^T C^{-1} y + y^T C^{-1} y. \tag{15}$$

For notation, let $a = \mu_f, A = \Sigma_f, b = \mu_g, B = \Sigma_g$

$$\begin{aligned}&(x-a)^T A^{-1}(x-a) + (x-b)^T B^{-1}(x-b)\\ &= x^T(A^{-1}+B^{-1})x - 2x^T(A^{-1}a + B^{-1}b) + a^T A^{-1} a\\ &+ b^T B^{-1} b.\end{aligned} \tag{16}$$

Completing the square from Eqn. 15, 16, we obtain the formulation

Let $C = (A^{-1} + B^{-1})^{-1}, \quad c = C(A^{-1}a + B^{-1}b),$

$$\begin{aligned}&x^T(A^{-1}+B^{-1})x - 2x^T(A^{-1}a + B^{-1}b) + a^T A^{-1} a\\ &+ b^T B^{-1} b\\ &= (x^T C^{-1} x - 2x^T C^{-1} c + c^T C^{-1} c)\\ &- c^T C^{-1} c + a^T A^{-1} a + b^T B^{-1} b\\ &= \left((x-c)^T C^{-1}(x-c)\right) - c^T C^{-1} c + a^T A^{-1} a + b^T B^{-1} b.\end{aligned} \tag{17}$$

Evaluating the remainder terms from Eqn. 17, we get

$$
\begin{aligned}
&-c^TC^{-1}c\\
&= -a^TA^{-1}CA^{-1}a - 2a^TA^{-1}CB^{-1}b - b^TB^{-1}CB^{-1}b,
\end{aligned}
$$

$$
\begin{aligned}
&a^TA^{-1}a + b^TB^{-1}b\\
&= a^TA^{-1}C(A^{-1}a + B^{-1}a) + b^TB^{-1}C(A^{-1}b + B^{-1}b),
\end{aligned}
$$

$$
\begin{aligned}
&-c^TC^{-1}c + a^TA^{-1}a + b^TB^{-1}b\\
&= a^TA^{-1}CB^{-1}a - 2a^TA^{-1}CB^{-1}b + b^TA^{-1}CB^{-1}b\\
&= (a-b)^T(A^{-1}CB^{-1})(a-b) \quad \text{From Eqn. 15}\\
&= (a-b)^T(A+B)^{-1}(a-b).
\end{aligned}
\tag{18}
$$

Substituting Eqn. 17, 18 back into Eqn. 14, we obtain the product of Gaussians

$$
\begin{aligned}
&\eta(x,\mu_f,\Sigma_f)\eta(x,\mu_g,\Sigma_g)\\
&= \left(\frac{(2\pi)^d|C|}{(2\pi)^d|C|}\right)^{\frac{1}{2}} (2\pi)^{-d}|AB|^{-\frac{1}{2}} e^{-\frac{1}{2}(x-c)^TC^{-1}(x-c)}\\
&e^{-\frac{1}{2}(a-b)^T(A+B)^{-1}(a-b)}\\
&= (2\pi)^{-\frac{d}{2}}|AB|^{-\frac{1}{2}}|C|^{\frac{1}{2}}e^{-\frac{1}{2}(a-b)^T(A+B)^{-1}(a-b)}\eta(x,c,C)\\
&= (2\pi)^{-\frac{d}{2}}|AC^{-1}B|^{-\frac{1}{2}}e^{-\frac{1}{2}(a-b)^T(A+B)^{-1}(a-b)}\eta(x,c,C)\\
&= \left((2\pi)^d|A+B|\right)^{-\frac{1}{2}} e^{-\frac{1}{2}(a-b)^T(A+B)^{-1}(a-b)}\eta(x,c,C)\\
&= \eta(a,b,A+B)\eta(x,c,C)\\
&= z_c\eta(x,c,C).
\end{aligned}
\tag{19}
$$

6.2. Matrix-Vector Derivatives

Lemma 6.2.1. *The partial derivative of a linear function is*

$$
\begin{aligned}
\frac{\partial a^Tx}{\partial x} &= \frac{\partial x^Ta}{\partial x} = a^T,\\
\frac{\partial Ax}{\partial x} &= \frac{x^TA}{x^T} = A.
\end{aligned}
$$

Lemma 6.2.2. *The partial derivative of a quadratic function is*

$$
\begin{aligned}
\frac{\partial x^TAx}{\partial x} &= \frac{\partial x^T}{\partial x}Ax + x^T\frac{\partial Ax}{\partial x} \quad \textit{by product rule}\\
&= 2x^TA \quad \textit{by Lemma 6.2.1 for symmetric } A.
\end{aligned}
$$

Lemma 6.2.3. *The partial derivative of a quadratic function with translated x is*

$$
\begin{aligned}
&\frac{\partial(a-x)^TA(a-x)}{\partial x} = \frac{\partial(x-a)^TA(x-a)}{\partial x}\\
&= \frac{\partial(x-a)^T}{\partial x}A(x-a) + (x-a)^T\frac{\partial A(x-a)}{\partial x}\\
&= 2(x-a)^TA \quad \textit{by Lemma 6.2.1 for symmetric } A.
\end{aligned}
$$

Lemma 6.2.4. *The partial derivative of matrix determinants with added A or B is*

$$
\begin{aligned}
|A+B| &= \sum_j(-1)^{i+j}(a_{ij}+b_{ij})M_{ij}\\
&\quad \textit{fixed } i\textit{, matrix } M \textit{ is minor of matrix } A{+}B,
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial|A+B|}{\partial a_{ij}} &= \frac{\partial|A+B|}{\partial b_{ij}}\\
&= (-1)^{i+j}M_{i,j} \quad \textit{is the cofactor matrix,}
\end{aligned}
$$

$$
\begin{aligned}
(A+B)^{-1} &= \frac{1}{|A+B|}adj(A+B)\\
&\quad \textit{Cramer's rule, adj(A+B) is adjoint}\\
&= \frac{1}{|A+B|}\left(\frac{\partial|A+B|}{\partial A}\right)^T\\
&\quad \textit{adj(A+B) is transpose of the cofactor matrix,}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial|A+B|}{\partial A} &= \frac{\partial|A+B|}{\partial B}\\
&= |A+B|(A+B)^{-T} = |A+B|(A+B)^{-1}\\
&\quad \textit{for symmetric } A, B.
\end{aligned}
$$

Lemma 6.2.5. *The partial derivative of matrix inverses with added B is*

$$
\begin{aligned}
0 &= \partial I\\
&= \partial((A+B)^{-1}(A+B))\\
&= \partial(A+B)^{-1}(A+B) + (A+B)^{-1}\partial(A+B),\\
\partial(A+B)^{-1} &= -(A+B)^{-1}\partial(A+B)(A+B)^{-1},
\end{aligned}
$$

$$
\begin{aligned}
&\frac{\partial c^T(A+B)^{-1}c}{\partial a_{ij}} = c^T\frac{\partial(A+B)^{-1}}{a_{ij}}c\\
&= -c^T(A+B)^{-1}\frac{\partial(A+B)}{\partial a_{ij}}(A+B)^{-1}c\\
&= -c^T(A+B)^{-1}e_ie_j^T(A+B)^{-1}c\\
&= -(c^T(A+B)^{-1}e_i)(e_j^T(A+B)^{-1}c)\\
&= -(c^T(A+B)^{-1}e_i)^T(e_j^T(A+B)^{-1}c)^T\\
&= -e_i^T((A+B)^{-T}cc^T(A+B)^{-T})e_j,
\end{aligned}
$$

$$\frac{\partial c^T(A+B)^{-1}c}{\partial A} = -(A+B)^{-T}cc^T(A+B)^{-T}$$
$$= -(A+B)^{-1}cc^T(A+B)^{-1} \text{ for symmetric } A,\ B.$$

6.3. Partial Derivatives of $F(\theta)$

Lemma 6.3.1. *The partial derivative of the function $\eta(\mu_A, \mu_B, \Sigma_A + \Sigma_B)$ with respect to μ_A is*

$$\frac{\partial \eta(\mu_A, \mu_B, \Sigma_A + \Sigma_B)}{\partial \mu_A} = \eta(\mu_A, \mu_B, \Sigma_A + \Sigma_B)$$
$$\frac{\partial \frac{-1}{2}(\mu_A - \mu_B)^T(\Sigma_A + \Sigma_B)^{-1}(\mu_A - \mu_B)}{\partial \mu_A}$$
$$= -\eta(\mu_A, \mu_B, \Sigma_A + \Sigma_B)(\mu_A - \mu_B)^T(\Sigma_A + \Sigma_B)^{-1}$$

by Lemma 6.2.3.

Lemma 6.3.2. *The partial derivative of the function $\eta(\mu_A, \mu_B, \Sigma_A + \Sigma_B)$ with respect to Σ_A is*

Let $g(\Sigma_A) = ((2\pi)^d|\Sigma_A + \Sigma_B|)^{-\frac{1}{2}}$,

Let $h(\Sigma_A) = e^{-\frac{1}{2}(\mu_A - \mu_B)^T(\Sigma_A + \Sigma_B)^{-1}(\mu_A - \mu_B)}$,

$$\frac{\partial g(\Sigma_A)}{\partial \Sigma_A} = -\frac{1}{2}(2\pi)^{\frac{-d}{2}}|\Sigma_A + \Sigma_B|^{-\frac{3}{2}}\frac{\partial|\Sigma_A + \Sigma_B|}{\partial \Sigma_A}$$
$$= -\frac{1}{2}g(\Sigma_A)(\Sigma_A + \Sigma_B)^{-1}$$

by Lemma 6.2.4,

$$\frac{\partial h(\Sigma_A)}{\partial \Sigma_A} = \frac{-h(\Sigma_A)\partial(\mu_A - \mu_B)^T(\Sigma_A + \Sigma_B)^{-1}(\mu_A - \mu_B)}{2\partial \Sigma_A}$$
$$= \frac{1}{2}h(\Sigma_A)(\Sigma_A + \Sigma_B)^{-1}$$
$$(\mu_A - \mu_B)(\mu_A - \mu_B)^T(\Sigma_A + \Sigma_B)^{-1}$$

by Lemma 6.2.5,

$$\frac{\partial \eta(\mu_A, \mu_B, \Sigma_A + \Sigma_B)}{\partial \Sigma_A} = \frac{\partial g(\Sigma_A)}{\partial \Sigma_A}h(\Sigma_A) + g(\Sigma_A)\frac{\partial h(\Sigma_A)}{\partial \Sigma_A}$$
$$= -\frac{1}{2}\eta(\mu_A, \mu_B, \Sigma_A + \Sigma_B)(\Sigma_A + \Sigma_B)^{-1}$$
$$(I - (\mu_A - \mu_B)(\mu_A - \mu_B)^T(\Sigma_A + \Sigma_B)^{-1}).$$

Lemma 6.3.3. *The partial derivatives of the objective function $F(\theta) = v_f^T Z^f v_f - 2v_f^T Z^{fg} v_g + v_g^T Z^g v_g$ w.r.t. the weight vector v_g is*

$$\frac{\partial F}{\partial v_g} = -2v_f^T Z^{fg} + 2v_g^T Z^g$$
$$= -2(v_f^T Z^{fg} - v_g^T Z^g) \quad \text{by Lemma 6.2.2.}$$

Lemma 6.3.4. *The partial derivative of the function $F(\theta)$ w.r.t. the means μ_g is*

$$\frac{\partial F}{\partial \mu_{l,g}} = -2\sum_i^{k_f}\sum_j^{k_g}\pi_{i,f}\pi_{j,g}\frac{\partial \eta(\mu_{i,f}, \mu_{j,g}, \Sigma_{i,f} + \Sigma_{j,g})}{\partial \mu_{l,g}}$$
$$+ \sum_i^{k_g}\sum_j^{k_g}\pi_{i,f}\pi_{j,g}\frac{\partial \eta(\mu_{i,g}, \mu_{j,g}, \Sigma_{i,g} + \Sigma_{j,g})}{\partial \mu_{l,g}}$$
$$= 2\pi_{l,g}\Bigg(\sum_i^{k_f}\pi_{i,f}\eta(\mu_{i,f}, \mu_{l,g}, \Sigma_{i,f} + \Sigma_{l,g})$$
$$(\mu_{l,g} - \mu_{i,f})^T(\Sigma_{i,f} + \Sigma_{l,g})^{-1}$$
$$- \sum_j^{k_g}\pi_{j,g}\eta(\mu_{l,g}, \mu_{j,g}, \Sigma_{l,g} + \Sigma_{j,g})$$
$$(\mu_{l,g} - \mu_{j,g})^T(\Sigma_{l,g} + \Sigma_{j,g})^{-1}\Bigg) \quad \text{by Lemma 6.3.1.}$$

Lemma 6.3.5. *The partial derivative of the function $F(\theta)$ w.r.t. the covariances Σ_g is*

$$\frac{\partial F}{\partial \Sigma_{l,g}} = -2\sum_i^{k_f}\sum_j^{k_g}\pi_{i,f}\pi_{j,g}\frac{\partial \eta(\mu_{i,f}, \mu_{j,g}, \Sigma_{i,f} + \Sigma_{j,g})}{\partial \Sigma_{l,g}}$$
$$+ \sum_i^{k_g}\sum_j^{k_g}\pi_{i,f}\pi_{j,g}\frac{\partial \eta(\mu_{i,g}, \mu_{j,g}, \Sigma_{i,g} + \Sigma_{j,g})}{\partial \Sigma_{l,g}}$$
$$= \pi_{l,g}\Bigg(\sum_i^{k_f}\pi_{i,f}\eta(\mu_{i,f}, \mu_{l,g}, \Sigma_{i,f} + \Sigma_{l,g})(\Sigma_{i,f} + \Sigma_{l,g})^{-1}$$
$$(I - (\mu_{i,f} - \mu_{l,g})(\mu_{i,f} - \mu_{l,g})^T(\Sigma_{i,f} + \Sigma_{l,g})^{-1})$$
$$- \sum_{j \neq l}^{k_g}\pi_{j,g}\eta(\mu_{l,g}, \mu_{j,g}, \Sigma_{l,g} + \Sigma_{j,g})(\Sigma_{l,g} + \Sigma_{j,g})^{-1}$$
$$(I - (\mu_{l,g} - \mu_{j,g})(\mu_{l,g} - \mu_{j,g})^T(\Sigma_{l,g} + \Sigma_{j,g})^{-1})\Bigg)$$
$$- \frac{\pi_{l,g}^2\Sigma_{l,g}^{-1}}{2\sqrt{(2\pi)^d|2\Sigma_{l,g}|}} \quad \text{by Lemma 6.3.2.}$$