Particle Swarm optimization of EM-algorithm for Gaussian Mixture Model

Sergei Kholkin [1] **Maxim Kurkin** [1] **Anastasia Batsheva** [1] **Anna Akhmatova** [1]

Abstract

GMM is optimized using EM algorithm. There were many attempts to beat EM algorithm (Hosseini & Sra, 2015) it is still mainly SOTA. Through EM algorithm find local minima and highly depends on initial data. The idea of the project is to optimize the initialization of EM algorithm using Particle Sworn Optimization (PSO) as it was already done with similar algorithm k-means (Li & Wang, 2022).

1. 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: $\theta = \{w_1, w_2, \ldots, w_K, \mu_1, \Sigma_1, \ldots, \mu_K, \Sigma_K\}$. Where $\{\mu_i, \Sigma_i\}$ is parametrization of ith 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 $X = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^d$ which are i.i.d. Mixture probability desnity 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 θ 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 covinient algorithm for obtaining global maxima.

2. Expectation Maximization

Conventional algorithm for maximizing log-likelihood is Expectation-Minimization(EM). We do add latent variables to the model $Z = \{z_1, \ldots, z_n\}, z_i \in \mathbb{Z}[0, k]$, $z_i = k$ if x_i was generated from kth 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 kth Gaussian component and reformulate out model and log-likelihood function considering that varibales.

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

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 $\theta^{(t)}$, whete 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:

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

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

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

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

*Equal contribution [1]Skolkovo Institute of Science and Technology, Russia. Correspondence to: Sergei Kholkin <sergei.kholkin@skoltech.ru>, Anastasia Batsheva <anastasia.batsheva@skoltech.ru>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

3. 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 prcodure 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 $f(X_i^{(PB,t)}) \geq f(X_{\theta,i}), i \in 1, \ldots, i_{last}$

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:

$$
\begin{aligned}
X_{v,i}^{(t+1)} = \eta X_{v,i}^{(t)} + c_1 U_1^{(t)} (X_i^{(PB,t)} - X_{(\theta,i)}) + \\
+ c_2 U_2^{(t)} (X^{(GB,t)} - X_{(\theta,i)}), \\
X_{(\theta,i)}^{(t+1)} = X_{(\theta,i)}^{(t)} + X_{v,i}^{(t+1)}, \\
X_i^{(PB,t+1)} = X_i^{(PB,t^*)}, t* = \arg\max_t f(X_i^{(PB,t)}), \\
X^{(GB,t+1)} = X_{\theta,i}^{(GB,t^*)}, t* = \arg\max_t f(X_i^{(GB,t)})
\end{aligned}
$$

Where η 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.

4. PSO for GMM

We do want apply PSO for optimizing GMM. Every particle parameters part will be representing full GMM candidate solution $X_{(\theta,i)} = \{w_1, w_2, \ldots, w_K, \mu_1, \Sigma_1, \ldots, \mu_K, \Sigma_K\}_i$. 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 $\Sigma = LL^\top$ 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 optimze. Another parametrization through eigenvalues and givens rotation matrix angles was presented in (Çağlar Arı et al., 2012).

4.1. Eigenvalue and Givens rotation matrix angles parametrization

Theorem 1 *An arbitary 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 eigenvactor 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 ± 1 entries.*

For proof see (Çağlar Arı et al., 2012). $\Sigma \in \mathbb{S}_{++}^d$ is symmetrid hence it is diagonizable in orthogonal basis and we can represent it as $\Sigma = V \Lambda V^\top = \sum_{i=1}^d \lambda_i v_i v_i^\top$. Regarding Lemma 1 that we don't need to store the diagonal matrix ± 1 entries because $v_i v_i^\top = (-v_i)(-v_i)^\top$. Then we can construct our covariance matrix as $\Sigma = V \Lambda V^\top = \sum_{i=1}^d \lambda_i v_i v_i^\top = \sum_{i=1}^d \lambda_i \prod_{p=1}^d \prod_{q=p+1}^d G(p, q, \phi^{pq})$, where $G(p, q, \phi^{pq})$ is Givens rotaion matrix with for dimensions p and q with ϕ^{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:

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

4.2. 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 Δ that $\Delta(\vartheta) \in \mathbb{S}_{++}^d, \forall \vartheta \in \mathbb{R}^m$. Than 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 Δ such that $\mathrm{rank}(\Delta(\vartheta)) < d$ and hence we can lower number of dimensions for ϑ so that $\vartheta \in \mathbb{R}^m, m < \frac{d(d+1)}{2}$.

We propose low rank delta parametrization satisfying all the

Algorithm 1 Algorithm without reinitialization of EM

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 η, c_1, c_2
Output: GMM solution $\{w_1, w_2, .., w_K, \mu_1, \Sigma_1, .., \mu_K, \Sigma_K\}$

```
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
```

above requirements as follows:

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

Notice that $\mathrm{diag}(a_1^2, \dots, a_d^2) \in \mathbb{S}_{++}^d, \forall a_i \in R$ and $b_i b_i^\top \in \mathbb{S}_{++}^d, \forall b_i \in \mathbb{R}^d$. $\mathrm{rank}(\sum_{i=1}^R b_i b_i^\top) \le R$ for approximating parameters of covariance matrix and $\mathrm{diag}(a_1^2, \dots, a_d^2)$ 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 $\Sigma_{(i,m)} = \Sigma_{(i,base)} + \mathrm{diag}(a_{(1,m)}^2, \dots, a_{(d,m)}^2) + \sum_{i=1}^R b_{(i,m)} b_{(i,m)}^\top$.

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

4.3. 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 will have an opportunity to init EM using constructed from PSO particles parameters weights, means and covariance matrices, then run EM and but we won't have an opportunity to construct PSO particle back using resulting EM weights, means and cavariances. 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 θ_{pso} we'll construct GMM parameters θ 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 θ and obtain new set of GMM parameters θ_{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 θ_{EM}.

Also for economy of computations we can run EM reinitialization for particles each N iterations (for example 10).

Resulting algorithm will be as follows:

5. Experiments

5.1. Experiments setup

We can compare time of algorithms execution 1

Or we can compare number of EM iterations for each algorithm (approach used in (Çağlar Arı et al., 2012)). For PSO algorithm we have M particles which are being init based on one EM which runs T_{init} iterations with M ran-

Algorithm 2 Algorithm with reinitialization of EM

Input: d-dimensional dataset with N samples, number of iterations T, number of initial EM iterations T_{init}, number of components K, times of EM reinit T_1 , number of iteration for EM reinit T_2, number of particles M, rank of addition R, PSO hyperparameters η, c_1, c_2

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

```
run EM for T_init iterations
for m = 1 to M do
    a^(0) ~ N(0, σ_1 I_d)                                      ▷ Initialize parameters for particles
    for r = 1 to R do
        b_r^(0) ~ N(0, σ_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                                          ▷ PSO update
        Update particle's μ_m^(t), w_m^(t), a_m^(t), b_m^(t)
    end for
    if t mod (t div T_1) = 0 then
        for m = 1 to M do
            Calculate covariance matrices using low rank addition parameters for each particle's personal best
            θ_EM = EM({w_{1,m}^(PB), w_{2,m}^(PB), .., w_{K,m}^(PB), μ_{1,m}^(PB), Σ_{1,m}^(PB), .., μ_{K,m}^(PB), Σ_{K,m}^(PB)})   ▷ Run EM for T_2 iterations
            Calculate log-likelihood (log p(θ_EM))
        end for
    end if
end for
Update global best EM reinit
return Global best EM reinit
```

dom init samples, then each particle through running of PSO algorithm is being reinit T_1 times using EM with T_2 iteration budget. So in total number of EM iterations for PSO algorithm run is: $M(T_{init} + T_1 * T_2)$. Then lets use this EM iterations budget for EM with random init and run M random samples with $(T_{init} + T_1 * T_2)$ iterations budget for each EM run. 2

5.2. Synthetic Data

We'll use model of generatin synthtic data from (Çağlar Arı et al., 2012). Data sets will be generated from some Gaussian Mixture Model itself with various dimenstions $d \in \{30, 50, 70\}$, number of components will be $M \in \{10, 15, 20\}$, with sample size $N \in \{500, 1000, 2000\}$. Also we add c separation as in (Dasgupta, 1999). Two Gaussians can be called c-separated if: $\|\mu_2 - \mu_1\|_2 \leq c\sqrt{d \max \lambda_{max}(\Sigma_1), \lambda_{max}(\Sigma_2)}$. We want our Gaussians to be less separated sooptimization problem of learning GMM will be harder, so we want Gaussian not to be c separated. Procedure of generating data sets will be as follows:

5.3. Real world data

We'll be testing on the following real world datasets:

- Cloud Dataset: 10 dimensions, 1024 data points
- Breast Cancer Dataset: 22 dimensions, 569 data points
- Landsat Satelite Dataset: 36 dimensions, 6435 data points

References

Dasgupta, S. Learning mixtures of gaussians. *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, pp. 634–644, 1999.

Hosseini, R. and Sra, S. Matrix manifold optimization for gaussian mixtures. In Cortes, C., Lawrence, N., Lee, D.,

Comparison with random init EM by running time						
Dataset	M	Rank	Ampl	Var of init	LL	Time, sec
Breast Cancer	50	10	0.003	0.003	75.77 +- 0.55	128
Breast Cancer	100	EM	-	-	73.95 +- 0.34	22
Breast Cancer	600	EM	-	-	74.46 +- 0.44	138

Table 1. PSO EM reinit and EM with random init algos comparison

Algorithm 3 Algorithm for generation of synthetic dataset

Input: dimensionality d**, number of components** M**, number of samples** N**,** c **- separation coefficient**
Output: $\{x_1, x_2, .., x_N\}$**, where** $x_i \in \mathbb{R}^d$

$w \sim U[0, 1]^d$
$w := w / \sum_{i=1}^{d} w_i$
$m := 0$
$GMM = \emptyset$
while $m < M$ **do**
  $\mu \sim U[0, 100]^d$
  for $i = 1$ to d **do**
    $\lambda_j \sim U[1, 16]$ ▷ Sample eigenvalues
    for $j = i + 1$ to d **do**
      $\phi_{(p,q)} \sim U[-\pi/4, 3\pi/4]$ ▷ Sample Givens rotation angles
    end for

    $\Sigma = f(\lambda_1, .., \lambda_d, \phi_{(1,1)}, \phi_{(d,d)})$ ▷ Construct covariance matrix from Givens rotation angles and eigenvalues
$\mu_{ref}, \Sigma_{ref} \in$
    if $\|\mu - \mu_{ref}\|_2 \leq c\sqrt{d \max \lambda_{max}(\Sigma), \lambda_{max}(\Sigma_{ref})}$ **then**
      $GMM = GMM \cup \{\mu, \Sigma\}$
      $m = m + 1$
    end if
  end for

  Sample $x_1, .., x_N$ from M Gaussian Mixtures w.r.t. w
  return $x_1, .., x_N$

Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/dbe272bab69f8e13f14b405e038deb64-Paper.pdf.

Li, H. and Wang, J. Collaborative annealing power k-means++ clustering. *Knowledge-Based Systems*, 255:109593, 2022. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2022.109593. URL https://www.sciencedirect.com/science/article/pii/S0950705122008036.

Çağlar Arı, Aksoy, S., and Arıkan, O. Maximum likelihood estimation of gaussian mixture models using stochastic search. *Pattern Recognition*, 45(7):2804–2816, 2012. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2011.12.023. URL https://www.sciencedirect.com/science/article/pii/S0031320312000167.

A. Experiment results

Comparison with random init EM by EM iterations budget						
Dataset	M	T_1	T_2	$T_1 \cdot T_2$	PSO LL	EM LL
Cloud Dataset	50	10	0.003	0.003	75.77 +- 0.55	128
Breast Cancer	100	EM	-	-	73.95 +- 0.34	22
Breast Cancer	600	EM	-	-	74.46 +- 0.44	138
Breast Cancer	600	EM	-	-	74.46 +- 0.44	138
Breast Cancer	600	EM	-	-	74.46 +- 0.44	138

Table 2. M — number of particles and number of random inits for initial PSO EM, T_1 — Number of EM reinits from PSO particle coordinates, T_2 — max number of iterations of EM reinits, EM LogLikelihood — log likelihood of best EM results from $2M$ random inited points and with max number of iterations equal to $T_1 \cdot T_2$, PSO LogLikelihood — log likelihood of EM reinit from PSO particle coordinates. Recap that PSO algorithm in these experiments setup runs as follows:

1. Run EM with M random inits for $T_1 \cdot T_2$ iterations
2. Run PSO for $F \cdot T_1$ iterations, each F PSO iterations run GMM reinit that starts from particle coordinates and runs for T_2 iterations

Total number of EM iterations for this PSO algorithm will be $2 \cdot M \cdot T_1 \cdot T_2$.

Comparison with random init EM by EM iterations budget for synthetic data								
D	N_{comp}	C	M	T_1	T_2	$T_1 \cdot T_2$	PSO LL	EM LL
30	10	2	50	5	100	500	31.72 +- 0.06	31.61 +- 0.09
50	15	$+\infty$	50	5	100	500	58.382 +- 0.49	57.97 +- 0.37
50	20	$+\infty$	50	5	100	500	112.62 +- 0.49	112.09 +- 0.33
70	30	$+\infty$	50	5	100	500	170.40 +- 0.7	169.58 +- 0.28

Table 3. Comparison by iterations on synthetic data. For parameters discription of M, T_1, T_2, $T_1 \cdot T_2$, PSO LL, EM LL check 2. Synthetic dataset parameters: D – number of dimensions, N_{comp} – number of clusters, C – separation coefficient