Learning Diagonal Gaussian Mixture Models and Incomplete Tensor Decomposition

Jiawang Nie

Joint with Bingni Guo and Zi Yang
University of California, San Diego

IPAM Tensor Program 2021, Workshop III:
Mathematical Foundations and Algorithms for Tensor Computations

Gaussian Mixture Model (GMM)

A Gaussian mixture random variable $y \in \mathbb{R}^d$ is such that

$$y = \begin{cases} z_1 \sim \mathcal{N}(\mu_1, \Sigma_1) & \text{with prob.} \quad \omega_1 \\ \vdots & \vdots \\ z_r \sim \mathcal{N}(\mu_r, \Sigma_r) & \text{with prob.} \quad \omega_r \end{cases}$$

with $\omega_1, \ldots, \omega_r > 0$, $\sum \omega_i = 1$. The density function is

$$f(y) := \sum_{i=1}^{r} \frac{\omega_i}{\sqrt{(2\pi)^d \det \Sigma_i}} \exp\Big\{ -\frac{1}{2}(y-\mu_i)^T \Sigma_i^{-1} (y-\mu_i) \Big\}.$$

The task of GMM is to learn Gaussian parameters

$$\mu_i \in \mathbb{R}^d, \quad \Sigma_i \in \mathbb{R}^{d\times d}, \quad \omega_i \in \mathbb{R}_+.$$

General approaches: estimate them from samples $y_1, \cdots, y_N$ of y.

Backgrounds and Related Work

Broad applications: automatic speech recognition, hyperspectral unmixing problem, background subtraction, anomaly detection.

- The expectation-maximization (EM) algorithm (Dempster, ...), based on the maximum likelihood estimation.
- Projection into low-dimensional subspace (Dasgupta, Arora-Kannan, Vempala and Wong, ...)
- Moment based methods (Pearson, Beklkin-sinha, Anandkumar-Ge-Hsu-Kahade, ...)
  - Spherical covariance Σ_i

This talk: Incomplete tensor decompositions based on moments.

Moments for GMMs

The first order moment vector is

$$M_1 := \mathbb{E}[y] = \omega_1 \mu_1 + \cdots + \omega_r \mu_r.$$

The second order moment matrix is

$$M_2 := \mathbb{E}[y \otimes y] = \sum_{i=1}^{r} \omega_i \Big(\mu_i \otimes \mu_i + \Sigma_i \Big).$$

The third order moment tensor is

$$M_3 := \mathbb{E}[y \otimes y \otimes y] = \sum_{i=1}^{r} \omega_i \mu_i \otimes \mu_i \otimes \mu_i + \cdots$$

The remaining terms are quite complicated.....

Moments for Diagonal GMMs

Assume every covariance matrix Σ_i is diagonal, i.e.,

$$\Sigma_i = \begin{bmatrix} \sigma_{i1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{i2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{id}^2 \end{bmatrix}.$$

Then one can show that

$$M_3 := \mathbb{E}[y \otimes y \otimes y] = \sum_{i=1}^{r} \omega_i \mu_i \otimes \mu_i \otimes \mu_i + \sum_{j=1}^{d} \Big(a_j \otimes e_j \otimes e_j + e_j \otimes a_j \otimes e_j + e_j \otimes e_j \otimes a_j \Big),$$

where $a_j := \sum_{i=1}^{r} \omega_i \sigma_{ij}^2 \mu_i \in \mathbb{R}^d$ for $j = 1, \cdots, d$.

The Incomplete Moment Tensor

When each Σ_i is diagonal, the cubic moment tensor is

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

For the tensor $\mathcal{F} := \sum_{i=1}^{r} \omega_i \mu_i \otimes \mu_i \otimes \mu_i$, it holds that

$$(M_3)_{i_1 i_2 i_3} = (\mathcal{F})_{i_1 i_2 i_3} \quad \text{for all} \quad i_1 \neq i_2 \neq i_3 \neq i_1.$$

Denote the label set

$$\Omega = \{(i_1, i_2, i_3) : i_1 \neq i_2 \neq i_3 \neq i_1,\ i_1, i_2, i_3 \in [d]\}.$$

The subtensor $\mathcal{F}|_\Omega = M_3|_\Omega$ is available from samplings of y.

Incomplete Tensor Decomposition

Denote the label set

$$\Omega = \{(i_1, i_2, i_3) : i_1 \neq i_2 \neq i_3 \neq i_1\}.$$

The subtensor $\mathcal{F}|_\Omega = M_3|_\Omega$ is available from sampling.

The computational task: for given subtensor $\mathcal{F}|_\Omega$, we look for an incomplete tensor decomposition (iTD)

$$\mathcal{F}_{i_1 i_2 i_3} = \left(v_1^{\otimes 3} + \cdots + v_r^{\otimes 3}\right)_{i_1 i_2 i_3}, \quad \text{for} \quad (i_1, i_2, i_3) \in \Omega,$$

$$v_1, \ldots, v_r \in \mathbb{R}^d.$$

Do Tensor Completion?

For given subtensor $\mathcal{F}|_\Omega$, we look for an iTD

$$\mathcal{F}_{i_1 i_2 i_3} = \left(v_1^{\otimes 3} + \cdots + v_r^{\otimes 3} \right)_{i_1 i_2 i_3}, \quad (i_1, i_2, i_3) \in \Omega.$$

One may do tensor completion: find missing tensor entries

$$\mathcal{F}_{i_1 i_2 i_3}, \quad (i_1, i_2, i_3) \notin \Omega$$

and then compute an TD for $\mathcal{F}$.

To learn GMMs, tensor completion is discouraged:

- The theory for tensor completion or recovery is premature.
- Convex relaxations (e.g., the nuclear norm or trace min) are applicable. The computation is still quite expensive, while completed tensors are typically not low rank.
- Low performance in numerical experiments.

Our Proposed Method

For the given subtensor $\mathcal{F}|_\Omega$, we compute an iTD

$$\mathcal{F}_{i_1 i_2 i_3} = \left(v_1^{\otimes 3} + \cdots + v_r^{\otimes 3} \right)_{i_1 i_2 i_3}, \quad (i_1, i_2, i_3) \in \Omega.$$

The method is based on *generating polynomials* (GPs) in the work

> *Jiawang Nie,* **Generating polynomials and symmetric tensor decompositions**, *Found. of Comp. Math. 17 (2), 423-465.*

It works efficiently when

- The $\mu_1, \cdots, \mu_k$ are linearly independent;
- $r \leq \frac{d-2}{2}$.

These assumptions are sufficient (may not necessary) for the success of the method.

Notation

- Let $n := d - 1$.
- The $\mathrm{S}^3(\mathbb{C}^d)$ denotes the space of all complex $d \times d \times d$ symmetric tensors. Label $\mathcal{F}$ as

$$\mathcal{F} = (\mathcal{F}_{i_1 \ldots i_m}), \quad 0 \leq i_1, \ldots, i_m \leq n.$$

- Define the bilinear functional ($x_0 := 1$)

$$\langle p, \mathcal{F} \rangle := \sum_{0 \leq i_1, i_2, i_3 \leq n} p_{i_1 i_2 i_3} \mathcal{F}_{i_1 i_2 i_3}$$

for

$$p = \sum_{0 \leq i_1, i_2, i_3 \leq n} p_{i_1 i_2 i_3} \cdot x_{i_1} x_{i_2} x_{i_3}.$$

Generating Polynomials (GPs)

A poly $g \in \mathbb{C}[x]_3$ is a GP for $\mathcal{F} \in \mathrm{S}^3(\mathbb{C}^{n+1})$ if

$$\langle g \cdot x^\beta, \mathcal{F} \rangle = 0 \quad \forall \beta : \deg(g \cdot x^\beta) \leq 3.$$

Denote $\mathbb{B}_0 := \{x_1, \cdots, x_r\}$ and

$$\mathbb{B}_1 := x_{r+1}\mathbb{B}_0 \cup \cdots \cup x_n\mathbb{B}_0 \backslash \mathbb{B}_0 = \{x_i x_j : 1 \leq i \leq r,\ r+1 \leq j \leq n\}.$$

For each $x_i x_j \in \mathbb{B}_1$ and $G \in \mathbb{C}^{\mathbb{B}_0 \times \mathbb{B}_1}$, denote

$$\varphi_{ij}[G](x) := \sum_{k=1}^{r} G(k, e_i + e_j) x_k - x_i x_j.$$

Want G such that each $\varphi_{ij}[G]$ is a GP, i.e.,

$$\langle x_t \varphi_{ij}[G](x), \mathcal{F} \rangle = \sum_{k=1}^{r} G(k, e_i + e_j) \mathcal{F}_{0kt} - \mathcal{F}_{ijt} = 0, \quad t = 0, 1, \ldots, n,$$

for all i, j. Such G is called the *generating matrix*.

Existence and Uniqueness of Generating Polynomials

For $G \in \mathbb{C}^{\mathbb{B}_0 \times \mathbb{B}_1}$, consider the polynomials

$$\varphi_{ij}[G](x) := \sum_{k=1}^{r} G(k, e_i + e_j) x_k - x_i x_j.$$

Theorem (Guo-N.-Yang): For $\mathcal{F} \in \mathrm{S}^3(\mathbb{C}^{n+1})$ with

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

if the subvectors $(u_1)_{1:r}, \ldots, (u_r)_{1:r}$ are linearly independent, then there is a unique $G \in \mathbb{C}^{\mathbb{B}_0 \times \mathbb{B}_1}$ such that each $\varphi_{ij}[G](x)$ is a GP, i.e.,

$$\langle x_t \varphi_{ij}[G](x), \mathcal{F} \rangle = \sum_{k=1}^{r} G(k, e_i + e_j) \mathcal{F}_{0kt} - \mathcal{F}_{ijt} = 0,$$

for all $x_i x_j \in \mathbb{B}_1$ and $t = 0, 1, \ldots, n$.

From GPs to Tensor Decompositions

For $\mathcal{F} = \lambda_1 \begin{pmatrix} 1 \\ u_1 \end{pmatrix}^{\otimes 3} + \cdots + \lambda_r \begin{pmatrix} 1 \\ u_r \end{pmatrix}^{\otimes 3}$, if $\varphi_{ij}[G, \alpha](x)$ is a GP, then for $s = 1, \ldots, r$,

$$\varphi_{ij}[G](u_s) = \sum_{k=1}^{r} G(k, e_i + e_j)(u_s)_k - (u_s)_i (u_s)_j = 0.$$

So $u_1, \ldots, u_r$ are solutions to eigen-equations

$$\underbrace{\begin{pmatrix} G(1, e_1 + e_l) & \cdots & G(r, e_1 + e_l) \\ \vdots & \ddots & \vdots \\ G(1, e_r + e_l) & \cdots & G(r, e_r + e_l) \end{pmatrix}}_{N_l(G)} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_r \end{pmatrix} = x_l \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_r \end{pmatrix}$$

for $l = r + 1, \ldots, n$. Then one can show that

$$N_l(G)(u_i)_{1:r} = (u_i)_l (u_i)_{1:r}.$$

We can obtain TDs from eigen-decompositions of $N_l(G)$.

An Example for GPs

For the tensor $\mathcal{F} \in \mathbb{S}^3(\mathbb{C}^6)$ that is

$$\mathcal{F} = 0.4 \cdot (1,2,3,4,5,6)^{\otimes 3} + 0.6 \cdot (1,-1,2,-1,2,3)^{\otimes 3}.$$

The rank $r = 2$, $\mathbb{B}_0 = \{x_1, x_2\}$ and

$$\mathbb{B}_1 = \{x_1x_3, x_1x_4, x_1x_5, x_2x_3, x_2x_4, x_2x_5\}.$$

One can calculate that

$$N_3 = \begin{pmatrix} G(1, e_1+e_3) & G(2, e_1+e_3) \\ G(1, e_2+e_3) & G(2, e_2+e_3) \end{pmatrix} = \begin{pmatrix} 1/3 & 2/3 \\ 4/3 & -1/3 \end{pmatrix},$$

$$N_4 = \begin{pmatrix} G(1, e_1+e_4) & G(2, e_1+e_4) \\ G(1, e_2+e_4) & G(2, e_2+e_4) \end{pmatrix} = \begin{pmatrix} 4/3 & -1/3 \\ -2/3 & 5/3 \end{pmatrix},$$

$$N_5 = \begin{pmatrix} G(1, e_1+e_5) & G(2, e_1+e_5) \\ G(1, e_2+e_5) & G(2, e_2+e_5) \end{pmatrix} = \begin{pmatrix} 5/3 & -2/3 \\ -4/3 & 7/3 \end{pmatrix}.$$

The u_1, u_2 can be determined by eigen-values-vectors of N_3, N_4, N_5.

An Algorithm for Incomplete Tensor Decomposition

Input: A cubic subtensor $\mathcal{F}|_\Omega$ and the rank $r \leq \frac{d}{2} - 1$.
Step 1: Determine G such that each $\varphi_{ij}[G](x)$ is a GP.
Step 2: Formulate the matrices $N_{r+1}(G), \ldots, N_n(G)$ and let

$$N(\xi) := \xi_{r+1} N_{r+1} + \cdots + \xi_n N_n$$

for a generic weight vector $\xi := (\xi_{r+1}, \cdots, \xi_n)$.
Step 3: Compute unit length eigenvectors $\tilde{v}_1, \ldots, \tilde{v}_r$ of $N(\xi)$ and let

$$\tilde{w}_i = \left(\tilde{v}_i^* N_{r+1}(G)\tilde{v}_i, \cdots, \tilde{v}_i^* N_n(G)\tilde{v}_i\right), \quad i = 1, \ldots, r.$$

Step 4: Determine scalars γ_i, λ_i such that (linear least squares)

$$\mathcal{F} = \lambda_1(1, \gamma_1\tilde{v}_1, \tilde{w}_1)^{\otimes 3} + \cdots + \lambda_r(1, \gamma_r\tilde{v}_r, \tilde{w}_r)^{\otimes 3}.$$

Output: $\mathcal{F} = \lambda_1(1, u_1)^{\otimes 3} + \cdots + \lambda_r(1, u_r)^{\otimes 3}$.

Robustness with noises

If the subtensor $\mathcal{F}_\Omega$ has noises, we can still apply the same method to get a rank-r tensor approximation:

Linear equations $\rightarrow$ *Linear Least Squares.*

Theorem (Guo-N.-Yang)

If the subtensor $\mathcal{F}_\Omega$ is sufficiently close to a low rank one, then the GP method can produce a quasi-optimal tensor decomposition.

Reason: Linear Least squares and Eigenvalue decompositions are numerically stable, under general assumptions.

The Choice of Rank r

Let $\text{Flat}(\mathcal{F})$ be the flattening matrix of $\mathcal{F}$ such that

$$\text{Flat}(\mathcal{F})_{i,(j,k)} = \mathcal{F}_{i,j,k}$$

for all $i, j, k = 0, 1, \ldots, n$. It can be shown that

$$\text{rank}(\text{Flat}(\mathcal{F})) = \text{rank}(\mathcal{F})$$

when the decomposing vectors of $\mathcal{F}$ are generic and $r \leq d$.

The rank of $\text{Flat}(\mathcal{F})$ is not available since only the subtensor $(\mathcal{F})_\Omega$ is known.

However, we can calculate the ranks of submatrices of $(\mathcal{F})_\Omega$ whose entries are all known.

The Choice of Rank r (continued)

To see this, consider $\mathcal{F} \in \mathrm{S}^3(\mathbb{C}^7)$. The matrix $\mathrm{Flat}(\mathcal{F})$ is

$$\begin{bmatrix} * & * & * & * & * & * & * \\ * & * & \mathcal{F}_{120} & \mathcal{F}_{130} & \mathcal{F}_{140} & \mathcal{F}_{150} & \mathcal{F}_{160} \\ * & \mathcal{F}_{210} & * & \mathcal{F}_{230} & \mathcal{F}_{240} & \mathcal{F}_{250} & \mathcal{F}_{260} \\ * & \mathcal{F}_{310} & \mathcal{F}_{320} & * & \mathcal{F}_{340} & \mathcal{F}_{350} & \mathcal{F}_{360} \\ * & \mathcal{F}_{410} & \mathcal{F}_{420} & \mathcal{F}_{430} & * & \mathcal{F}_{450} & \mathcal{F}_{460} \\ * & \mathcal{F}_{510} & \mathcal{F}_{520} & \mathcal{F}_{530} & \mathcal{F}_{540} & * & \mathcal{F}_{560} \\ * & \mathcal{F}_{610} & \mathcal{F}_{620} & \mathcal{F}_{630} & \mathcal{F}_{640} & \mathcal{F}_{650} & * \end{bmatrix},$$

where $*$ means *unknown*. The submatrices with all entries known are

$$\begin{bmatrix} \mathcal{F}_{410} & \mathcal{F}_{420} & \mathcal{F}_{430} \\ \mathcal{F}_{510} & \mathcal{F}_{520} & \mathcal{F}_{530} \\ \mathcal{F}_{610} & \mathcal{F}_{620} & \mathcal{F}_{630} \end{bmatrix}, \quad \begin{bmatrix} \mathcal{F}_{140} & \mathcal{F}_{150} & \mathcal{F}_{160} \\ \mathcal{F}_{240} & \mathcal{F}_{250} & \mathcal{F}_{260} \\ \mathcal{F}_{340} & \mathcal{F}_{350} & \mathcal{F}_{360} \end{bmatrix}.$$

Use the biggest rank of them to estimate r (works if $r \leq \frac{d}{2} - 1$).

How to learn GMM parameters?

Let y be the GMM random variable such that

$$y = z_i \sim \mathcal{N}(\mu_i, \Sigma_i), \quad \text{with prob.} \quad \omega_i.$$

Assume that $r \leq \frac{d}{2} - 1$. For samples $y_1, \ldots, y_N$ of y,

$$\widehat{M}_1 := \frac{1}{N} \sum_{i=1}^{N} y_i \quad \approx \quad M_1 := \mathbb{E}[y] = \sum_{i=1}^{N} \omega_i \mu_i,$$

$$\widehat{M}_3 := \frac{1}{N} \sum_{i=1}^{N} y_i^{\otimes 3} \quad \approx \quad M_3 := \mathbb{E}[y^{\otimes 3}].$$

For $\mathcal{F} = \omega_1 \mu_1^{\otimes 3} + \cdots + \omega_r \mu_r^{\otimes 3}$, if all Σ_i are diagonal, then

$$M_3|_\Omega = \mathcal{F}|_\Omega.$$

Determine ω_i and μ_i

First, compute the tensor approximation

$$\mathcal{F} \approx (p_1)^{\otimes 3} + \cdots + (p_r)^{\otimes 3}.$$

Second, determine scalars β_i such that

$$\widehat{M}_1 \approx \beta_1 p_1 + \cdots + \beta_r p_r.$$

Third, for each $i = 1, \ldots, r$, let

$$\omega_i := \beta_i^{3/2}, \quad \mu_i = \beta_i^{-1/2} p_i.$$

Then we get

$$M_1 \approx \sum_{i=1}^{r} \omega_i \mu_i, \quad M_3 \approx \sum_{i=1}^{r} \omega_i \mu_i^{\otimes 3}.$$

If necessary, apply nonlinear optimization to improve the accuracy.

Determine Covariance Matrices Σ_i

Observe that

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

where $a_j = \sum_{i=1}^{r} \omega_i \sigma_{ij}^2 \mu_i$ for $j = 1, \cdots, d$. The vectors a_j can be estimated as

$$(a_j)_j = \frac{1}{3}\widetilde{\mathcal{A}}_{jjj}, \quad (a_j)_i = \frac{1}{3}\widetilde{\mathcal{A}}_{ijj} \quad \text{for} \quad i \neq j.$$

Determine diagonals σ_{ij}^2 from the linear least squares

$$\min_{\sigma_{1j}^2, \ldots, \sigma_{rj}^2} \quad \left\| a_j - \sum_{i=1}^{r} \sigma_{ij}^2 \cdot (\omega_i \mu_i) \right\|_2^2, \quad j = 1, \cdots, d.$$

If necessary, apply nonlinear optimization to improve the accuracy.

Robustness of the Method

Due to sampling errors,

$$\widehat{M}_1 := \frac{1}{N}\sum_{i=1}^{N} y_i \quad \approx \quad M_1 := \mathbb{E}[y] = \sum_{i=1}^{N} \omega_i \mu_i,$$

$$\widehat{M}_3 := \frac{1}{N}\sum_{i=1}^{N} y_i^{\otimes 3} \quad \approx \quad M_3 := \mathbb{E}[y^{\otimes 3}].$$

Theorem (Guo-N.-Yang)

Consider the GMM with Gaussian parameters $(\omega_i, \mu_i, \Sigma_i)$, $i = 1, \ldots, r \leq \frac{d}{2} - 1$. *If* $\epsilon = \max(\|M_3 - \widehat{M}_3\|, \|M_1 - \widehat{M}_1\|)$ *is small enough, then*

$$\|\mu_i - \mu_i^{comp}\| = O(\epsilon),\ \|\omega_i - \omega_i^{comp}\| = O(\epsilon),\ \|\Sigma_i - \Sigma_i^{comp}\| = O(\epsilon).$$

Relative Errors for iTD

Generate random tensors $\mathcal{F} = (p_1)^{\otimes 3} + \cdots + (p_r)^{\otimes 3}$. Apply random perturbations $(\widehat{\mathcal{F}})_\Omega = (\mathcal{F})_\Omega + \mathcal{E}_\Omega$. Then apply the Algorithm to get the rank-r tensor $\widehat{\mathcal{F}}$. The accuracy is measured by

$$\text{rel-error} := \frac{\|(\mathcal{F}^* - \widehat{\mathcal{F}})_\Omega\|}{\|(\mathcal{F} - \widehat{\mathcal{F}})_\Omega\|}.$$

			rel-error						rel-error		
d	r	ϵ	min	average	max	d	r	ϵ	min	average	max
20	3	0.1	0.9610	0.9731	0.9835	30	4	0.1	0.9816	0.9854	0.9890
	5	0.01	0.9634	0.9700	0.9742		8	0.01	0.9634	0.9700	0.9742
	7	0.001	0.9148	0.9373	0.9525		11	0.001	0.9501	0.9587	0.9667
40	6	0.1	0.9853	0.9877	0.9904	50	7	0.1	0.9887	0.9911	0.9925
	10	0.01	0.9761	0.9795	0.9820		13	0.01	0.9812	0.9831	0.9854
	15	0.001	0.9653	0.9690	0.9734		18	0.001	0.9739	0.9767	0.9792

Performance for Diagonal GMMs

Generate 100 random instances of $\{(\omega_i, \mu_i, \Sigma_i) : i = 1, \cdots, r\}$ for $d \in \{20, 30, 40\}$, and 20 random instances for $d \in \{50, 60\}$. For each case, 10000 samples are generated, from one of r component Gaussian distributions. The Incomplete tensor decomposition (iTD) and EM are applied. As the difference $d - r$ increases, the iTD becomes more accurate, opposite to EM.

Accuracy	$d = 20$			$d = 30$			$d = 40$		
	$r = 3$	$r = 5$	$r = 7$	$r = 4$	$r = 8$	$r = 11$	$r = 6$	$r = 10$	$r = 15$
iTD	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	$d = 50$			$d = 60$		
	$r = 7$	$r = 13$	$r = 18$	$r = 8$	$r = 15$	$r = 22$
iTD	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

Apply iTD to do Texture Classifications

We select 8 textured images of 512×512 pixels from VisTex, converted into grayscale version. For each image, apply iTD and EM to fit a GMM to the image. Then compare the accuracy with the EM method.

Table: Classification Accuracy Results on 8 textures

Accuracy	iTD	EM
Bark.0000	0.5376	0.8413
Bark.0009	0.5107	0.7150
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

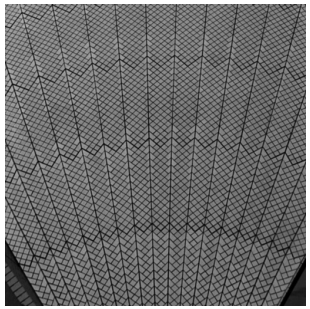
Pictures of Selected Images

Figure: Textures from VisTex

Bark.0000

Bark.0009

Flowers.0001

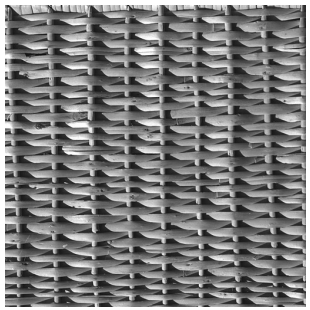


Tile.0000

Ptings.11.0001

Grass.0001

Brick.0004



Fabric.0013

Conclusions

- Learn Gaussian mixture models with diagonal covariance matrices.
- Reduced to incomplete tensor decomposition (iTD).
- An algorithm, based on generating polynomials, is proposed to compute iTD.
- Stability analysis is given. If the incomplete tensor is accurate enough, the output iTD is quasi-optimal.
- Numerical experiments are provided.

Thank You Very Much!