Manifold Optimization for Gaussian Mixture Models

Reshad Hosseini
School of ECE, College of Engineering
University of Tehran, Tehran, Iran
reshad.hosseini@ut.ac.ir

Suvrit Sra
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA.
suvrit@mit.edu

Abstract

We take a new look at parameter estimation for Gaussian Mixture Models (GMMs). Specifically, we propose *Riemannian manifold optimization* as a powerful counterpart to Expectation Maximization (EM). An out-of-the-box invocation of manifold optimization, however, fails spectacularly: it converges vastly slower than EM. Using intuition from manifold convexity, we propose a reformulation that has remarkable empirical consequences. It makes manifold optimization not only match EM (which is a highly encouraging result in itself, given the poor record other nonlinear methods have had against EM) but also outperform it in practical settings, while displaying much less variability in running times.

1 Introduction

Gaussian Mixture Models (GMMs) are key to numerous applications [5, 10, 16, 18]. And for estimating their parameters, Expectation Maximization (EM) [9] remains the *de facto* approach. Although other numerical approaches have been considered, usual nonlinear methods such as conjugate gradients, quasi-Newton, Newton, are typically seen to be much inferior to EM [31].

The main difficulty in applying usual nonlinear optimization techniques to GMMs is the positive definiteness (psd) constraint. This constraint is difficult to handle, especially with increasing data dimensionality. A partial remedy is to use Cholesky decompositions, as was also exploited for semidefinite programming in [7], though at the cost of added nonconvexity. Alternatively, one can resort to interior-point methods [20]. Both approaches turn out to be much inferior to EM.

Considering that the psd constraints make parameter estimation numerically hard, an attractive idea is to use *Riemannian manifold optimization* [1], with the hope that by operating on the manifold of psd matrices, we would implicitly satisfy the psd constraints and thereby have a better focus on likelihood maximization. Unfortunately, this line of thought turns out to be a complete failure! Should we thus discard manifold optimization too? *No.* But we do need to develop a more careful approach, as we now outline.

Intuitively, the mismatch lies in the geometry. Recall that for GMMs the M-step of EM is a Euclidean convex optimization problem (which even has a closed form solution), whereas the log-likelihood is not manifold convex[1] even for a single Gaussian. This suggests that it may be fruitful to consider a reformulation which makes at least the single component gaussian log-likelihood manifold convex. This intuition turns out to have remarkable empirical consequences (Fig. 1), which ultimately enables manifold optimization to compete with EM and often even surpass it.

Contributions. In light of the above background, the main contributions of this work are as follows:

- Introduction of manifold optimization as a powerful numerical tool for GMM parameter estimation. Most importantly, we show a reformulation key to making manifold optimization succeed.
- A solver based on manifold-LBFGS; our contribution here is the design and implementation of a powerful line-search procedure. This line-search helps ensure convergence, and beyond that, it helps LBFGS outperform both EM and the usual manifold conjugate gradient (CG) method; our solver may thus also be of independent interest.

[1] That is, convex along geodesic curves on a manifold.

Related work. The published work on EM is huge, so a summary is impossible and we only mention a few lines of related work. Xu and Jordan [31] examine several aspects of EM for GMMs and counter the claims of Redner and Walker [22], who claimed EM to be inferior to general purpose nonlinear programming. However, it is well-known, see e.g., [22, 31], that EM can attain good likelihood values rapidly, and it scales to larger problems than amenable to usual second-order methods. Local convergence analysis of EM is available in [31], with more refined and precise results in [15]. Our paper develops manifold LBFGS, which can also display local superlinear convergence.

The idea of manifold optimization is new for GMM, but in itself it is a well-developed branch of nonlinear optimization. A classic reference is [27]; a more recent work is [1]; and even a MATLAB toolbox exists now [6]. In machine learning, manifold optimization has witnessed increasing interest[2], e.g., for low-rank optimization [13, 28], or optimization based on geodesic convexity [25, 30].

2 Background and problem setup

We begin with the *Gaussian Mixture Model (GMM)* for vectors $\boldsymbol{x} \in \mathbb{R}^d$, which assigns the density

$$p(\boldsymbol{x}) := \sum\nolimits_{j=1}^{K} \alpha_j p_{\mathcal{N}}(\boldsymbol{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j),$$

in which $p_{\mathcal{N}}$ is a Gaussian with mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance $\boldsymbol{\Sigma} \succ 0$. Given i.i.d. samples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, we estimate $\{\hat{\boldsymbol{\mu}}_j \in \mathbb{R}^d, \hat{\boldsymbol{\Sigma}}_j \succ 0\}_{j=1}^K$ and $\hat{\boldsymbol{\alpha}} \in \Delta_K$, the K-dimensional probability simplex, via maximum-likelihood. This requires solving the ptimization problem:

$$\max_{\boldsymbol{\alpha} \in \Delta_K, \{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j \succ 0\}_{j=1}^K} \quad \sum\nolimits_{i=1}^{n} \log\Big(\sum\nolimits_{j=1}^{K} \alpha_j p_{\mathcal{N}}(\boldsymbol{x}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)\Big). \tag{2.1}$$

Problem (2.1) in general is hard [17].[3] But our focus is more pragmatic: similar to EM, we also seek to efficiently compute local solutions; we approach (2.1) via manifold optimization.

2.1 Manifolds and geodesic convexity

A smooth manifold is a non-Euclidean space that locally resembles Euclidean space [14]. For optimization, it is more convenient to consider Riemannian manifolds (smooth manifolds equipped with an inner product on the tangent space at each point) [1, 27]. Algorithms on manifolds often rely *geodesics*, i.e., curves that (locally) join points along shortest paths. Geodesics help generalize Euclidean convexity to *geodesic convexity*. In particular, say $\mathcal{M}$ is a Riemmanian manifold, and $x, y \in \mathcal{M}$; also let

$$\gamma_{xy} : [0,1] \to \mathcal{M}, \quad \gamma_{xy}(0) = x, \ \gamma_{xy}(1) = y,$$

be a geodesic joining x to y. Then, a set $\mathcal{A} \subseteq \mathcal{M}$ is *geodesically convex* if for all $x, y \in \mathcal{A}$ there is a geodesic γ_{xy} contained within $\mathcal{A}$. Further, a function $f : \mathcal{A} \to \mathbb{R}$ is geodesically convex if for all $x, y \in \mathcal{A}$, the composition $f \circ \gamma_{xy} : [0,1] \to \mathbb{R}$ is convex in the usual sense.

The manifold of interest to us in this paper is $\mathbb{P}^d$, the manifold of $d \times d$ symmetric positive definite matrices. On $\mathbb{P}^d$ the geodesic is given by $\gamma_{\boldsymbol{\Sigma}_1,\boldsymbol{\Sigma}_2}(t) := \boldsymbol{\Sigma}_1^{1/2}(\boldsymbol{\Sigma}_1^{-1/2}\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_1^{-1/2})^t\boldsymbol{\Sigma}_1^{1/2}$ for $0 \le t \le 1$. A function $f : \mathbb{P}^d \to \mathbb{R}$ if geodesically convex on $\mathbb{P}^d$ if it satisfies

$$f(\gamma_{\boldsymbol{\Sigma}_1,\boldsymbol{\Sigma}_2}(t)) \le (1-t)f(\boldsymbol{\Sigma}_1) + tf(\boldsymbol{\Sigma}_2), \qquad t \in [0,1], \ \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2 \in \mathcal{A}.$$

Such functions can be nonconvex in the Euclidean sense, but remain globally optimizable due to geodesic convexity. This property has been important in some matrix theoretic applications [4, 26], and has gained more extensive coverage in several recent works [23, 25, 30].

2.2 Problem reformulation

We begin with parameter estimation for a single Gaussian: although this has a closed-form solution (which ultimately benefits EM), it requires more subtle handling when applying manifold optimization. Maximum likelihood parameter estimation for a single Gaussian

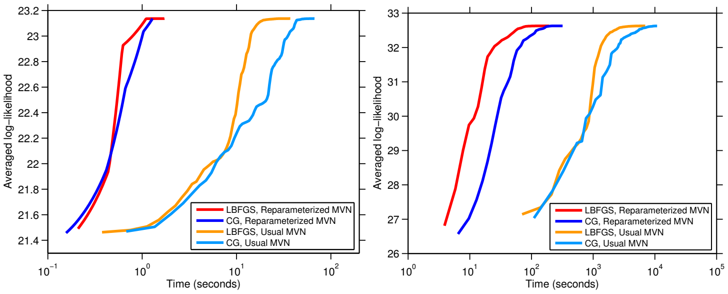
$$\max_{\boldsymbol{\mu}, \boldsymbol{\Sigma} \succ 0} \mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) := \sum\nolimits_{i=1}^{n} \log p_{\mathcal{N}}(\boldsymbol{x}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{2.2}$$

Although (2.2) is convex in the Euclidean sense, but it is *not* geodesically convex. To fix this mismatch, we invoke a simple reformulation[4] that has far-reaching impact. We augment the vectors

[2] Manifold optimization should not be confused with "manifold learning" a separate problem altogether.

[3] Though recent work shows that under strong assumptions, it has polynomial smoothed complexity [11].

[4] This reparamerization in itself is probably folklore; its role in GMM optimization is what is crucial here.



(a) Single Gaussian (b) Mixture of seven Gaussians

Figure 1: The effect of reformulation on convergence speed ($d = 35$); notice x-axis is on a logarithmic scale.

$\boldsymbol{x}_i$ by an extra dimension, and consider $\boldsymbol{y}_i^T = [\boldsymbol{x}_i^T \; 1]$; therewith, we transform (2.2) into the problem

$$\max_{\boldsymbol{S} \succ 0} \; \widehat{\mathcal{L}}(\boldsymbol{S}) := \sum\nolimits_{i=1}^{n} \log q_{\mathcal{N}}(\boldsymbol{y}_i; \boldsymbol{S}), \tag{2.3}$$

where we define $q_{\mathcal{N}}(\boldsymbol{y}_i; \boldsymbol{S}) := 2\pi \exp(\frac{1}{2}) p_{\mathcal{N}}(\boldsymbol{y}_i; \boldsymbol{S})$. Prop. 1 proves the key property of (2.3).

Proposition 1. *Let $\phi(\boldsymbol{S}) \equiv -\widehat{\mathcal{L}}(\boldsymbol{S})$, where $\widehat{\mathcal{L}}(\boldsymbol{S})$ is as in (2.3). Then, ϕ is geodesically convex.*

Theorem 2.1 shows that solving the reformulation (2.3) also solves the original problem (2.2).

Theorem 2.1. *If $\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*$ maximize (2.2), and if $\boldsymbol{S}^*$ maximizes (2.3), then $\widehat{\mathcal{L}}(\boldsymbol{S}^*) = \mathcal{L}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$ for*

$$\boldsymbol{S}^* = \begin{pmatrix} \boldsymbol{\Sigma}^* + \boldsymbol{\mu}^* \boldsymbol{\mu}^{*T} & \boldsymbol{\mu}^* \\ \boldsymbol{\mu}^{*T} & 1 \end{pmatrix}.$$

Thm. 2.1 shows that the reformulation is "faithful" as it leaves the optimum unchanged. Figure 1 shows the true impact of this reformulation. Thm. 2.2 states the mixture model version of Thm. 2.1.

Theorem 2.2. *A local maximum of the reparameterized GMM log-likelihood is a local minimum of the original GMM log-likelihood.*

Finally, we also replace the constraint $\boldsymbol{\alpha} \in \Delta_K$ to make the problem unconstrained. We do this via a commonly used change of variables [12]: $\eta_k = \log(\frac{\alpha_k}{\alpha_K})$, $k = 1, \ldots, K-1$. Assume $\eta_K = 0$ to be a constant, then the final optimization problem is given by:

$$\max_{\{\boldsymbol{S}_j \succ 0\}_{j=1}^K, \{\eta_j\}_{j=1}^{K-1}} \widehat{\mathcal{L}}(\{\boldsymbol{S}_j\}_{j=1}^K, \{\eta_j\}_{j=1}^{K-1}) := \sum_{i=1}^{n} \log\Big(\sum_{j=1}^{K} \frac{\exp(\eta_j)}{\sum_{k=1}^K \exp(\eta_k)} q_{\mathcal{N}}(\boldsymbol{y}_i; \boldsymbol{S}_j)\Big) \tag{2.4}$$

We view (2.4) as a manifold optimization problem; specifically, it is an optimization problem on the product manifold $\big(\prod_{j=1}^K \mathbb{P}^d\big) \times \mathbb{R}^{K-1}$. Let us see how to solve it.

2.3 Manifold Optimization

Successful large-scale (Euclidean) optimization methods such as conjugate-gradient and LBFGS, combine gradients at the current point with gradients and descent directions from previous points to generate a descent direction at the current point. To adapt such algorithms to manifolds, in addition to defining gradients on manifolds, we also need to define how to transport vectors in a tangent space at one point, to vectors in a different tangent space at another point. We refer the reader to [1, 27] for an in depth introduction to manifold optimization.

Different variants of LBFGS can be defined depending where to perform vector transport. We found that the version developed in [26] gives the best performance. We implemented this algorithm together with a crucial Wolfe line-search algorithm; we omit details due to lack of space.

3 Experimental Results

We report performance on both real and simulated data. We initialized mixture parameters using k-means++ [2], and started all methods using the same initialization. The termination criteria are also the same for all methods: they stop when the difference of average log-likelihood falls below 10^{-6}, or when the number of iterations exceeds 1500.

Algorithm 1: Sketch of optimization algorithms (CG, LBFGS) on manifold

```
Given: Riemannian manifold $\mathcal{M}$ with Riemannian metric $g$; parallel transport $\mathcal{T}$ on $\mathcal{M}$; exponential map $R$; initial value $X_0$; a smooth function $f$
for $k = 0, 1, \dots$ do
    Obtain a descent direction based on stored information and grad$f(X_k)$ using defined $g$ and $\mathcal{T}$
    Use line-search to find $\alpha$ such that it satisfies appropraite conditions
    Calculate $X_{k+1} = R_{X_k}(\alpha\xi_k)$
    Based on the memory and need of algorithm store $X_k$, grad$f(X_k)$ and $\alpha\xi_k$
end for
return $X_k$
```

Simulated Data. EM's performance is well-known to depend on the degree of separation of the mixture components [15, 31]. To assess the impact of this separation on our methods, we generate data as proposed in [8, 29]. The distributions are sampled so their means satisfy:

$$\forall_{i \neq j} : \|\boldsymbol{m}_i - \boldsymbol{m}_j\| \geq c \max_{i,j}\{\mathrm{tr}(\boldsymbol{\Sigma}_i) - \mathrm{tr}(\boldsymbol{\Sigma}_j)\},$$

where c models the degree of separation. We apply different algorithms for the case where there is no *eccentricity* (condition number of the covariance matrix); the results are shown in Table 1. For the low separation case $c = 0.2$, the problem becomes ill-conditioned. As predicted by theory, EM converges very slowly in this case; Table 1 confirms this claim. The performance of powerful optimization approaches like CG and LBFGS also degrades [21]. But both CG and LBFGS suffer lesser than EM, and LBFGS fares noticeably better than CG.

Real Data. GMMs have been reported to be a good fit for some natural images [32]. We extracted 200,000 image patches of size 6 × 6 from images and subtracted the DC component, leaving us with 35-dimensional vectors. Performance of different algorithms are reported in Table 2. As for simulated results, performance of EM and manifold CG on the reparametrized parameter space is similar. Manifold LBFGS converges notably faster (except for $K = 6$) than both EM and CG. Without our reformulation, performance of the manifold methods degrades substantially; because the experiments take too long to run, we report only the degraded behavior of CG, which runs about 20 times slower than reparametrized CG and LBFGS.

		EM Algorithm		LBFGS Reparametrized		CG Reparametrized	
		Time (s)	ALL	Time (s)	ALL	Time (s)	ALL
$c = 0.2$	$K = 2$	72.9 ± 37.7	17.6	40.6 ± 21.6	17.6	49.4 ± 31.7	17.6
	$K = 5$	396.7 ± 136.6	17.5	156.1 ± 80.2	17.5	216.3 ± 51.4	17.5
$c = 1$	$K = 2$	7.0 ± 8.4	17.1	13.9 ± 13.7	17.0	16.7 ± 18.7	17.0
	$K = 5$	38.6 ± 67.0	16.2	43.8 ± 38.5	16.2	58.4 ± 47.4	16.2
$c = 5$	$K = 2$	0.2 ± 0.1	17.1	3.0 ± 0.5	17.1	2.7 ± 0.8	17.1
	$K = 5$	26.4 ± 55.3	16.1	20.2 ± 18.4	16.1	23.3 ± 27.8	16.1

Table 1: Speed and ALL comparisons for $d = 20$, $e = 1$.

	EM Algorithm		LBFGS Reparametrized		CG Reparametrized		CG Usual	
	Time (s)	ALL	Time (s)	ALL	Time (s)	ALL	Time (s)	ALL
$K = 2$	16.61	29.28	14.23	29.28	17.52	29.28	947.35	29.28
$K = 3$	90.54	30.95	38.29	30.95	54.37	30.95	3051.89	30.95
$K = 4$	165.77	31.65	106.53	31.65	153.94	31.65	6380.01	31.64
$K = 5$	202.36	32.07	117.14	32.07	140.21	32.07	5262.27	32.07
$K = 6$	228.80	32.36	245.74	32.35	281.32	32.35	10566.76	32.33
$K = 7$	365.28	32.63	192.44	32.63	318.95	32.63	10844.52	32.63
$K = 8$	596.01	32.81	332.85	32.81	536.94	32.81	14282.80	32.58
$K = 9$	900.88	32.94	657.24	32.94	1449.52	32.95	15774.88	32.77
$K = 10$	2159.47	33.05	658.34	33.06	1048.00	33.06	17711.87	33.03

Table 2: Speed and ALL comparisons for natural image data $d = 35$.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[2] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 1027–1035, 2007.

[3] S. Balakrishnan, M. J. Wainwright, and B. Yu. Statistical guarantees for the EM algorithm: From population to sample-based analysis. *arXiv preprint arXiv:1408.2156*, 2014.

[4] R. Bhatia. *Positive Definite Matrices*. Princeton University Press, 2007.

[5] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2007.

[6] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 15(1):1455–1459, 2014.

[7] S. Burer, R. D. Monteiro, and Y. Zhang. Solving semidefinite programs via nonlinear programming. part i: Transformations and derivatives. Technical Report TR99-17, Department of Computational and Applied Mathematics, Rice University, Houston TX, 1999.

[8] S. Dasgupta. Learning mixtures of gaussians. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 634–644. IEEE, 1999.

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

[10] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2nd edition, 2000.

[11] R. Ge, Q. Huang, and S. M. Kakade. Learning Mixtures of Gaussians in High Dimensions. *arXiv:1503.00424*, 2015.

[12] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

[13] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010.

[14] J. M. Lee. *Introduction to Smooth Manifolds*. Number 218 in GTM. Springer, 2012.

[15] J. Ma, L. Xu, and M. I. Jordan. Asymptotic convergence rate of the em algorithm for gaussian mixtures. *Neural Computation*, 12(12):2881–2907, 2000.

[16] G. J. McLachlan and D. Peel. *Finite mixture models*. John Wiley and Sons, New Jersey, 2000.

[17] A. Moitra and G. Valiant. Settling the polynomial learnability of mixtures of gaussians. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 93–102. IEEE, 2010.

[18] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[19] I. Naim and D. Gildea. Convergence of the EM algorithm for gaussian mixtures with unbalanced mixing coefficients. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1655–1662, 2012.

[20] Y. Nesterov and A. S. Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[21] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2006.

[22] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood, and the EM algorithm. *Siam Review*, 26:195–239, 1984.

[23] W. Ring and B. Wirth. Optimization methods on riemannian manifolds and their application to shape space. *SIAM Journal on Optimization*, 22(2):596–627, 2012.

[24] R. Salakhutdinov, S. T. Roweis, and Z. Ghahramani. Optimization with EM and Expectation-Conjugate-Gradient. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 672–679, 2003.

[25] S. Sra and R. Hosseini. Geometric optimisation on positive definite matrices for elliptically contoured distributions. In *Advances in Neural Information Processing Systems*, pages 2562–2570, 2013.

[26] S. Sra and R. Hosseini. Conic Geometric Optimization on the Manifold of Positive Definite Matrices. *SIAM Journal on Optimization*, 25(1):713–739, 2015.

[27] C. Udrişte. *Convex functions and optimization methods on Riemannian manifolds*. Kluwer Academic, 1994.

[28] B. Vandereycken. Low-rank matrix completion by riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013.

[29] J. J. Verbeek, N. Vlassis, and B. Kröse. Efficient greedy learning of gaussian mixture models. *Neural computation*, 15(2):469–485, 2003.

[30] A. Wiesel. Geodesic convexity and covariance estimation. *IEEE Transactions on Signal Processing*, 60 (12):6182–89, 2012.

[31] L. Xu and M. I. Jordan. On convergence properties of the EM algorithm for Gaussian mixtures. *Neural Computation*, 8:129–151, 1996.

[32] D. Zoran and Y. Weiss. Natural images, gaussian mixtures and dead leaves. In *Advances in Neural Information Processing Systems*, pages 1736–1744, 2012.