# Learning mixtures of spherical Gaussians: moment methods and spectral decompositions

Machine Learning Reading Group Fall 2020

---

Jonathan Wilder Lavington

October 28, 2020

University of British Columbia, Department of Computer Science

# Overview

THE UNIVERSITY OF BRITISH COLUMBIA

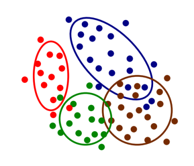
**What are the main contributions?**

- Provide a computationally efficient and statistically consistent moment-based estimator for mixtures of spherical Gaussians
- Derive computational and information-theoretic barriers to efficient estimation in mixture models (for spherical Gaussians).
- Make connections to estimation problems related to independent component analysis (ICA).

# What is the Goal?

## Just as a little refresher...

### Problem Statement



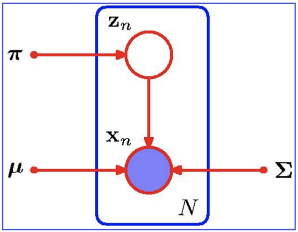
- K sub-populations
- Each modeled as multivariate Gaussian.
- Each label picked according to some mixture weight.

# What is the Goal?

THE UNIVERSITY OF BRITISH COLUMBIA

## In case you prefer the PGM...

### Problem Statement




- K sub-populations
- Each modeled as multivariate Gaussian.
- Each label picked according to some mixture weight.

**Note that the "labels" are not observed.**

# What is the Goal?



**What do we want to do?**

We want an efficient algorithm that approximately recovers parameters from samples.

**One example of such an algorithm...**

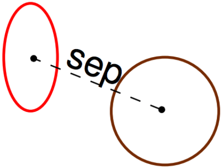
This can be done with a local search for maximum likelihood parameters (EM algorithm).

# When Can We Make Efficient Algorithms

## Well Separated Mixtures

Estimation is easier if there is a large minimum separation between component means.

## What is Separation?




$$\text{sep} := \min_{i \neq j} \frac{\|\vec{\mu}_i - \vec{\mu}_j\|}{\max\{\sigma_i, \sigma_j\}}.$$

**This is not required in general but leads to exponential (**$\exp(\Omega(k))$ **where k is the number of clusters) running time / sample size.**

# Main Paper Result



**Result**

We can create an efficient algorithms for "non-degenerate" models in high-dimensions ($d \geq k$) with spherical covariances.

**How is this done?**

Using the Method of Moments, they approximate the first three moments of the GMM and then solve for the parameters of the true models (means, covariances, and labels) with respect to those estimated moments.

# Notation

The Spherical Gaussian Mixture model is defined in as follows.

- Let $w_i$ be the probability of choosing a component $i \in [k] := \{1, 2., ....., k\}$
- Let $\mu_1, \mu_2, ..., \mu_k \in \mathbb{R}^d$ be the component vectors
- Let $\sigma_1^2, \sigma_2^2, ..., \sigma_k^2 \geq 0$ be component variances

We then define two matrices for convenience:

$$w := [w_1, w_2, ..., w_k]^\top \in \mathbb{R}^k, \qquad A := [\mu_1|\mu_2|...|\mu_k] \in \mathbb{R}^{d\times k}$$

**Assuming the following data distribution...**

We assume that the data was generated following,

$$x \sim \sum_{i=1}^{k} \mathcal{I}[c = i]\, z_i \quad \text{where} \quad z_i \sim \mathcal{N}(\mu_i^*, \sigma_i^{2*} I), c \sim Cat(w^*)$$

**Algorithm Idea**

Now we take these samples, compute empirical estimates of some of the moments, and then match our current set of parameters to those moments.

## Definition of Moment Generating function

$$M_{\mathbf{X}}(\mathbf{t}) := \mathrm{E}\left(e^{\mathbf{t}^{\mathrm{T}}\mathbf{X}}\right).$$

## Recall the Taylor series expansion

$$e^{t\,X} = 1 + t\,X + \frac{t^2\,X^2}{2!} + \frac{t^3\,X^3}{3!} + \cdots + \frac{t^n\,X^n}{n!} + \cdots.$$

## Taylor Expansion of Moment Generating functions

$$\begin{aligned} M_X(t) = \mathrm{E}(e^{t\,X}) &= 1 + t\,\mathrm{E}(X) + \frac{t^2\,\mathrm{E}(X^2)}{2!} + \frac{t^3\,\mathrm{E}(X^3)}{3!} + \cdots + \frac{t^n\,\mathrm{E}(X^n)}{n!} + \cdots \\ &= 1 + tm_1 + \frac{t^2 m_2}{2!} + \frac{t^3 m_3}{3!} + \cdots + \frac{t^n m_n}{n!} + \cdots, \end{aligned}$$

## Taylor Expansion of Moment Generating functions

$$
\begin{aligned}
M_X(t) = \mathrm{E}(e^{t\,X}) &= 1 + t\,\mathrm{E}(X) + \frac{t^2\,\mathrm{E}(X^2)}{2!} + \frac{t^3\,\mathrm{E}(X^3)}{3!} + \cdots + \frac{t^n\,\mathrm{E}(X^n)}{n!} + \cdots \\
&= 1 + tm_1 + \frac{t^2 m_2}{2!} + \frac{t^3 m_3}{3!} + \cdots + \frac{t^n m_n}{n!} + \cdots,
\end{aligned}
$$

## Important Fact

Recall from intro to probability, that a moment generating function uniquely identifies a given distribution.

# Method of Moments



## How do we apply the method of moments to GMMs?

- Define some subset of moments in terms of the parameters (means, variances, labels)
- Then solve for these parameters using empirical estimates of these moments.

## Couple Questions

- Which moments to use?
- How do we approximate them?

# Method of Moments: Which moments to use?

THE UNIVERSITY OF BRITISH COLUMBIA

## Low-Order Estimates

| moment order | reliable estimates? | unique solution? |
|---|---|---|
| 1st, 2nd | ✓ | ✗ |

**1st- and 2nd-order moments** (*e.g.*, mean, covariance)

- Fairly easy to get reliable estimates.

$$\mathbb{E}_{\vec{x} \in S}[\vec{x} \otimes \vec{x}] \approx \mathbb{E}_{\theta^\star}[\vec{x} \otimes \vec{x}]$$

- Can have multiple solutions to moment equations.

$$\mathbb{E}_{\theta_1}[\vec{x} \otimes \vec{x}] \approx \mathbb{E}_{\vec{x} \in S}[\vec{x} \otimes \vec{x}] \approx \mathbb{E}_{\theta_2}[\vec{x} \otimes \vec{x}], \quad \theta_1 \neq \theta_2$$

# Method of Moments: Which moments to use?

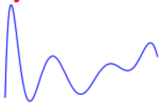
THE UNIVERSITY OF BRITISH COLUMBIA

## High-Order Estimates

| moment order | reliable estimates? | unique solution? |
|---|---|---|
| 1st, 2nd | ✓ | ✗ |
| $\Omega(k)^{th}$ | ✗ | ✓ |

**$\Omega(k)^{th}$-order moments** (*e.g.*, $\mathbb{E}_\theta[\text{degree-}k\text{-poly}(\vec{x})]$)

- Uniquely pins down the solution.
- Empirical estimates very unreliable.



# Method of Moments: Which moments to use?

THE UNIVERSITY OF BRITISH COLUMBIA

## Big Idea

$$\mathbb{E}_\theta[\vec{x} \otimes \vec{x}] = \sum_{i=1}^{k} w_i \, \vec{\mu}_i \otimes \vec{\mu}_i + \text{some sparse matrix;}$$

$$\mathbb{E}_\theta[\vec{x} \otimes \vec{x} \otimes \vec{x}] = \sum_{i=1}^{k} w_i \, \vec{\mu}_i \otimes \vec{\mu}_i \otimes \vec{\mu}_i + \text{some sparse tensor.}$$

**Trick**: "sparse stuff" can be estimated and thus removed.

**Upshot**: the following can be readily estimated (with $\widehat{M}$, $\widehat{T}$).

$$M_{\theta\star} := \sum_{i=1}^{k} w_i^\star \, \vec{\mu}_i^\star \otimes \vec{\mu}_i^\star \quad \text{and} \quad T_{\theta\star} := \sum_{i=1}^{k} w_i^\star \, \vec{\mu}_i^\star \otimes \vec{\mu}_i^\star \otimes \vec{\mu}_i^\star.$$

**Claim**: $\{(\vec{\mu}_i, w_i)\}$ uniquely determined by $M_\theta$ and $T_\theta$.

# How do we know this works?



## Important Definitions

Define $M_\theta : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and $T_\theta : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ as bi-linear and tri-linear functions.

## Variational Lemma

**Lemma**

*If $\{\vec{\mu}_i\}$ are linearly independent and all $w_i > 0$, then each of the k distinct, isolated local maximizers $\vec{u}^*$ of*

$$\max_{\vec{u} \in \mathbb{R}^d} T_\theta(\vec{u}, \vec{u}, \vec{u}) \quad s.t. \quad M_\theta(\vec{u}, \vec{u}) \leq 1$$

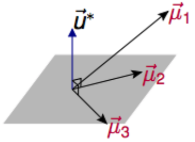
*satisfies, for some $i \in [k]$,*

$$M_\theta(\cdot, \vec{u}^*) = \sqrt{w_i}\, \vec{\mu}_i, \qquad T_\theta(\vec{u}^*, \vec{u}^*, \vec{u}^*) = \frac{1}{\sqrt{w_i}}.$$

## Orthogonal Directions and Solutions

$$\max_{\vec{u}\in\mathbb{R}^d} \sum_{i=1}^{k} w_i \langle \vec{\mu}_i, \vec{u}\rangle^3 \text{ s.t. } \sum_{i=1}^{k} w_i \langle \vec{\mu}_i, \vec{u}\rangle^2 \leq 1$$

Maximizers are directions $\vec{u}^*$ orthogonal to all but one $\vec{\mu}_j$.




Combine with constraints $w_j\langle\vec{\mu}_j, \vec{u}^*\rangle^2 \leq 1$ to get

$$M\vec{u}^* = \left(\sum_{i=1}^{k} w_i\, \vec{\mu}_i \otimes \vec{\mu}_i\right)\vec{u}^* = \sum_{i=1}^{k} w_i\, \vec{\mu}_i\langle\vec{\mu}_i, \vec{u}^*\rangle = \pm\sqrt{w_j}\,\vec{\mu}_j. \blacksquare$$

THE UNIVERSITY OF BRITISH COLUMBIA

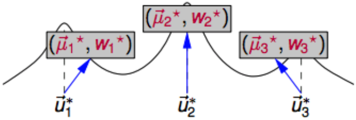
## Getting an Approximate Solution

Effectively want to solve

$$\min_\theta \|T_\theta - \widehat{T}\|^2 \quad \text{s.t.} \quad M_\theta = \widehat{M}. \qquad (\dagger)$$

**Not convex in parameters** $\theta = \{(\vec{\mu}_i, w_i)\}$.

**What we do:** find one component $(\vec{\mu}_i, w_i)$ at a time, using local optimization of related (also non-convex) objective function.




**New robust algorithm for "tensor eigen-decomposition"** efficiently approximates *all* local optima, each corresponding to a component. $\longrightarrow$ Near-optimal solution to ($\dagger$). ∎

## Initialization and Convergence

Want to find *all* local maximizers of

$$\max_{\vec{u} \in \mathbb{R}^d} \widehat{T}(\vec{u}, \vec{u}, \vec{u}) \quad \text{s.t.} \quad \widehat{M}(\vec{u}, \vec{u}) \leq 1. \tag{‡}$$

Must address **initialization** and **convergence** issues.

**Crucially using special tensor structure of $\widehat{T} \approx T_{\theta^\star}$, together with non-linearity of $\vec{u} \mapsto \widehat{T}(\,\cdot\,, \vec{u}, \vec{u})$:**

- Random initialization is good with significant probability.
  ("Good" ⇒ simple iteration will quickly converge to some local max.)
- Can check if initialization was good by checking objective value after a few steps.
  - If value large enough: initialization was good; improve by taking a few more steps.
  - Else: abandon and restart.

## Condition from Lemma 1 (Restated)

**Condition 1** (Non-degeneracy)**.** The component means span a $k$-dimensional subspace (*i.e.*, the matrix $A$ has column rank $k$), and the vector $w$ has strictly positive entries.

# Moment Based Estimation: Theorems

## Computing Moments with Tensor Arithmetic

**Theorem 1** (Observable moment structure). *Assume Condition 1 holds. The average variance* $\bar{\sigma}^2 := \sum_{i=1}^{k} w_i \sigma_i^2$ *is the smallest eigenvalue of the covariance matrix* $\mathbb{E}[(x - \mathbb{E}[x])(x - \mathbb{E}[x])^\top]$. *Let* $v \in \mathbb{R}^d$ *be any unit norm eigenvector corresponding to the eigenvalue* $\bar{\sigma}^2$. *Define*

$$M_1 := \mathbb{E}[x(v^\top(x - \mathbb{E}[x]))^2] \in \mathbb{R}^d,$$

$$M_2 := \mathbb{E}[x \otimes x] - \bar{\sigma}^2 I \in \mathbb{R}^{d \times d},$$

$$M_3 := \mathbb{E}[x \otimes x \otimes x] - \sum_{i=1}^{d} \big(M_1 \otimes e_i \otimes e_i + e_i \otimes M_1 \otimes e_i + e_i \otimes e_i \otimes M_1\big) \in \mathbb{R}^{d \times d \times d}$$

*(where* $\otimes$ *denotes tensor product, and* $\{e_1, e_2, \ldots, e_d\}$ *is the coordinate basis for* $\mathbb{R}^d$*). Then*

$$M_1 = \sum_{i=1}^{k} w_i \; \sigma_i^2 \mu_i, \qquad M_2 = \sum_{i=1}^{k} w_i \; \mu_i \otimes \mu_i, \qquad M_3 = \sum_{i=1}^{k} w_i \; \mu_i \otimes \mu_i \otimes \mu_i.$$

## Estimating parameters using moments

**Theorem 2** (Moment-based estimator). *The following can be added to the results of Theorem 1. Suppose $\eta^\top \mu_1, \eta^\top \mu_2, \ldots, \eta^\top \mu_k$ are distinct and non-zero (which is satisfied almost surely, for instance, if $\eta$ is chosen uniformly at random from the unit sphere in $\mathbb{R}^d$). Then the matrix*

$$M_{\text{GMM}}(\eta) := M_2^{\dagger 1/2} M_3(\eta) M_2^{\dagger 1/2}$$

*is diagonalizable (where $^\dagger$ denotes the Moore-Penrose pseudoinverse); its non-zero eigenvalue / eigenvector pairs $(\lambda_1, v_1), (\lambda_2, v_2), \ldots, (\lambda_k, v_k)$ satisfy $\lambda_i = \eta^\top \mu_{\pi(i)}$ and $M_2^{1/2} v_i = s_i \sqrt{w_{\pi(i)}} \mu_{\pi(i)}$ for some permutation $\pi$ on $[k]$ and signs $s_1, s_2, \ldots, s_k \in \{\pm 1\}$. The $\mu_i$, $\sigma_i^2$, and $w_i$ are recovered (up to permutation) with*

$$\mu_{\pi(i)} = \frac{\lambda_i}{\eta^\top M_2^{1/2} v_i} M_2^{1/2} v_i, \qquad \sigma_i^2 = \frac{1}{w_i} e_i^\top A^\dagger M_1, \qquad w_i = e_i^\top A^\dagger \mathbb{E}[x].$$

Here for a third order tensor $T \in \mathbb{R}^{d \times d \times d}$ $T(\eta) = \sum_{i_1=1}^d \sum_{i_2=1}^d \sum_{i_3=1}^d T_{i_1,i_2,i_3} \eta_{i_3} e_{i_1} \otimes e_{i_2} \in \mathbb{R}^{d \times d}$ for any vector in $\mathbb{R}^d$

**Where does this leave us?**

Combining Theorem 1 and Theorem 2 basically gives us a plug in estimator that can be converted into an algorithm.

# Creating an Algorithm: Moment Estimates



## Moment Estimates

Let $\{(x_i, h_i) : i \in [n]\}$ be $n$ i.i.d. copies of (x,h), and $S := \{x_1, x_2, ....x_n\}$ with $\bar{S}$ being some independent copy also of size n. Then we can define our empirical moment estimates:

$$\mu := \mathbb{E}[x], \qquad \mathcal{M}_2 := \mathbb{E}[xx^\top], \qquad \mathcal{M}_3 := \mathbb{E}[x \otimes x \otimes x],$$

$$\hat{\mu} := \frac{1}{|\mathcal{S}|}\sum_{x \in \mathcal{S}} x, \qquad \widehat{\mathcal{M}}_2 := \frac{1}{|\mathcal{S}|}\sum_{x \in \mathcal{S}} xx^\top, \qquad \underline{\widehat{\mathcal{M}}_3} := \frac{1}{|\underline{\mathcal{S}}|}\sum_{x \in \underline{\mathcal{S}}} x \otimes x \otimes x, \qquad \underline{\hat{\mu}} := \frac{1}{|\underline{\mathcal{S}}|}\sum_{x \in \underline{\mathcal{S}}} x.$$

# Creating an Algorithm: Solving for Parameters

THE UNIVERSITY OF BRITISH COLUMBIA

## Relationship of parameters and Moments

If we restrict ourselves to the case where $\sigma_1^2 = \sigma_2^2, \ldots, \sigma_k^2$, then the relationships between each moment and our mixture parameters is defined under the following lemma:

**Lemma 3** (Structure of moments).

$$\mu = \sum_{i=1}^{k} w_i \mu_i,$$

$$\mathcal{M}_2 = \sum_{i=1}^{k} w_i \mu_i \mu_i^\top + \sigma^2 I,$$

$$\mathcal{M}_3 = \sum_{i=1}^{k} w_i \mu_i \otimes \mu_i \otimes \mu_i + \sigma^2 \sum_{j=1}^{d} \Big( \mu \otimes e_j \otimes e_j + e_j \otimes \mu \otimes e_j + e_j \otimes e_j \otimes \mu \Big).$$

THE UNIVERSITY OF BRITISH COLUMBIA

### Notational Aside

For a third-order tensor $Y \in \mathbb{R}^{m \times m \times m}$ and $U, V, W \in \mathbb{R}^{m \times n}$, this paper lets $Y[U, V, W] \in \mathbb{R}^{n \times n \times n}$ denote the third order tensor given by:

$$Y[U,V,W]_{j_1,j_2,j_3} = \sum_{1 \leq i_1, i_2, i_3 \leq m} U_{i_1,j_1} V_{i_2,j_2} W_{i_3,j_3} Y_{i_1,i_2,i_3}, \quad \forall j_1, j_2, j_3 \in [n].$$

# LearnGMM Algorithm: Technical Overview



## What are the key steps?

The algorithm can be broken up unto a couple of Key parts

- Split data set (of size 2n) into $S$ and $\bar{S}$ (each of size n)
- Use $S$ to compute empirical moments $\hat{\mu}$, and $\hat{\mathcal{M}}_3$ whhich are used to construct $\hat{\sigma}^2$, $\hat{M}_2$, $\hat{W}$, and $\hat{B}$
- Use $\bar{S}$ to compute $\hat{W}^\top \underline{\hat{\mu}}$ and $\underline{\hat{\mathcal{M}}}_3[\hat{W}, \hat{W}, \hat{W}]$, which are then used to construct $\underline{\hat{M}}_3[\hat{W}, \hat{W}, \hat{W}]$
- Do some magic with random projections....

# LearnGMM Algorithm: Technical Overview

THE UNIVERSITY OF BRITISH COLUMBIA

## Part 1

1. Using the first half of the sample, compute empirical mean $\hat{\mu}$ and empirical second-order moments $\widehat{\mathcal{M}}_2$.
2. Let $\hat{\sigma}^2$ be the $k$-th largest eigenvalue of the empirical covariance matrix $\widehat{\mathcal{M}}_2 - \hat{\mu}\hat{\mu}^\top$.
3. Let $\widehat{M}_2$ be the best rank-$k$ approximation to $\widehat{\mathcal{M}}_2 - \hat{\sigma}^2 I$

$$\widehat{M}_2 := \arg \min_{X \in \mathbb{R}^{d \times d} : \mathrm{rank}(X) \leq k} \|(\widehat{\mathcal{M}}_2 - \hat{\sigma}^2 I) - X\|_2$$

   which can be obtained via the singular value decomposition.
4. Let $\widehat{U} \in \mathbb{R}^{d \times k}$ be the matrix of left orthonormal singular vectors of $\widehat{M}_2$.
5. Let $\widehat{W} := \widehat{U}(\widehat{U}^\top \widehat{M}_2 \widehat{U})^{\dagger 1/2}$, where $X^\dagger$ denotes the Moore-Penrose pseudoinverse of a matrix $X$.
   Also define $\widehat{B} := \widehat{U}(\widehat{U}^\top \widehat{M}_2 \widehat{U})^{1/2}$.

# LearnGMM Algorithm: Technical Overview



## Part 2

6. Using the second half of the sample, compute whitened empirical averages $\widehat{W}^\top \underline{\hat{\mu}}$ and third-order moments $\underline{\widehat{\mathcal{M}}_3}[\widehat{W}, \widehat{W}, \widehat{W}]$.
7. Let $\widehat{M}_3[\widehat{W}, \widehat{W}, \widehat{W}] := \underline{\widehat{\mathcal{M}}_3}[\widehat{W}, \widehat{W}, \widehat{W}] - \hat{\sigma}^2 \sum_{i=1}^{d} \big( (\widehat{W}^\top \underline{\hat{\mu}}) \otimes (\widehat{W}^\top e_i) \otimes (\widehat{W}^\top e_i) + (\widehat{W}^\top e_i) \otimes (\widehat{W}^\top \underline{\hat{\mu}}) \otimes (\widehat{W}^\top e_i) + (\widehat{W}^\top e_i) \otimes (\widehat{W}^\top e_i) \otimes (\widehat{W}^\top \underline{\hat{\mu}}) \big)$.

## Part 3

8. Repeat the following steps $t$ times (where $t := \lceil \log_2(1/\delta) \rceil$ for confidence $1 - \delta$):
   (a) Choose $\theta \in \mathbb{R}^k$ uniformly at random from the unit sphere in $\mathbb{R}^k$.
   (b) Let $\{(\hat{v}_i, \hat{\lambda}_i) : i \in [k]\}$ be the eigenvector/eigenvalue pairs of $\widehat{M}_3[\widehat{W}, \widehat{W}, \widehat{W}\theta]$.

   Retain the results for which $\min\big(\{|\hat{\lambda}_i - \hat{\lambda}_j| : i \neq j\} \cup \{|\hat{\lambda}_i| : i \in [k]\}\big)$ is largest.
9. Return the parameter estimates $\hat{\sigma}^2$,

$$\hat{\mu}_i := \frac{\hat{\lambda}_i}{\theta^\top \hat{v}_i} \widehat{B} \hat{v}_i, \quad i \in [k],$$

$$\hat{w} := [\hat{\mu}_1 | \hat{\mu}_2 | \cdots | \hat{\mu}_k]^\dagger \hat{\mu}.$$

# LearnGMM Sample Complexity

THE UNIVERSITY OF BRITISH COLUMBIA

## Finite Sample Complexity

**Theorem 3** (Finite sample bound). *There exists a polynomial* poly$(\cdot)$ *such that the following holds. Let $M_2$ be the matrix defined in Theorem 2, and $\varsigma_t[M_2]$ be its $t$-th largest singular value (for $t \in [k]$). Let $b_{\max} := \max_{i\in[k]} \|\mu_i\|_2$ and $w_{\min} := \min_{i\in[k]} w_i$. Pick any $\varepsilon, \delta \in (0,1)$. Suppose the sample size $n$ satisfies*

$$n \geq \operatorname{poly}\Big(d, k, 1/\varepsilon, \log(1/\delta), 1/w_{\min}, \varsigma_1[M_2]/\varsigma_k[M_2], b_{\max}^2/\varsigma_k[M_2], \sigma^2/\varsigma_k[M_2], \Big).$$

*Then with probability at least $1-\delta$ over the random sample and the internal randomness of the algorithm, there exists a permutation $\pi$ on $[k]$ such that the $\{\hat{\mu}_i : i \in [k]\}$ returned by* LearnGMM *satisfy*

$$\|\hat{\mu}_{\pi(i)} - \mu_i\|_2 \leq \Big(\|\mu_i\|_2 + \sqrt{\varsigma_1[M_2]}\Big)\varepsilon$$

*for all $i \in [k]$.*

**Is this a nice result? I am not sure, but it relies on the empirical moments converging by CLT at rate of $n^{-1/2}$, which does not seem great.**

# Note On Practical Implementation

THE UNIVERSITY OF BRITISH COLUMBIA

## Recommendation From Authors

Alternatives to LearnGMM used to extract the parameters from estimates of M2 and M3 include:

- Simultaneous diagonalization techniques (Bunse-Gerstner et al., 1993)
- Orthogonal tensor decompositions (Anandkumar et al., 2012a)

These alternative methods are more robust to sampling error.

## ICA Overview

- $h \in \mathbb{R}^k$ be some random vector with independent entries (unobserved signal)
- $h \in \mathbb{R}^k$ be Multivariate Gaussian (noise)
- We observe $x := Ah + z$ for some $A \in \mathbb{R}^{k \times k}$ and h / z are independent
- Given a set of $\{x_i, i = 1, 2, ...m\}$, we want to recover h

**This means that we can use this third order moment matching scheme to solve ICA problems**

## Formal Result

**Theorem 4.** *In the ICA model described above, assume* $\mathbb{E}[h_i] = 0$, $\mathbb{E}[h_i^2] = 1$, *and* $\kappa_i := \mathbb{E}[h_i^4] - 3 \neq 0$ (i.e., *the excess kurtosis is non-zero*), *and that* $A$ *is non-singular. Define* $f \colon \mathbb{R}^k \to \mathbb{R}$ *by*

$$f(\eta) := 12^{-1}\left(m_4(\eta) - 3m_2(\eta)^2\right)$$

*where* $m_p(\eta) := \mathbb{E}[(\eta^\top x)^p]$. *Suppose* $\phi \in \mathbb{R}^k$ *and* $\psi \in \mathbb{R}^k$ *are such that* $\frac{(\phi^\top \mu_1)^2}{(\psi^\top \mu_1)^2}, \frac{(\phi^\top \mu_2)^2}{(\psi^\top \mu_2)^2}, \ldots, \frac{(\phi^\top \mu_k)^2}{(\psi^\top \mu_k)^2} \in \mathbb{R}$ *are distinct. Then the matrix*

$$M_{\mathrm{ICA}}(\phi, \psi) := \left(\nabla^2 f(\phi)\right)\left(\nabla^2 f(\psi)\right)^{-1}$$

*is diagonalizable; the eigenvalues are* $\frac{(\phi^\top \mu_1)^2}{(\psi^\top \mu_1)^2}, \frac{(\phi^\top \mu_2)^2}{(\psi^\top \mu_2)^2}, \ldots, \frac{(\phi^\top \mu_k)^2}{(\psi^\top \mu_k)^2}$ *and each have geometric multiplicity one, and the corresponding eigenvectors are* $\mu_1, \mu_2, \ldots, \mu_k$ *(up to scaling and permutation).*

**Theorem 4 lets us estimate the columns of A up to scaling, which in turn lets us estimate h.**

## What did we talk about today?

- Gaussian Mixture Models (GMM)
- Method of Moments Estimators
- How to use Method of Moments to estimate parameters in GMM
- Sample complexity of such an algorithm
- How it can be extended to ICA like algorithms

**for more recent work in this area see:**

`http://proceedings.mlr.press/v65/li17a/li17a.pdf`

# Sources

## Recommended Reading

- https://www.cs.ubc.ca/~jnutini/documents/mlrg_pca.pdf
- https://www.cs.columbia.edu/~djhsu/papers/mog-slides.pdf
- https://arxiv.org/pdf/1206.5766.pdf
- https://www.cse.wustl.edu/~bjuba/cse519t/f19/papers/Li17a.pdf