

# Abstract

Gaussian Mixture Models (GMM), commonly used in pattern recognition and machine learning, providing a flexible probabilistic model for the data. Conventional algorithm for maximum likelihood estimation of parameters is called Expectation-Minimization (EM) provides fixed point iteration procedure, not global optimal in general case and strongly depends on initialization. Stochastic search algorithms and in particular particle swarm optimization (PSO) are quite popular alternative for global optimization and have been already used for optimizing GMM. Our contribution is that we suggest new covariance matrix parametrization that allows us to successfully apply PSO for optimizing parameters of GMM.

## Intoduction

### GMM parameters estimation

Provide statement for GMM: We consider a family of mixture of  $K$  multivariate Gaussian distributions in  $\mathbb{R}^d$  that are parametrized as follows:  $\theta = \{w_1, w_2, \dots, w_K, \mu_1, \Sigma_1, \dots, \mu_K, \Sigma_K\}$ . Where  $\{\mu_i, \Sigma_i\}$  is parametrization of  $i$ th multivariate Gaussian distribution,  $\mu_k \in \mathbb{R}^d$  is vector of means and covariance matrix  $\Sigma_k \in \mathbb{S}_{++}^d$  where  $\mathbb{S}_{++}^d$  is a set of symmetric positive definite  $d \times d$  matrices and  $w_k \in [0, 1]$  is a weight of each gaussian distribution and  $\sum_{k=1}^K w_k = 1$ . Given a set of data  $X = \{x_1, \dots, x_n\}, x_i \in \mathbb{R}^d$  which are i.i.d. Mixture probability density function is  $p(x|\theta) = \sum_{k=1}^K w_k p_k(x|\mu_k, \Sigma_k)$ . Our objective is to obtain maximum likelihood estimation of  $\theta$  through maximizing log-likelihood function:

$$\log(X|\theta) = \sum_{i=1}^N \log\left(\sum_{k=1}^K w_k p_k(x_i|\mu_k, \Sigma_k)\right)$$

Since this functional is not convex we don't have convenient algorithm for obtaining global maxima.

### Expectation Maximization

Conventional algorithm for maximizing log-likelihood is Expectation-Minimization(EM). We do add latent variables to the model  $Z = \{z_1, \dots, z_n\}, z_i \in \mathbb{Z}[0, k], z_i = k$  if  $x_i$  was generated from  $k$ th Gaussian component. Then we do add probability  $z_{ik} = q(z_i = k) = p(z_i = k|x_i, \mu_k, \Sigma_k)$  that  $x_i$  was generated from  $k$ th Gaussian component and reformulate our model and log-likelihood function considering that variables.

$$\log p(X|\theta) = \mathbb{E}_{q(z)}\left[\log \frac{p(X, Z|\theta)}{q(z)}\right] + KL(q||p)$$

Since  $KL(q||p) \geq 0$  we get that  $\log p(X|\theta) \geq \mathbb{E}_{q(z)}\left[\log \frac{p(X, Z|\theta)}{q(z)}\right]$ .  $\mathbb{E}_{q(z)}\left[\log \frac{p(X, Z|\theta)}{q(z)}\right] = \mathcal{L}(q, \theta)$  is called Evidence Lower Bound (ELBO) and we do optimize it instead of  $\log p(X|\theta)$ . Then consider fixed point iteration:

For fixed  $\theta^{(t)}$ , where  $t$  is the number of iteration:

$$q(Z)^{(t+1)} = \arg\max_q \mathcal{L}(q, \theta^{(t)}) = p(Z|X, \theta^{(t)})$$

$$z_{ik}^{(t+1)} = q(z_i = k)^{(t+1)} = \frac{w_k^{(t)} p_k(x_i, \theta_k^{(t)})}{\sum_{j=1}^K w_j^{(t)} p_j(x_i, \theta_j^{(t)})}$$

Then for fixed  $q(Z)^{(t+1)}$

$$\theta^{(t+1)} = \arg\max_{\theta} \mathcal{L}(q^{(t+1)}, \theta) = \arg\max_{\theta} \mathbb{E}_{q(z)} \log p(X, Z|\theta)$$

That leads up to update of parameters:

$$w_k^{(t+1)} = \frac{1}{N} \sum_{i=1}^N z_{ik}^{(t+1)}$$

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^N z_{ik}^{(t+1)} x_i}{\sum_{i=1}^N z_{ik}^{(t+1)}}$$

$$\Sigma_k^{(t+1)} = \frac{\sum_{i=1}^N z_{ik} (x_i - \mu_k^{(t+1)})(x_i - \mu_k^{(t+1)})^T}{\sum_{i=1}^N z_{ik}}$$

After a number of iteration EM converge and one the EM problems is that it strongly depend on initialization.

## Particle swarm optimization

Particle swarm optimization (PSO) is a population-based stochastic search algorithm that is inspired by the social interactions of swarm animals. In PSO each member of a polutaion is called particle. We have some function we want to optimize  $f(X_\theta)$ ,  $X_\theta \in \mathbb{R}^d$ . Each particle is represented as two vectors:  $X_i = \{X_{(\theta,i)}, x_v\}$ ,  $X_\theta, x_v \in \mathbb{R}^d$  parameters and velocity respectively. Parameters part is the candidate solution that particle does represent and velocity is the part used for defining optimization step. We can evaluate criterion value for each particle  $f(X_{(\theta,i)})$ . As the optimization prcedure we do iterative updates. At the each iteration we do remember the particle position that does have the best criterion among all the particle positions among all the iterations (global best) and is denoted as  $X^{(GB,t)}$ . Also through the search each particle remebers its personal best position  $f(X_i^{(PB,t)}) \geq f(X_{\theta,i})$ ,  $i \in 1, \dots, i_{last}$

We start from initializing particles by some way, choosing  $X^{(GB,1)}$  and  $X_i^{(PB,1)}$  and then we do run optimization steps as follows:

$$X_{v,i}^{(t+1)} = \eta X_{v,i}^{(t)} + c_1 U_1^{(t)} (X_i^{(PB,t)} - X_{(\theta,i)}) + c_2 U_2^{(t)} (X^{(GB,t)} - X_{(\theta,i)})$$

$$X_{(\theta,i)}^{(t+1)} = X_{(\theta,i)}^{(t)} + X_{v,i}^{(t+1)}$$

$$X_i^{(PB,t+1)} = X_i^{(PB,t^*)}, t^* = \operatorname{argmax}_t f(X_i^{(PB,t)})$$

$$X^{(GB,t+1)} = X_{\theta,i}^{(GB,t^*)}, t^* = \operatorname{argmax}_t f(X_i^{(GB,t)})$$

Where  $\eta$  is inertia coefficient,  $c_1$  and  $c_2$  are acceleration weights,  $U_1, U_2 \sim U[0, 1]$  random numbers and  $t$  is iteration number. We have some function we want to optimize.

## PSO for GMM

We do want apply PSO for optimizing GMM. Every particle parameters part will be representing full GMM candidate solution  $X_{(\theta,i)} = \{w_1, w_2, \dots, w_K, \mu_1, \Sigma_1, \dots, \mu_K, \Sigma_K\}_i$ . But we would need to maintain positive definite constraint on covariance matrix  $\Sigma \in \mathbb{S}_{++}^d$  and PSO update doesn't necessary do it.  $\Sigma_{new} = \Sigma_1 + (\Sigma_2 - \Sigma_3)$  may not be positive definite. Then to perform optimization on the manifold of positive definite matrices lets consider some parametrization of covariance matrix  $\Sigma = g(\vartheta)$  so that changing  $g(\vartheta) \in \mathbb{S}_{++}^d, \forall \vartheta \in \mathbb{R}^m$ . One possible parametrization is Cholesky decomposition  $\Sigma = LL^T$  but as a result we have  $\frac{d(d+1)}{2}$  parameters which are unbounded, can have very diverse values  $(-\infty, +\infty)$  and that doesn't allow us to properly optimize. Another parametrization through eigenvalues and givens rotation matrix angles was presented in [1].

### Eigenvalue and Givens rotation matrix angles parametrization

**Theorem 1** *An arbitrary covariance matrix with  $\frac{d(d+1)}{2}$  degrees of freedom can be parametrized by using  $d$  eigenvalues in a particular order and  $\frac{d(d-1)}{2}$  Givens rotation matrix angles  $\phi^{pq} \in [-\pi/4, 3\pi/4]$  for  $1 \leq p < q \leq d$  computed from the eigenvector matrix whose columns store the eigenvectors in the same order as the corresponding eigenvalues.*

**Lemma 1** *An eigenvector matrix  $V \in \mathbb{R}^{d \times d}$  can be written as a product of  $\frac{d(d-1)}{2}$  Givens rotation matrices with angles  $\phi^{pq} \in [-\pi/4, 3\pi/4]$  and a diagonal matrix with  $\pm 1$  entries.*

For proof see [1].  $\Sigma \in \mathbb{S}_{++}^d$  is symmetric hence it is diagonalizable in orthogonal basis and we can represent it as  $\Sigma = V\Lambda V^T = \sum_{i=1}^d \lambda_i v_i v_i^T$ . Regarding Lemma 1 that we don't need to store the diagonal matrix  $\pm 1$  entries because  $v_i v_i^T = (-v_i)(-v_i)^T$ . Then we can construct our covariance matrix as  $\Sigma = V\Lambda V^T = \sum_{i=1}^d \lambda_i v_i v_i^T = \sum_{i=1}^d \lambda_i \prod_{p=1}^d \prod_{q=p+1}^d G(p, q, \phi^{pq})$ , where  $G(p, q, \phi^{pq})$  is Givens rotation matrix with for dimensions  $p$  and  $q$  with  $\phi^{pq}$  rotation angle. Also notice since our eigendecomposition is unique up to eigenvalues/eigenvectors permutation so we need to maintain order of eigenvalues and corresponding Givens rotation angles after making the decomposition. To tackle this issue authors proposed algorithm of eigenvectors ordering w.r.t. reference eigenvectors matrix  $V_{ref}$  which happens to be the personal best for each particle.

Algorithm with such parametrization stated as follows:

---

#### Algorithm 1

---

**Input:** d-dimensional dataset with N samples, number of iterations  $T_1$ , number of iteration for EM local convergence  $T_2$ , number of components  $K$ , number of particles  $M$ , PSO hyperparameters  $\eta, c_1, c_2$

**Output:** GMM solution  $\{w_1, w_2, \dots, w_K, \mu_1, \Sigma_1, \dots, \mu_K, \Sigma_K\}$

**for**  $t = 1$  to  $T_1$  **do**

**for**  $m = 1$  to  $M$  **do**

        Construct K eigenvalue matrices

        Construct K eigenvector matrices by multiplying Givens rotation angles

        Run EM for local convergence for  $T_2$  iterations  $T_2$ : number of EM iterations for each PSO iteration

        Compute K eigenvalue and eigenvector matrices via singular value decomposition of new covariance matrices

        Reorder eigenvalues and eigenvectors of each covariance matrix according to personal best

        Extract Givens rotation angles using QR factorization

        Replace particle's means, eigenvalues, and angles

        Calculate log-likelihood

        Update personal best

**end for**

    Update global best

**for**  $m = 1$  to  $M$  **do**

        Update particle's means, eigenvalues, and angles

**end for**

**end for**

---

But this parametrization still have a problem since it has  $\frac{d(d+1)}{2}$  parameters and we need to compute SVD and QR factorization for each Gaussian covariance matrix for each particle for each iteration.

### Low rank delta parametrization

We propose novel method of parametrization not the covariance matrix but the addition to it.  $\Sigma_{new} = \Sigma + \Delta(\vartheta)$ . Using property that for  $\Sigma_1, \Sigma_2 \in \mathbb{S}_{++}^d$  holds true  $\Sigma_1 + \Sigma_2 \in \mathbb{S}_{++}^d$ . We choose such  $\Delta$  that  $\Delta(\vartheta) \in \mathbb{S}_{++}^d, \forall \vartheta \in \mathbb{R}^m$ . Then we get that that  $\Sigma + \Delta(\vartheta) \in \mathbb{S}_{++}^d$ .

Let's try to avoid  $\frac{d(d+1)}{2}$  parameters for covariance matrix. We can do that by making low rank addition, in other words making  $\Delta$  such that  $rank(\Delta(\vartheta)) < d$  and hence we can lower number of dimensions for  $\vartheta$  so that  $\vartheta \in \mathbb{R}^m, m < \frac{d(d+1)}{2}$ .

We propose low rank delta parametrization satisfying all the above requirements as follows:

$$\Sigma_{new} = \Sigma + \text{diag}(a_1^2, \dots, a_d^2) + \sum_{i=1}^R b_i b_i^T$$

Notice that  $\text{diag}(a_1^2, \dots, a_d^2) \in \mathbb{S}_{++}^d, \forall a_i \in \mathbb{R}$  and  $b_i b_i^T \in \mathbb{S}_{++}^d, \forall b_i \in \mathbb{R}^d$ .  $rank(\sum_{i=1}^R b_i b_i^T) \leq R$  for approximating parameters of covariance matrix and  $\text{diag}(a_1^2, \dots, a_d^2)$  because we need special attention to diagonal elements. Total number of parameters is  $(R+1)d$ . For the initialization we run EM algorithm and get  $\{w_1, w_2, \dots, w_K, \mu_1, \Sigma_1, \dots, \mu_K, \Sigma_K\}$  parameters, we freeze  $\Sigma_{(i,base)}$ , and set  $w_1, w_2, \dots, w_K, \mu_1, \dots, \mu_K$  as particle initial parameters. Then we randomly initialize low rank addition  $a_i \sim N(0, \sigma_1)$  and  $b_i \sim N(0, \sigma_2 I_d)$  and compute covariance matrices for each Gaussian for mth particle as  $\Sigma_{(i,m)} = \Sigma_{(i,base)} + \text{diag}(a_{(1,m)}^2, \dots, a_{(d,m)}^2) + \sum_{i=1}^R b_{(i,m)} b_{(i,m)}^T$ .

Then PSO particle will contain parameters as follows:  $\{w_1, w_2, \dots, w_K, \mu_1, \dots, \mu_K, a_1, \dots, a_d, b_1, \dots, b_r\}$

Then PSO algorithm with such a parametrization will look as follows:

---

**Algorithm 2**

**Input:** d-dimensional dataset with N samples, number of iterations  $T$ , number of initial EM iterations  $T_{EM}$ , number of components  $K$ , number of particles  $M$ , rank of addition  $R$ , PSO hyperparameters  $\eta, c_1, c_2$

**Output:** GMM solution  $\{w_1, w_2, \dots, w_K, \mu_1, \Sigma_1, \dots, \mu_K, \Sigma_K\}$

run EM for  $T_{EM}$  iterations

**for**  $m = 1$  to  $M$  **do**

$a^{(0)} \sim N(0, \sigma_1 I_d)$

▷ Initialize parameters for particles

**for**  $r = 1$  to  $R$  **do**

$b_r^{(0)} \sim N(0, \sigma_2 I_d)$

**end for**

**end for**

**for**  $t = 1$  to  $T$  **do**

**for**  $m = 1$  to  $M$  **do**

Calculate covariance matrices using low rank addition parameters

Calculate log-likelihood

Update personal best

**end for**

Update global best

**for**  $m = 1$  to  $M$  **do**

Update particle's  $\mu_m^{(t)}, w_m^{(t)}, a_m^{(t)}, b_m^{(t)}$

**end for**

**end for**

**return** Global best

---

## Low rank delta parametrization for PSO with EM reinit

EM is still very effective algo to search for a local minima so let's try to utilize it inside our PSO framework. Unfortunately we are not able to extract our parametrization from covariance matrix in a unique way so we won't have an opportunity to init EM using constructed from PSO particles parameters weights, means and covariance matrices, then run EM and construct PSO particle back using resulting EM weights, means and covariances. So let's try to use PSO optimization as a proxy metric. Then we'd have criterion corresponding to each PSO particle parametrization itself,  $\theta_{pso} = \{w_1, w_2, \dots, w_K, \mu_1, \dots, \mu_K, a_1, \dots, a_d, b_1, \dots, b_r\}$ , from  $\theta_{pso}$  we'll construct GMM parameters  $\theta$  in a deterministic way  $\theta_{pso} = \{w_1, w_2, \dots, w_K, \mu_1, \dots, \mu_K, a_1, \dots, a_d, b_1, \dots, b_r\} \rightarrow \{w_1, w_2, \dots, w_K, \mu_1, \Sigma_1, \dots, \mu_K, \Sigma_K\} = \theta$ .  $\log p(x|\theta)$  will be criterion corresponding to each PSO particle parametrization itself. And we are able to launch EM starting from  $\theta$  and obtain new set of GMM parameters  $\theta_{EM}$ .  $EM(init = \theta) = \theta_{EM}$  and corresponding criterion will be  $\log p(x|\theta_{EM})$ . So we'll optimize using PSO w.r.t. first criterion but we'll evaluate second criterion and output as an algorithm result parameters  $\theta_{EM}$ .

Resulting algorithm will be as follows:

1. Additional words about sparsity and non spherical gaussians

## Experiments

### Synthetic Data

Experimenting on rank of addition				
Dataset	Rank	Amplidute	Variation of init	LogLikelihood
Cloud	3	0	0	0
Cloud	5	0	0	0
Cloud	7	0	0	0
Breast Cancer	5	0	0	0
Breast Cancer	10	0	0	0
Breast Cancer	15	0	0	0

---

**Algorithm 3**

**Input:** d-dimensional dataset with N samples, number of iterations  $T$ , number of initial EM iterations  $T_{EM}$ , number of components  $K$ , number of particles  $M$ , rank of addition  $R$ , PSO hyperparameters  $\eta, c_1, c_2$

**Output:** GMM solution  $\{w_1, w_2, \dots, w_K, \mu_1, \Sigma_1, \dots, \mu_K, \Sigma_K\}$

run EM for  $T_{EM}$  iterations

**for**  $m = 1$  to  $M$  **do**

$a^{(0)} \sim N(0, \sigma_1 I_d)$

$\triangleright$  Initialize parameters for particles

**for**  $r = 1$  to  $R$  **do**

$b_r^{(0)} \sim N(0, \sigma_2 I_d)$

**end for**

**end for**

**for**  $t = 1$  to  $T$  **do**

**for**  $m = 1$  to  $M$  **do**

        Calculate covariance matrices using low rank addition parameters

        Calculate log-likelihood

        Update personal best

**end for**

    Update global best

**for**  $m = 1$  to  $M$  **do**

        Update particle's  $\mu_m^{(t)}, w_m^{(t)}, a_m^{(t)}, b_m^{(t)}$

**end for**

**end for**

**for**  $m = 1$  to  $M$  **do**

    Calculate covariance matrices using low rank addition parameters for each particle's personal best

$\theta_{EM} = \text{EM}(\{w_{1,m}^{(PB)}, w_{2,m}^{(PB)}, \dots, w_{K,m}^{(PB)}, \mu_{1,m}^{(PB)}, \Sigma_{1,m}^{(PB)}, \dots, \mu_{K,m}^{(PB)}, \Sigma_{K,m}^{(PB)}\})$

    Calculate log-likelihood ( $\log p(\theta_{EM})$ )

**end for**

Update global best EM reinit

**return** Global best EM reinit

---

Experimenting on variation of initialization				
Dataset	Variation of init	Amplidute	Rank	LogLikelihood
Cloud	0.03	0	0	0
Cloud	0.001	0	0	0
Cloud	0.003	0	0	0
Cloud	0.0001	0	0	0
Breast Cancer	0.03	0	0	0
Breast Cancer	0.001	0	0	0
Breast Cancer	0.003	0	0	0
Breast Cancer	0.0001	0	0	0

Experimenting on Amplitude (learning rate)				
Dataset	Amplidute	Variation of init	Rank	LogLikelihood
Cloud	0.3	0	0	0
Cloud	0.1	0	0	0
Cloud	0.03	0	0	0
Cloud	0.01	0	0	0
Breast Cancer	0.3	0	0	0
Breast Cancer	0.1	0	0	0
Breast Cancer	0.03	0	0	0
Breast Cancer	0.01	0	0	0

**Real World data**

## References

- [1] Çağlar Arı, Selim Aksoy, and Orhan Arıkan. Maximum likelihood estimation of gaussian mixture models using stochastic search. *Pattern Recognition*, 45(7):2804–2816, 2012.