Learning Gaussian Mixture Models and Tensor Decompositions

Zi Yang
University of California, San Diego

UC San Diego

Introduction

A Gaussian mixture model of r components consists of r Gaussian distributions, $\mathcal{N}(\mu_i, \Sigma_i)$ for $i = 1, \dots, r$, where $\mu_i \in \mathbb{R}^d$ is the expectation (or mean) and $\Sigma_i \in \mathbb{R}^{d\times d}$ is the covariance matrix. Each sample of the distribution is drawn from one Gaussian component and the probability that a sample is drawn from the ith Gaussian distribution is $\omega_i > 0$. The density function of the Gaussian mixture model is the weighted sum of the r density functions of component Gaussian distributions. Learning a Gaussian mixture model is to estimate parameters of the model $\omega_i, \mu_i, \Sigma_i$ from given samples of the model. We proposed an algorithm[1] to learn Gaussian mixture models by using tensor decompositions.

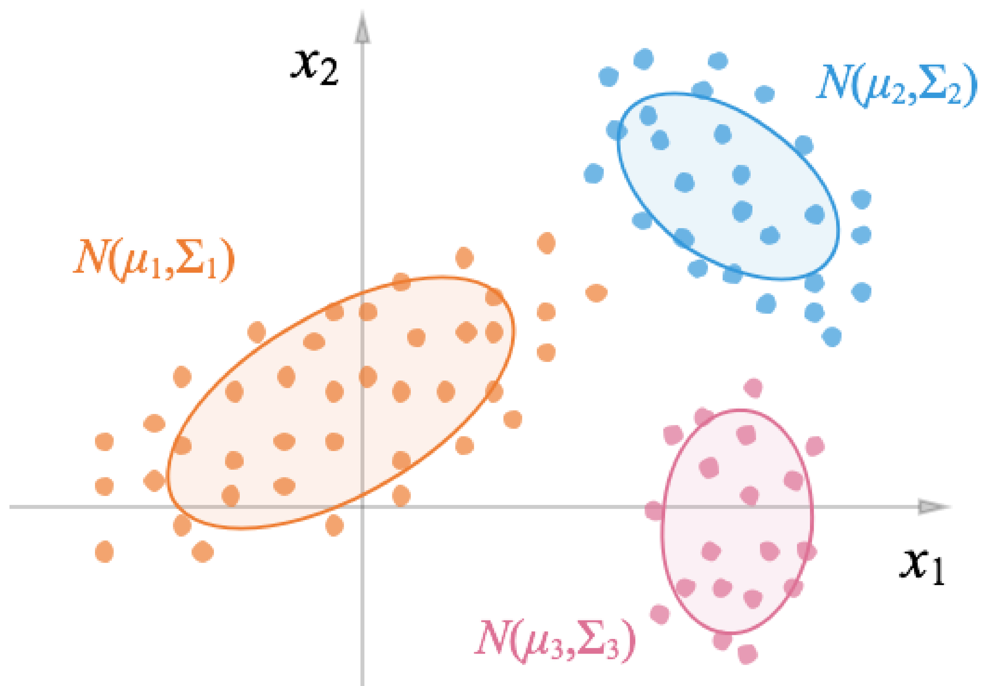

Fig. 1: Gaussian Mixture Model

Gaussian Mixture and Tensor Decompositions

Let x be the random variable of the d-dimensional Gaussian mixture model with r components and parameters $\{(\omega_i, \mu_i, \Sigma_i) : 1 \le i \le r\}$. When covariance matrices $\Sigma_i = \text{diag}(\sigma_{i1}^2, \dots, \sigma_{id}^2)$ are all diagonal matrices, the third order moment tensor $M_3 := \mathbb{E}[x \otimes x \otimes x]$ can be written as

$$M_3 = \sum_{i=1}^{k} \omega_i \mu_i \otimes \mu_i \otimes \mu_i + \sum_{j=1}^{d} \Big(a_j \otimes e_j \otimes e_j + e_j \otimes a_j \otimes e_j + e_j \otimes e_j \otimes a_j \Big), \quad (1)$$

where $a_j := \sum_{i=1}^r \omega_i \sigma_{ij}^2 \mu_i, \quad j = 1, \dots, d$. We are particularly interested in the following cubic symmetric tensor

$$\mathcal{F} := \sum_{i=1}^{r} \omega_i \mu_i \otimes \mu_i \otimes \mu_i. \quad (2)$$

The entries of M_3 and $\mathcal{F}$ are the same, when their labels i_1, i_2, i_3 are distinct from each other, i.e.,

$$(M_3)_{i_1 i_2 i_3} = (\mathcal{F})_{i_1 i_2 i_3} \quad \text{for} \quad (i_1, i_2, i_3) \in \Omega,$$

where $\Omega := \{(i_1, i_2, i_3) | i_1 \ne i_2 \ne i_3 \ne i_1\}$ is the label set. The tensor M_3 can be estimated by samples, so partial entries $(\mathcal{F})_\Omega$ can be obtained from M_3. We aim to find the decomposition of $\mathcal{F}$ from $(\mathcal{F})_\Omega$, i.e. we are looking for vectors $p_1, \dots, p_r$ such that

$$\mathcal{F}_{i_1 i_2 i_3} = \left(p_1^{\otimes 3} + \cdots + p_r^{\otimes 3} \right)_{i_1 i_2 i_3}, \quad \text{for all} \quad (i_1, i_2, i_3) \in \Omega. \quad (3)$$

In the definition, $\mathcal{F}$ has the decomposition $\mathcal{F} = \sum_{i=1}^r \omega_i \mu_i \otimes \mu_i \otimes \mu_i$, which is unique when r is small. Thus, the recovered decomposition can be used to estimate ω_i, μ_i of the Gaussian mixture model and then covariance matrices Σ_i can be found by considering $M_3 - \mathcal{F}$.

Tensor Decompositions with Incomplete Entries

The tensor $\mathcal{F}$ is rewritten as

$$\mathcal{F} = \lambda_1 \begin{pmatrix} 1 \\ u_1 \end{pmatrix}^{\otimes 3} + \cdots + \lambda_r \begin{pmatrix} 1 \\ u_r \end{pmatrix}^{\otimes 3}, \quad (4)$$

where $\lambda_i = \omega_i (\mu_i)_1^3$ and $u_i = (\mu_i)_{2:d}/(\mu_i)_1 \in \mathbb{C}^{d-1}$. In the following, we discuss how to find the decomposition of $\mathcal{F}$ from partial entries $(\mathcal{F})_\Omega$.

- When $r \le \frac{d}{2} - 1$, there is a unique generating matrix $G := \big(G(k, e_i + e_j)\big)_{1 \le k, i \le r < j \le d-1}$ of $\mathcal{F}$ such that

$$\sum_{k=1}^{r} G(k, e_i + e_j) \mathcal{F}_{0kt} - \mathcal{F}_{ijt} = 0, \quad t = 0, 1, \dots, d-1, \quad (5)$$

  for all $1 \le i \le r < j \le d-1$ and $0 \le t \le d-1$. By choosing $t \ge r+1, t \ne j$, entries $\mathcal{F}_{0kt}, \mathcal{F}_{ijt}$ in the above equations are parts of $(\mathcal{F})_\Omega$ and hence are known. Thus, the matrix G can be found by solving the above linear equations.

- For $l = r+1, \dots, d-1$, it holds that

$$N_l(G) \begin{pmatrix} (u_i)_1 \\ \vdots \\ (u_i)_r \end{pmatrix} = (u_i)_l \begin{pmatrix} (u_i)_1 \\ \vdots \\ (u_i)_r \end{pmatrix}, \text{ where, } N_l(G) = \begin{pmatrix} G(1, e_1 + e_l) & \cdots & G(r, e_1 + e_l) \\ \vdots & \ddots & \vdots \\ G(1, e_r + e_l) & \cdots & G(r, e_r + e_l) \end{pmatrix}.$$

  The above equations illustrate that vectors $(u_1)_{1:r}, \dots, (u_r)_{1:r}$ are common eigenvectors of $N_{r+1}, \dots, N_{d-1}$ and $(u_1)_l, \dots, (u_r)_l$ are corresponding eigenvalues of N_l. Thus, vectors $u_1, \dots, u_r$ can be recovered by finding common eigenvectors and corresponding eigenvalues of $N_{r+1}, \dots, N_{d-1}$. Finally, scalars $\lambda_1, \dots, \lambda_r$ are obtained by solving a linear system.

Theorem 1. *Suppose the tensor $\mathcal{F}$ has the decomposition as in* (4)*, where $r \le \frac{d}{2} - 1$ and $\{(u_i)_{1:r}\}_{i=1}^r, \{(u_i)_{r+1:d-1}\}_{i=1}^r$ are both linearly independent, then our algorithm with the input $(\mathcal{F})_\Omega$ must find the decomposition* (4).

When we only have an estimation $(\widehat{\mathcal{F}})_\Omega$ of $(\mathcal{F})_\Omega$, our algorithm with the input $(\widehat{\mathcal{F}})_\Omega$ can still produce a rank-r approximation of $\mathcal{F}$.

Theorem 2. *Suppose the tensor $\mathcal{F} = (p_1)^{\otimes 3} + \cdots + (p_r)^{\otimes 3}$ has rank $r \le \frac{d}{2} - 1$ and satisfies some conditions. Let $\mathcal{F} \approx (\hat{p}_1)^{\otimes 3} + \cdots + (\hat{p}_r)^{\otimes 3}$ be the rank-r approximation produced by our algorithm with the input $(\widehat{\mathcal{F}})_\Omega$. If the distance $\epsilon := \|(\widehat{\mathcal{F}} - \mathcal{F})_\Omega\|$ is small enough, then it holds*

$$\|\hat{p}_i - p_i\| = O(\epsilon),$$

up to a permutation of $(p_1, \dots, p_r)$, where the constant in $O(\cdot)$ only depends on $\mathcal{F}$.

Learning Diagonal Gaussian Mixture Models

For simplicity, we assume that moments $M_1 := \mathbb{E}[x]$ and $M_3 := \mathbb{E}[x \otimes x \otimes x]$ are given exactly.

- The decomposition of $\mathcal{F}$ can be recovered from $(\mathcal{F})_\Omega = (M_3)_\Omega$ by using incomplete tensor decomposition. The recovered decomposition can be written as $\mathcal{F} = \sum_{i=1}^r (p_i)^{\otimes 3}$, where $p_i := \sqrt[3]{\omega_i} \mu_i$. It holds that

$$M_1 = \omega_1 \mu_1 + \cdots + \omega_r \mu_r = \omega_1^{2/3} p_1 + \cdots + \omega_r^{2/3} p_r.$$

  The scalars $\omega_1^{2/3}, \dots, \omega_r^{2/3}$ can be found by solving the above linear system. Therefore, weights ω_i are obtained and the mean vectors μ_i are recovered by $\mu_i = p_i / \sqrt[3]{\omega_i}$.

- Define the tensor $\mathcal{A}$ such that

$$\mathcal{A} := M_3 - \mathcal{F} = \sum_{j=1}^{d} (a_j \otimes e_j \otimes e_j + e_j \otimes a_j \otimes e_j + e_j \otimes e_j \otimes a_j),$$

  where $a_j = \sum_{i=1}^{r} \omega_i \sigma_{ij}^2 \mu_i$. The above equations are linear systems in σ_{ij}^2. Therefore, covariance matrices Σ_i are obtained by solving those linear equations.

Simulations

We perform classifications on synthetic data. Our algorithm is compared with the classical Expectation-Maximization (EM) algorithm under various dimensions and ranks.

Accuracy	$d=20$			$d=30$			$d=40$		
	$r=3$	$r=5$	$r=7$	$r=4$	$r=8$	$r=11$	$r=6$	$r=10$	$r=15$
Ours	0.9861	0.9740	0.9659	0.9965	0.9923	0.9895	0.9990	0.9981	0.9971
EM	0.9763	0.9400	0.9252	0.9684	0.9277	0.9219	0.9117	0.8931	0.9111

Accuracy	$d=50$			$d=60$		
	$r=7$	$r=13$	$r=18$	$r=8$	$r=15$	$r=22$
Ours	0.9997	0.9995	0.9993	0.9999	0.9998	0.9995
EM	0.8997	0.9073	0.9038	0.8874	0.8632	0.8929

Table 1: Classification accuracy on simulations

Texture Classification

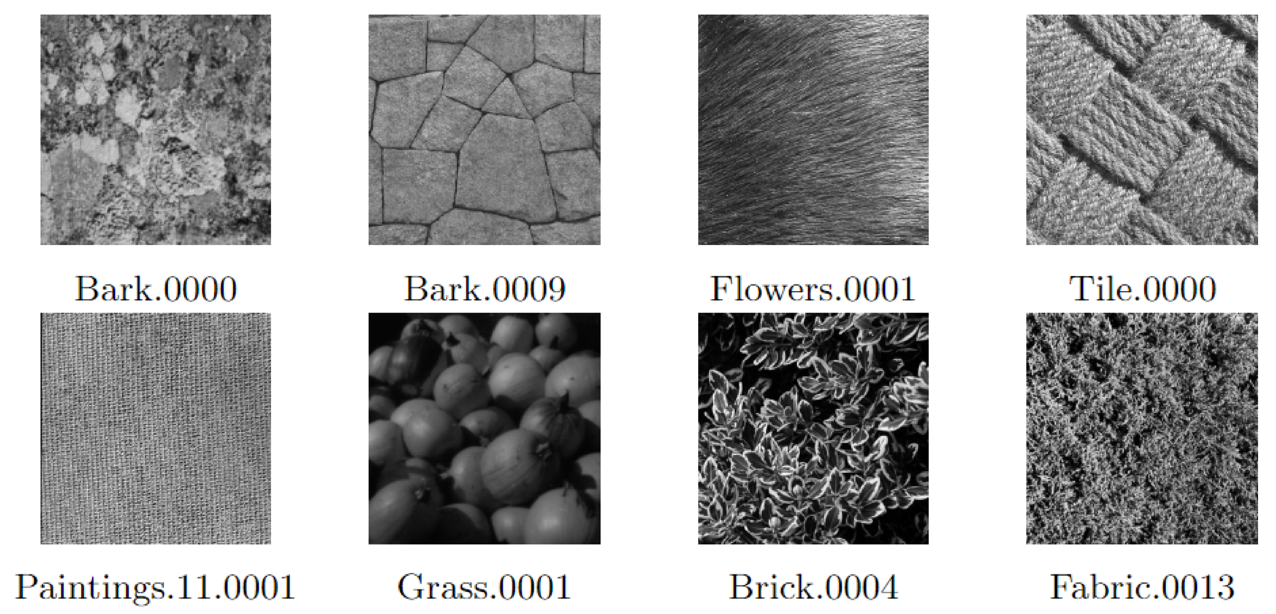

Fig. 2: Textures

Every texture is divided into 256 subimages, among which 160 subimages are uses for training and the remaining are used for testing. Our algorithm and EM algorithm are used to fit the Gaussian mixture model to each texture. The classification accuracy on test subimages is reported in the following table.

Accuracy	Our Algorithm	EM
Flowers.0001	0.8137	0.6315
Tile.0000	0.8219	0.7239
Paintings.11.0001	0.8047	0.7350
Grass.0001	0.9841	0.9068
Brick.0004	0.9406	0.8854
Fabric.0013	0.9220	0.9048
Bark.0000	0.5376	0.8413
Bark.0009	0.5107	0.7150

Table 2: Classification accuracy on 8 textures

References

[1] Bingni Guo, Jiawang Nie, and Zi Yang. "Learning diagonal Gaussian mixture models and incomplete tensor decompositions". In preparation.