Estimation in Mixture Models through Implicit Tensor Decompositions

Joe Kileel

University of Texas at Austin, Department of Mathematics

Oden Institute for Computational Engineering and Sciences

January 4, 2023

Symmetric Moment Tensor

Given data $\mathbf{x}_1, \ldots, \mathbf{x}_p \in \mathbb{R}^n$. It is often useful to form the moment

$$\mathbf{M}_d = \frac{1}{p} \sum_{i=1}^{p} \mathbf{x}_i^{\otimes d} \in S^d(\mathbb{R}^n)$$

where $(\mathbf{x}^{\otimes d})_{i_1,\ldots,i_d} = \mathbf{x}_{i_1} \ldots \mathbf{x}_{i_d}$ for each $(i_1, \ldots, i_d) \in [n]^d$.

Symmetric Moment Tensor

Given data $\mathbf{x}_1, \ldots, \mathbf{x}_p \in \mathbb{R}^n$. It is often useful to form the moment

$$\mathbf{M}_d = \frac{1}{p} \sum_{i=1}^{p} \mathbf{x}_i^{\otimes d} \in S^d(\mathbb{R}^n)$$

where $(\mathbf{x}^{\otimes d})_{i_1,\ldots,i_d} = \mathbf{x}_{i_1} \ldots \mathbf{x}_{i_d}$ for each $(i_1, \ldots, i_d) \in [n]^d$.

- $d = 1 \rightsquigarrow$ sample average
- $d = 2 \rightsquigarrow$ sample covariance matrix (uncentered)
- $d = 3 \rightsquigarrow n \times n \times n$ real symmetric tensor (3rd moment)

Symmetric Moment Tensor

Given data $\mathbf{x}_1, \dots, \mathbf{x}_p \in \mathbb{R}^n$. It is often useful to form the moment

$$\mathbf{M}_d = \frac{1}{p} \sum_{i=1}^{p} \mathbf{x}_i^{\otimes d} \in S^d(\mathbb{R}^n)$$

where $(\mathbf{x}^{\otimes d})_{i_1,\dots,i_d} = \mathbf{x}_{i_1} \dots \mathbf{x}_{i_d}$ for each $(i_1, \dots, i_d) \in [n]^d$.

- $d = 1 \rightsquigarrow$ sample average
- $d = 2 \rightsquigarrow$ sample covariance matrix (uncentered)
- $d = 3 \rightsquigarrow n \times n \times n$ real symmetric tensor (3rd moment)

Decomposing the tensor $\mathbf{M}_d$ ***reveals structure in*** $\{\mathbf{x}_1, \dots, \mathbf{x}_p\}$**.**

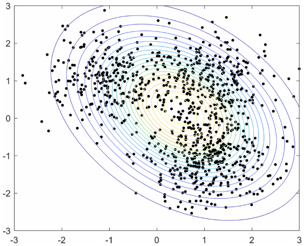
PART I: *Gaussian Mixture Models & CP Tensor Decompositions*

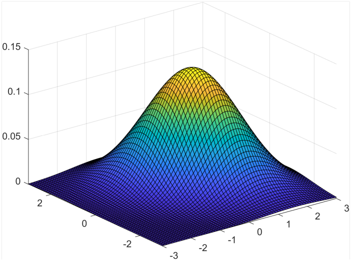
Gaussian Distribution

Gaussian vector: $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$

probability density function: $\frac{\exp(-\frac{1}{2}(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu))}{\sqrt{(2\pi)^n \det(\Sigma)}}$

parameters: $\mu = \mathbb{E}[\mathbf{x}] \in \mathbb{R}^n, \; \Sigma = \mathbb{E}[(\mathbf{x}-\mu)^{\otimes 2}] \in S^2(\mathbb{R}^n)$



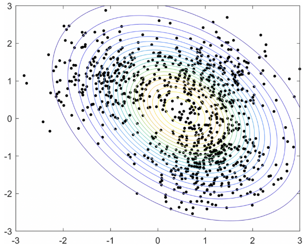


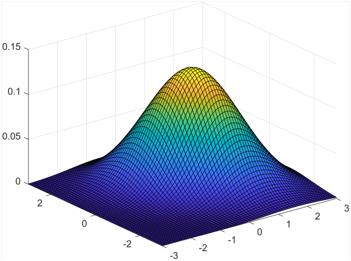
Gaussian Distribution

Gaussian vector: $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$

probability density function: $\frac{\exp(-\frac{1}{2}(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu))}{\sqrt{(2\pi)^n \det(\Sigma)}}$

parameters: $\mu = \mathbb{E}[\mathbf{x}] \in \mathbb{R}^n, \ \Sigma = \mathbb{E}[(\mathbf{x}-\mu)^{\otimes 2}] \in S^2(\mathbb{R}^n)$





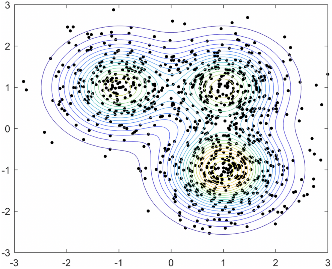
- Limiting average of any (suff. integrable) i.i.d. random vectors
- Marginals are themselves lower-dimensional Gaussians

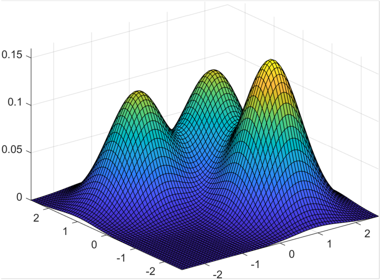
Gaussian Mixture Models

GMM: $\mathbf{x} \sim \sum_{j=1}^{r} \lambda_j \mathcal{N}(\mu_j, \Sigma_j)$

r is the number of components, λ_j are the mixing weights (convex combination)

parameters: $\{(\lambda_j, \mu_j, \Sigma_j) : j = 1, \ldots, r\}$



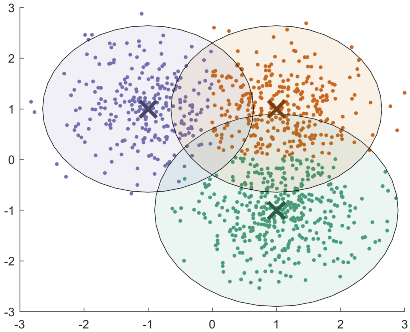


Many Applications of Gaussian Mixtures

Density Estimation

Clustering

Anomaly Detection

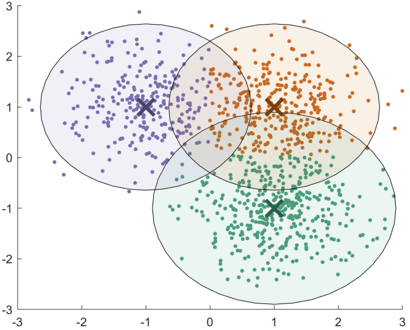


Many Applications of Gaussian Mixtures

Density Estimation

Clustering

Anomaly Detection



GMMs are one of the most prevalent tools in data analysis!

Neat Formula for Moment Tensors of GMM

Lemma (Wick '50, Pereira-K.-Kolda '22)

Let $\mathbf{x}_1, \ldots, \mathbf{x}_p$ *be i.i.d. realizations of a GMM with parameters* $\{(\lambda_j, \mu_j, \Sigma_j)\}$*. Then*

$$\mathbf{M}_d \longrightarrow \sum_{j=1}^{r} \lambda_j \sum_{k=0}^{\lfloor d/2 \rfloor} \binom{d}{2k} \frac{(2k)!}{k! 2^k} \operatorname{sym}(\mu_j^{\otimes (d-2k)} \otimes \Sigma_j^{\otimes k}) \quad \textit{as } p \to \infty.$$

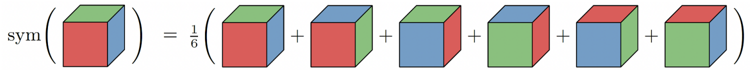
Neat Formula for Moment Tensors of GMM

Lemma (Wick '50, Pereira-K.-Kolda '22)

Let $\mathbf{x}_1, \ldots, \mathbf{x}_p$ *be i.i.d. realizations of a GMM with parameters* $\{(\lambda_j, \mu_j, \Sigma_j)\}$*. Then*

$$\mathbf{M}_d \longrightarrow \sum_{j=1}^{r} \lambda_j \sum_{k=0}^{\lfloor d/2 \rfloor} \binom{d}{2k} \frac{(2k)!}{k!2^k} \operatorname{sym}(\mu_j^{\otimes(d-2k)} \otimes \Sigma_j^{\otimes k}) \quad \text{as } p \to \infty.$$

The proof is most easily done using the bijection Φ from symmetric tensors to homogeneous forms, because $\Phi(\operatorname{sym}(S \otimes T)) = \Phi(S)\Phi(T)$.



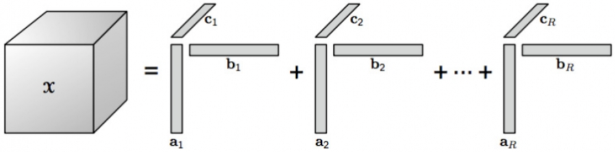
Common $\Sigma \iff$ CP Tensor Decomposition

Lemma (Pereira-Kileel-Kolda '22)

Let $\mathbf{x}_1, \ldots, \mathbf{x}_p$ *be i.i.d. realizations of a GMM with parameters* $\{(\lambda_j, \mu_j, \Sigma)\}$*, i.e. there is a common covariance. Then as* $p \to \infty$*,*

$$\sum_{k=0}^{\lfloor d/2 \rfloor} (-1)^k \binom{d}{2k} \frac{(2k)!}{k! 2^k} \operatorname{sym}(\mathbf{M}_{d-2k} \otimes \Sigma^{\otimes k}) \longrightarrow \sum_{j=1}^{r} \lambda_j \mu_j^{\otimes d}.$$

The right-hand side is a CP symmetric tensor decomposition:



Numerical Algorithm Beating the Curse of Dimensionality

To fit a general GMM to data, consider minimizing the cost function

$$\operatorname{argmin}_{\lambda_j,\mu_j,\Sigma_j} \sum_{k=1}^{d} w_k \|\mathbf{M}_k - (\text{aforementioned formula in parameters})\|_F^2$$

Numerical Algorithm Beating the Curse of Dimensionality

To fit a general GMM to data, consider minimizing the cost function

$$\text{argmin}_{\lambda_j,\mu_j,\Sigma_j} \sum_{k=1}^{d} w_k \|\mathbf{M}_k - (\text{aforementioned formula in parameters})\|_F^2$$

Naively forming the terms would take $\mathcal{O}(pn^d)$ flops and $\mathcal{O}(n^d)$ storage.

Numerical Algorithm Beating the Curse of Dimensionality

To fit a general GMM to data, consider minimizing the cost function

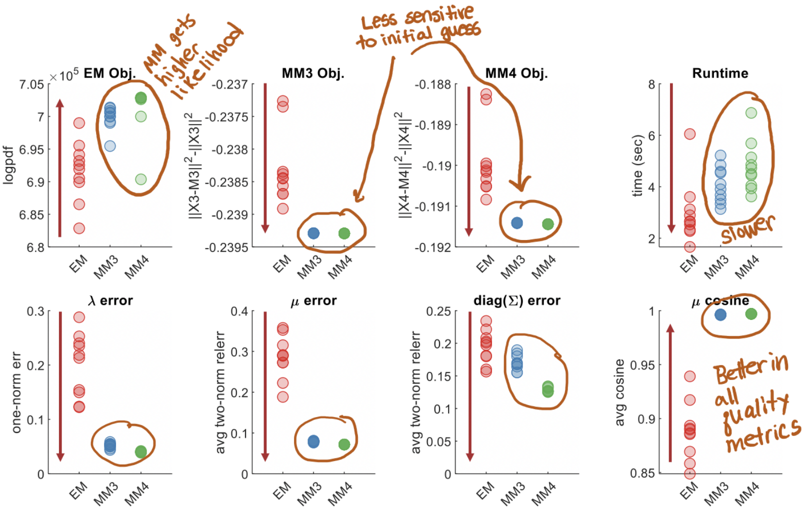
$$\operatorname{argmin}_{\lambda_j, \mu_j, \Sigma_j} \sum_{k=1}^{d} w_k \|\mathbf{M}_k - (\text{aforementioned formula in parameters})\|_F^2$$

Naively forming the terms would take $\mathcal{O}(pn^d)$ flops and $\mathcal{O}(n^d)$ storage.

Theorem (Pereira-Kileel-Kolda '22)

Given the parameters $\lambda_j, \mu_j, \Sigma_j$ and data $\mathbf{x}_i$, there is an algorithm to evaluate the above cost and its gradient in $\mathcal{O}(prn^2 + r^2n^3)$ flops and $\mathcal{O}(rn^2 + pn)$ storage. If Σ_j are diagonal, these drop to $\mathcal{O}(prn + r^2n)$ flops and $\mathcal{O}(rn + pn)$ storage.

In Practice: Method of Moments Can Outperform EM



- Randomly-generated problems with overlapping Gaussians
- $n = 100$, $r = 20$, $p = 8000$, common diagonal Σ
- Compared EM, MM3 (moments $d = 3$), MM4 (moments $d = 4$)

Sketch: Expanding Out The Inner Products

Idea is to operate on moment tensors without forming them!

$$\min_{\theta} f(\theta) \equiv \left\| \frac{1}{p} \sum_{i=1}^{p} \mathbf{x}_i^{\otimes d} - \sum_{j=1}^{m} \lambda_j \mathcal{M}_j^{(d)} \right\|^2$$

$$f(\theta) = \underbrace{\cancel{\left\| \frac{1}{p} \sum_{i=1}^{p} \mathbf{x}_i^{\otimes d} \right\|^2}}_{\text{constant}} + \left\| \sum_{j=1}^{m} \lambda_j \mathcal{M}_j^{(d)} \right\|^2 - 2 \left\langle \frac{1}{p} \sum_{i=1}^{p} \mathbf{x}_i^{\otimes d}, \sum_{j=1}^{m} \lambda_j \mathcal{M}_j^{(d)} \right\rangle$$

$$f(\theta) = C + \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i \lambda_j \underbrace{\left\langle \mathcal{M}_i^{(d)}, \mathcal{M}_j^{(d)} \right\rangle}_{\text{dot product of 2 moments}} - \frac{2}{p} \sum_{i=1}^{p} \sum_{j=1}^{m} \lambda_j \underbrace{\left\langle \mathbf{x}_i^{\otimes d}, \mathcal{M}_j^{(d)} \right\rangle}_{\text{dot product of moment + vector}}$$

Example Calculation: $d = 3$

$$f(\theta) = C + \sum_{i=1}^{m}\sum_{j=1}^{m} \lambda_i \lambda_j \left\langle \mathcal{M}_i^{(d)}, \mathcal{M}_j^{(d)} \right\rangle - \frac{2}{p} \sum_{i=1}^{p}\sum_{j=1}^{m} \lambda_j \left\langle \mathbf{x}_i^{\otimes d}, \mathcal{M}_j^{(d)} \right\rangle$$

$$\mathcal{M}_j^{(3)} = \boldsymbol{\mu}_j^{\otimes 3} + 3\,\mathrm{sym}(\boldsymbol{\mu}_j \otimes \boldsymbol{\Sigma}_j)$$

$$\begin{aligned} \langle \mathbf{x}_i^{\otimes 3}, \mathcal{M}_j^{(3)} \rangle &= \left\langle \mathbf{x}_i^{\otimes 3}, \boldsymbol{\mu}_j^{\otimes 3} \right\rangle + 3\left\langle \mathbf{x}_i^{\otimes 3}, \mathrm{sym}(\boldsymbol{\mu}_j \otimes \boldsymbol{\Sigma}_j) \right\rangle \\ &= (\mathbf{x}_i^\intercal \boldsymbol{\mu}_j)^3 + 3\left\langle \mathbf{x}_i^{\otimes 3}, \boldsymbol{\mu}_j \otimes \boldsymbol{\Sigma}_j \right\rangle \\ &= (\mathbf{x}_i^\intercal \boldsymbol{\mu}_j)^3 + 3(\mathbf{x}_i^\intercal \boldsymbol{\mu}_j)(\mathbf{x}_i^\intercal \boldsymbol{\Sigma}_j \mathbf{x}_i) \end{aligned}$$

$$\left\langle \mathbf{a}^{\otimes 3}, \mathrm{sym}(\mathcal{B}) \right\rangle = \left\langle \mathbf{a}^{\otimes 3}, \mathcal{B} \right\rangle$$

$$\left\langle \mathbf{a}^{\otimes 3}, \mathbf{b}^{\otimes 3} \right\rangle = (\mathbf{a}^\intercal \mathbf{b})^3$$

$$\left\langle \mathbf{a}^{\otimes 3}, \mathbf{b} \otimes \mathbf{C} \right\rangle = \mathbf{a}^\intercal \mathbf{b}\, \mathbf{a}^\intercal \mathbf{C} \mathbf{a}$$

Computing terms $\langle \mathbf{M}_i^{(d)}, \mathbf{M}_j^{(d)} \rangle$ more involved (Bell polynomials).

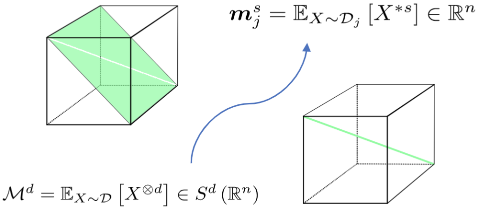
PART II: *Conditionally-Independent Mixture Models & Incomplete CP Tensor Decompositions (briefly)*

Conditionally-Independent Mixture Models

I extended these ideas to other noise models with UT Ph.D. student Yifan Zhang. We considered **mixtures of product distributions on $\mathbb{R}^n$**:

$$\mathcal{D} = \sum_{j=1}^{r} \lambda_j \mathcal{D}_j \quad \text{for} \quad \mathcal{D}_j = \otimes_{i=1}^{n} \mathcal{D}_{ij}.$$

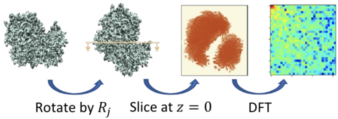
where $\mathcal{D}_{ij}$ are any distributions on $\mathbb{R}$ whose moments exist.

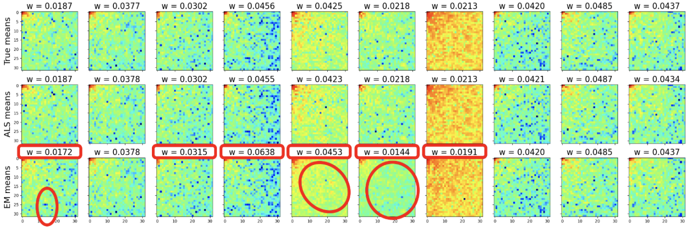


Estimating moments of $\mathcal{D}_{ij}$ from sample moments of $\mathcal{D}$ is a CP tensor decomposition problem where the diagonal is missing.

Application: Clustering X-Ray Free Laser Images

$$I_j := |\mathcal{P}\mathcal{F}(\phi \circ R_j)| : \mathbb{R}^2 \to \mathbb{R}$$





- Simulation with $n = 1024, r = 30, p = 20000$.
- Noise is pixelwise Poisson. Our algorithm doesn't know this, but EM does.
- We take ~ 40 min to converge. Error 0.9% in weights, 0.5% in means.
- EM is initialized with best of 30 k-means runs. We then run EM three times with different seeds. It takes $\sim 50 - 70$ min. Error in means is $> 13\%$.

CONCLUSIONS

Summary

- Moment formulas for general Gaussian Mixture Models and a tensor-based algorithm avoiding exponential cost in order d.
- Extensions to mixtures of other distributions with applications.
- It is useful to build algorithms to decompose sample moments which do not explicitly form the high-dimensional tensors.

References

- "Tensor moments of Gaussian mixture models: theory and applications", J. Pereira, J. Kileel, T. Kolda, arXiv:2202.06930
- "Moment estimation for nonparametric mixture models through implicit tensor decomposition", Y. Zhang, J. Kileel, arXiv:2210.14386

THANK YOU!