Learning General Gaussian Mixture Model with Integral Cosine Similarity

Guanglin Li[1], **Bin Li**[1,2] *, **Changsheng Chen**[1,2], **Shunquan Tan**[1,2], **Guoping Qiu**[1,2]

[1]Guangdong Key Laboratory of Intelligent Information Processing and Shenzhen Key Laboratory of Media Security, Shenzhen University, Shenzhen 518060, China.

[2]Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen 518129, China

liguanglin@email.szu.edu.cn, {libin, cschen, tansq, qiu}@szu.edu.cn

Abstract

Gaussian mixture model (GMM) is a powerful statistical tool in data modeling, especially for unsupervised learning tasks. Traditional learning methods for GMM such as expectation maximization (EM) require the covariance of the Gaussian components to be non-singular, a condition that may not often satisfied in real-world applications. This paper presents a new learning method called G^{2}M^2 (General Gaussian Mixture Model) by fitting an unnormalized Gaussian mixture function (UGMF) to a data distribution. At the core of G^{2}M^2 is the introduction of an integral cosine similarity (ICS) function for comparing the UGMF and the unknown data density distribution without having to explicitly estimate it. By maximizing the ICS through Monte Carlo sampling, the UGMF can be made to overlap with the unknown data density distribution such that the two only differ by a constant scalar, and the UGMF can be normalized to obtain the data density distribution. A Siamese convolutional neural network is also designed for optimizing the ICS function. Experimental results show that our method is more competitive in modeling data having correlations that may lead to singular covariance matrices in GMM, and it outperforms state-of-the-art methods in unsupervised anomaly detection.

1 Introduction

Gaussian Mixture Model (GMM) is a parametric probability density function consisting of a weighted sum of Gaussian distribution functions [Sorenson and Alspach, 1971]. It can be applied to approximate any complicated density defined on $\mathbb{R}^d$ with a sufficiently large number of mixture components [Kolouri *et al.*, 2018]. GMM is a popular and powerful tool for a wide range of applications, such as pattern classification [Duda *et al.*, 1973], statistical modeling [Zoran *et al.*, 2012], unsupervised anomaly detection [Qu *et al.*, 2020], etc.

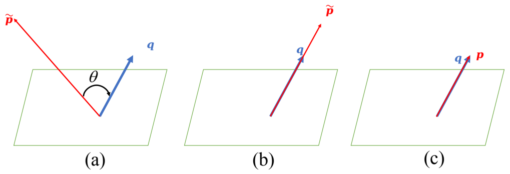


Figure 1: Schematics illustration of the idea of G^{2}M^2. (a) Start from an unnormalized Gaussian mixture function $\tilde{\boldsymbol{p}}$ and the probability density distribution of the data $\boldsymbol{q}$. (b) Move $\tilde{\boldsymbol{p}}$ towards $\boldsymbol{q}$ until $\tilde{\boldsymbol{p}}$ overlap with $\boldsymbol{q}$ such that they only differ by a constant. (c) Scale $\tilde{\boldsymbol{p}}$ by a constant to obtain $\boldsymbol{p}$, which will be the same as $\boldsymbol{q}$, the probability density distribution of the data.

Expectation Maximum (EM) algorithm is the most widely used method to estimate a GMM model [Dempster *et al.*, 1977; Ng, 2000]. However, in many practical applications, the covariance matrices of some Gaussian components of the mixture model are singular (referred to as *the singularity problem* in the remaining text) due to insufficient data or data correlation. As the inverse of a singular matrix does not exist and the normalization term of the Gaussian distribution function is infinite, this will cause the EM algorithm to fail. A variety of methods have been proposed to overcome this difficulty. One kind of method uses data transformation as a preprocessing step and then performs modeling in the transformed space. PCA (principal component analysis) and feature selection methods are representatives of this category [Xu *et al.*, 2014; Venegas *et al.*, 2019]. Another kind of method, i.e., regularization, has been proposed to handle the problem by modifying the objective function or constraining the structure of covariance matrix. For instance, small values on the diagonal entries of the covariance matrix is penalized [Zong *et al.*, 2018], or a positive quantity is added to the diagonal of the covariance matrix [Do and Ohsaki, 2021; Davari *et al.*, 2018; Cho *et al.*, 2019]. Another way is to restrict the covariance matrix to be diagonal [Ban *et al.*, 2018; Ma *et al.*, 2019; Yang *et al.*, 2019]. Although these methods overcome the singularity problem, they may lead to the loss of information or limit the powerful representation capability of GMM [Zong *et al.*, 2018; Zhou *et al.*, 2020].

This paper proposes a new learning method called G^{2}M^2 (General Gaussian Mixture Model), i.e., being capable of working in general conditions even when the covariance matrices are singular. The idea is straightforward. Starting from an unnormalized Gaussian mixture function (UGMF) without

*Corresponding author

a normalization term compared to GMM, we move the function towards a target but unknown data density distribution until the function and the data density distribution are at the same orientation, but may only differ by a scaling constant. Normalizing the UGMF by making its integration to 1, the unnormalized function then becomes the data density distribution we want to estimate. The key to realize this idea is the computation of the similarity between the UGMF and the unknown target data density distribution. We tackle this by introducing an integral cosine similarity (ICS) function and using Monte Carlo (MC) sampling to compute the cosine similarity without the need to explicitly estimate the data density distribution. The schematic illustration of the idea of G^{2}M^2 is shown in Fig. 1 and summarized as follows.

- Step 1 (Fig. 1(a)): starting from an UGMF $\tilde{\boldsymbol{p}}$ (see Eq. (3)) and assuming the unknown target probability density distribution of the data is $\boldsymbol{q}$, we resort to the angle between $\tilde{\boldsymbol{p}}$ and $\boldsymbol{q}$, i.e., θ. We introduce the ICS to compute $cos^2(\theta)$ through MC sampling without the need to explicitly estimating $\boldsymbol{q}$ (see Section 3.2).
- Step 2 (Fig 1(b)): move $\tilde{\boldsymbol{p}}$ towards $\boldsymbol{q}$ to maximize $cos^2(\theta)$. When $cos^2(\theta) = 1$ or $\theta = 0$, $\tilde{\boldsymbol{p}}$ and $\boldsymbol{q}$ will overlap and only differ by a scaling constant.
- Step 3 (Fig 1(c)): scale $\tilde{\boldsymbol{p}}$ by a constant such that its integration is 1, and we obtain $\boldsymbol{p}$, the density distribution of the data which is in the form of a Gaussian mixture (see Section 3.3).

The contributions of this paper are summarized as follows.

- A learning method called G^{2}M^2 (general Gaussian mixture model) is proposed. It can fit an unnormalized Gaussian mixture function (UGMF) to data distribution thus overcoming the limitations associated with traditional GMM learning methods when some covariance matricies of the estimated Gaussian are singular.
- An integral cosine similarity (ICS) metric is introduced. It overcomes the limitation of Kullback-Leibler divergence or Jensen-Shannon divergence that can only evaluate the similarly between two probability density functions.
- A Siamese neural network (NN) architecture is constructed to optimize an ICS-based optimization function for learning the parameters of UGMF. With Monte Carlo sampling, the parameters of UGMF can be estimated in an elegant way.
- Experimental results show that the proposed solution outperforms conventional EM-based methods in estimating the probability density functions of complex data and achieves state-of-the-art performance in the application of unsupervised anomaly detection.

2 Preliminary and Related Work

Fundamentals of GMM. A GMM is defined as

$$p(\boldsymbol{x}|\boldsymbol{\lambda}) = \sum_{i=1}^{M} \omega_i g(\boldsymbol{x}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}), \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is a d-dimensional real-valued vector; $\boldsymbol{\lambda} = \{\omega_i, \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}(i = 1, \cdots, M)\}$ is a set of parameters characterizing the GMM; ω_i $(i = 1, \cdots, M)$ is the mixture weight with the constraint $\sum_{i=1}^{M} \omega_i = 1(\omega_i > 0)$; and $g(\boldsymbol{x}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})$ $(i = 1, \cdots, M)$ is a multivariate Gaussian density component parametrized by a mean vector $\boldsymbol{\mu_i}$ and a covariance matrix $\boldsymbol{\Sigma_i}$, which is given as:

$$g(\boldsymbol{x}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma_i}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_i})^T \boldsymbol{\Sigma_i}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_i})}. \tag{2}$$

Note that if the matrix $\boldsymbol{\Sigma_i}$ is singular in (2), the normalization term $\frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma_i}|^{\frac{1}{2}}} = \frac{1}{0}$ becomes infinite and $\boldsymbol{\Sigma_i}^{-1}$ does not exist. This happens due to insufficient data or data having correlations, which will result in the failure of EM.

Estimation of GMM parameters with EM. The EM algorithm [Dempster *et al.*, 1977; Moon, 1996] is effective in estimating the GMM parameters. However, the algorithm suffers from some drawbacks, including the failure to work when the covariance matrix is singular, and being sensitive to noise and the initialization of parameters [McLachlan and Krishnan, 1997; Xu and Wunsch, 2005; Rousseeuw and Hubert, 2011]. We will show that these drawbacks can be overcome by our method.

Regularized GMM methods. The most popular way to solve the singularity problem is by adding a positive value to the diagonal of the covariance matrix [Do and Ohsaki, 2021; Davari *et al.*, 2018; Cho *et al.*, 2019]. However, this is equivalent to adding noise to the covariance matrix and reduces the estimation accuracy. Some other methods assume data features are independent of each other and thus setting the covariance matrices to be diagonal [Ban *et al.*, 2018; Ma *et al.*, 2019; Yang *et al.*, 2019]. However, such a regularization strategy overlooks correlation in data, which is nonnegligible in many practical applications.

3 Proposed Method

3.1 Unnormalized Gaussian Mixture Function

A UGMF is defined as

$$\tilde{p}(\boldsymbol{x}|\boldsymbol{\eta}) = \sum_{i=1}^{M} \xi_i \tilde{g}(\boldsymbol{x}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}^{-1}),\ (\xi_i > 0), \tag{3}$$

where $\boldsymbol{\eta} = \{\xi_i, \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}^{-1}(i = 1, \cdots, M)\}$ is a set of parameters characterizing the UGMF, and $\tilde{g}(\boldsymbol{x}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}^{-1})$ is a Gaussian function which removes the normalization term $\frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma_i}|^{\frac{1}{2}}}$ as:

$$\tilde{g}(\boldsymbol{x}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}^{-1}) = e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_i})^T \boldsymbol{\Sigma_i}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_i})}.$$

Note that there will be no infinite term in $\tilde{p}(\boldsymbol{x}|\boldsymbol{\eta})$ even when the covariance matrix is singular, and we resort to direct estimation of $\boldsymbol{\Sigma_i}^{-1}$.

3.2 Integral Cosine Similarity

Let $q(\boldsymbol{x})$ be the ground truth probability density function of data. Let $\boldsymbol{Q} = [q(\boldsymbol{x}_1), q(\boldsymbol{x}_2), \cdots, q(\boldsymbol{x}_N)]$ and $\tilde{\boldsymbol{P}} = [\tilde{p}(\boldsymbol{x}_1|\boldsymbol{\eta}), \tilde{p}(\boldsymbol{x}_2|\boldsymbol{\eta}), \cdots, \tilde{p}(\boldsymbol{x}_N|\boldsymbol{\eta})]$ be two vectors, where $\boldsymbol{x}_i$ $(i \in \{1, 2, \cdots, N\})$ are uniformly sampled from $\Omega(\mathbb{R}^d)$. As

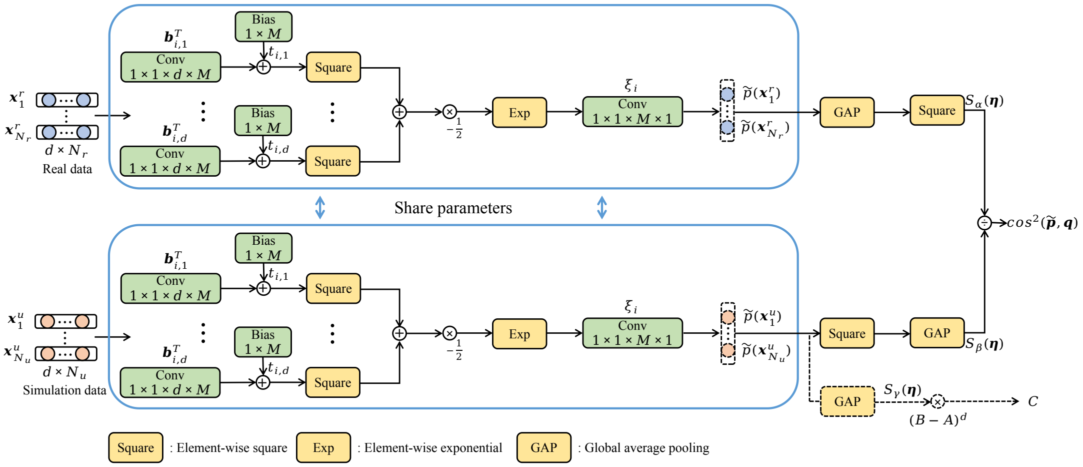


Figure 2: The Siamese neural network structure for learning real data distribution with G^{2}M^2. The two identical networks inside the blue boxes share parameters (weights) and all calculations are implemented based on standard convolutional neural network operations. Simulation data is randomly sampled from a d-dimensional uniform distribution $U(A, B)$. The optimization of parameters is achieved by training the network based on the objective function in (11) using a standard back-propagation algorithm.

there are infinite samples ($N \to \infty$) , the squared cosine similarity of two functions $\tilde{\boldsymbol{p}}$ and $\boldsymbol{q}$ is defined as:

$$\begin{aligned}\cos^2(\tilde{\boldsymbol{p}}, \boldsymbol{q}) &= \lim_{N\to\infty}\left(\frac{\tilde{\boldsymbol{P}}^T\boldsymbol{Q}}{||\tilde{\boldsymbol{P}}||_2||\boldsymbol{Q}||_2}\right)^2 \\ &= \lim_{N\to\infty}\frac{(\sum_{i=1}^N \tilde{p}(\boldsymbol{x}_i|\boldsymbol{\eta})q(\boldsymbol{x}_i))^2}{\sum_{i=1}^N \tilde{p}(\boldsymbol{x}_i|\boldsymbol{\eta})^2\sum_{i=1}^N q(\boldsymbol{x}_i)^2} \\ &= \lim_{N\to\infty}\frac{(\frac{1}{N}\sum_{i=1}^N \tilde{p}(\boldsymbol{x}_i|\boldsymbol{\eta})q(\boldsymbol{x}_i))^2}{\frac{1}{N}\sum_{i=1}^N \tilde{p}(\boldsymbol{x}_i|\boldsymbol{\eta})^2\frac{1}{N}\sum_{i=1}^N q(\boldsymbol{x}_i)^2} \\ &= \frac{(\int_\Omega \tilde{p}(\boldsymbol{x}|\boldsymbol{\eta})q(\boldsymbol{x})\,d\boldsymbol{x})^2}{\int_\Omega \tilde{p}^2(\boldsymbol{x}|\boldsymbol{\eta})\,d\boldsymbol{x}\int_\Omega q^2(\boldsymbol{x})\,d\boldsymbol{x}}\end{aligned} \tag{4}$$

Note that unlike Kullback-Leibler divergence or Jensen-Shannon divergence that can only evaluate the similarly between two probability density functions, Eq. (4) can evaluate the similarity between two arbitrary functions.

Define integrals $I_{\tilde{p},q} = \int_\Omega \tilde{p}(\boldsymbol{x}|\boldsymbol{\eta})q(\boldsymbol{x})\,d\boldsymbol{x}$, $I_{\tilde{p}^2} = \int_\Omega \tilde{p}^2(\boldsymbol{x}|\boldsymbol{\eta})\,d\boldsymbol{x}$, and $I_{q^2} = \int_\Omega q^2(\boldsymbol{x})\,d\boldsymbol{x}$. With MC sampling, $I_{\tilde{p},q}$ can be approximated by the average of the outputs of the function $\tilde{p}(\boldsymbol{x}|\boldsymbol{\eta})$ with samples following the distribution of $q(\boldsymbol{x})$. Consequently, given the data $\boldsymbol{x}_i^r$ $(i \in \{1, 2, \cdots, N_r\})$ which are drawn from real distribution $q(\boldsymbol{x})$, we have

$$(I_{\tilde{p},q})^2 \approx \left(\frac{1}{N_r}\sum_{i=1}^{N_r}\tilde{p}(\boldsymbol{x}_i^r|\boldsymbol{\eta})\right)^2 \doteq S_\alpha(\boldsymbol{\eta}). \tag{5}$$

$I_{\tilde{p}^2}$ can be computed similarly. Let $\boldsymbol{x}_i^u$ $(i \in \{1, 2, \cdots, N_u\})$ be random samples drawn from a uniform distribution defined on $\Omega(\mathbb{R}^d)$. For simplicity and without loss of generality, we assume each dimension in Ω is bounded, and the minimum and the maximum value among all dimensions are A and B, respectively. We adopt the MC sampling method for computing integral by inputting the samples $\boldsymbol{x}_i^u$ into $\tilde{p}(\boldsymbol{x}|\boldsymbol{\eta})$:

$$\begin{aligned}I_{\tilde{p}^2} &= (B-A)^d\int_\Omega \tilde{p}^2(\boldsymbol{x}|\boldsymbol{\eta})\frac{1}{(B-A)^d}\,d\boldsymbol{x} \\ &\approx (B-A)^d\frac{1}{N_u}\sum_{i=1}^{N_u}\tilde{p}^2(\boldsymbol{x}_i^u|\boldsymbol{\eta}) \doteq (B-A)^d S_\beta(\boldsymbol{\eta}).\end{aligned} \tag{6}$$

In other words, $I_{\tilde{p}^2}$ can be approximated by the average of the squared outputs of $\tilde{p}(\boldsymbol{x}|\boldsymbol{\eta})$ with a multiplicative constant $(B-A)^d$. In this way, the cosine similarity can be obtained:

$$cos^2(\tilde{\boldsymbol{p}}, \boldsymbol{q}) = \frac{I_{\tilde{p},\tilde{q}}^2}{I_{p^2}I_{q^2}} = (B-A)^d I_{q^2}\frac{S_\alpha(\boldsymbol{\eta})}{S_\beta(\boldsymbol{\eta})}. \tag{7}$$

As $q(\boldsymbol{x})$ and data are given, I_{q^2} and $(B-A)^d I$ are constant and they can be ignored. Hence we define a metric called integral cosine similarity (ICS) as:

$$ICS(\tilde{\boldsymbol{p}}, \boldsymbol{q}) = \frac{S_\alpha(\boldsymbol{\eta})}{S_\beta(\boldsymbol{\eta})} = \frac{(\frac{1}{N_r}\sum_{i=1}^{N_r}\tilde{p}(\boldsymbol{x}_i^r|\boldsymbol{\eta}))^2}{\frac{1}{N_u}\sum_{i=1}^{N_u}\tilde{p}^2(\boldsymbol{x}_i^u|\boldsymbol{\eta})}. \tag{8}$$

It can be used to measure the similarity between the UGMF and the data probability density function so as to directly learn the parameter $\boldsymbol{\eta}$ of $\tilde{p}(\boldsymbol{x}|\boldsymbol{\eta})$ from data without the need to compute matrix inverse, which is rather different from traditional learning methods such as EM. In implementation, $1/ICS$ is used for optimization.

3.3 Scaling Constant

After the $1/ICS$ is minimized, we need a scaling constant, which is defined as $C = \int_\Omega \tilde{p}(\boldsymbol{x}|\boldsymbol{\eta})d\boldsymbol{x}$, to normalize the UGMF $\tilde{p}(\boldsymbol{x})$. Resorting to MC sampling again, it can be approximated by

$$\begin{aligned}C &= (B-A)^d\int_\Omega \tilde{p}(\boldsymbol{x}|\boldsymbol{\eta})\frac{1}{(B-A)^d}\,d\boldsymbol{x} \\ &\approx (B-A)^d\frac{1}{N_u}\sum_{i=1}^{N_u}\tilde{p}(\boldsymbol{x}_i^u|\boldsymbol{\eta}) \doteq (B-A)^d S_\gamma(\boldsymbol{\eta}).\end{aligned} \tag{9}$$

The resultant estimated density distribution after normalization is $p(\boldsymbol{x}|\boldsymbol{\eta}) = \frac{1}{C}\tilde{p}(\boldsymbol{x}|\boldsymbol{\eta})$.

3.4 A Siamese Network for Learning Parameters

This paper implements all the calculations in (8) with deep learning operations so that the optimization process can be supported by existing deep learning systems. To this end, we decompose $\tilde{p}(\boldsymbol{x}|\boldsymbol{\eta})$ into some general linear and non-linear operations available in existing deep learning platforms. The decomposed form of $\tilde{p}(\boldsymbol{x}|\boldsymbol{\eta})$ can be expressed as:

$$\begin{aligned}\tilde{p}(\boldsymbol{x}|\boldsymbol{\eta}) &= \sum_{i=1}^{M} \xi_i e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_i})^T \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu_i})} \\ &= \sum_{i=1}^{M} \xi_i e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_i})^T \boldsymbol{B}_i^T \boldsymbol{B_i}(\boldsymbol{x}-\boldsymbol{\mu_i})} \\ &= \sum_{i=1}^{M} \xi_i e^{-\frac{1}{2}(\boldsymbol{B_i x}-\boldsymbol{B_i \mu_i})^T(\boldsymbol{B_i x}-\boldsymbol{B_i \mu_i})} \\ &= \sum_{i=1}^{M} \xi_i e^{-\frac{1}{2}\sum_{v=1}^{d}(\boldsymbol{b}_{i,v}^T \boldsymbol{x}+t_{i,v})^2}\end{aligned} \tag{10}$$

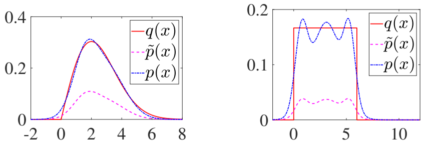
where $\boldsymbol{B}_i = [\boldsymbol{b}_{i,1}^T, \boldsymbol{b}_{i,2}^T, \cdots, \boldsymbol{b}_{i,d}^T]^T$ is a matrix composed of d vectors, and $[t_{i,1}, t_{i,2}, \cdots, t_{i,d}]^T = \boldsymbol{B}_i \boldsymbol{u}_i$. The objective function (8) can be rewritten as:

$$1/ICS(\tilde{\boldsymbol{p}}, \boldsymbol{q}) = \frac{\frac{1}{N_u}\sum_{j=1}^{N_u}(\sum_{i=1}^{M}\xi_i e^{-\frac{1}{2}\sum_{v=1}^{d}(\boldsymbol{b}_{i,v}^T \boldsymbol{x}_j^u+t_{i,v})^2})^2}{(\frac{1}{N_r}\sum_{j=1}^{N_r}\sum_{i=1}^{M}\xi_i e^{-\frac{1}{2}\sum_{v=1}^{d}(\boldsymbol{b}_{i,v}^T \boldsymbol{x}_j^r+t_{i,v})^2})^2}. \tag{11}$$

Note that $\boldsymbol{b}_{i,v}^T \boldsymbol{x}_j^r$ and $\boldsymbol{b}_{i,v}^T \boldsymbol{x}_j^u$ are inner products, which can be implemented by convolution. $t_{i,v}$ is an additive parameter, and therefore it can be set as a bias term. $\frac{1}{N_u}\sum_{j=1}^{N_u}(\cdot)$ and $\frac{1}{N_r}\sum_{j=1}^{N_r}(\cdot)$ can be implemented by global average pooling (GAP). A Siamese NN structure is used and shown in Fig. 2. Note that $S_\beta(\boldsymbol{\eta})$ and $S_\gamma(\boldsymbol{\eta})$ can share the same branch to obtain $\tilde{p}(\boldsymbol{x}_i^u|\boldsymbol{\eta})$ $(i \in \{1 \cdots, N_u\})$. The overall learning process can be realized by minimizing the objective function Eq.(11) by using standard back propagation.

4 Experiments

In this section, the characteristics of the proposed method are demonstrated by experiments, where the Adam [Kingma and Ba, 2014] optimizer was used. The method was also applied to unsupervised anomaly detection. For characteristic evaluations, the learning rate was set to 0.1 and decayed by 0.9 every 20 epochs. For unsupervised anomaly detection, the learning rate was set to 0.01 and decayed by 0.1 every 20 epochs. The total number of training epochs was set to $40,000$ for characteristic evaluations and $80,000$ for unsupervised anomaly detection. The parameters in GMM or UGMF are randomly initialized. In this study, G^{2}M^2 was implemented[1] with Tensorflow (version 1.13.1) and run on a computer equipped with Intel Xeon E5-2640 CPU, 252 GB memory, and NVIDIA 1080Ti GPU (11 GB memory).



(a) $R(2), M = 2$ (b) $U(0,6), M = 3$

Figure 3: Density estimation for some typical distribution functions.

4.1 Characteristic Evaluations

In this part, we first evaluate the performance of G^{2}M^2 in density estimation for 1-D data that drawn from some typical distribution functions. Then we visualize the the learning process with the proposed ICS metric for fitting 2-D Gaussian mixture. Finally, we show the robustness against noisy data and the insensitivity to parameter initialization. In the experiments, the parameters were set as: $N_u = 10^6$, $A = -30$, $B = 30$ for 1-D data and $N_u = 10^6$, $A = -12$, $B = 12$ for 2-D Gaussian mixture data.

UGMF and Normalized Density in Density Estimation. In our method, the parameters of UGMF $\tilde{p}(\boldsymbol{x}|\boldsymbol{\eta})$ are estimated first, and then the normalization term C is calculated to obtain $p(\boldsymbol{x}|\boldsymbol{\eta})$. In this way, our method does not require regularization like traditional methods. In this study, two toy examples were set up to demonstrate this characteristic. G^{2}M^2 was applied to two typical 1-D data distribution functions (with $N_r = 60,000$), including Rayleigh (*R*) and Uniform (*U*). Results are shown in Fig. 3. It can be seen that $\tilde{p}(x)$ is close to true distribution and only differs by a constant. When $\tilde{p}(x)$ was normalized, $p(x)$ was close to the true density with M Gaussian components.

Learning with the ICS. We propose to minimize $1/ICS$ (Eq. (11)) for learning the G^{2}M^2 parameters. In this experiment, we show that as the $1/ICS$ is minimized, the estimated density is close to the true density. We used random samples ($N_r = 60,000$ and $d = 2$) drawn from a Gaussian mixture distribution ($M = 200$), in which 1% of the samples were replaced with some correlated data as noise samples, which may lead to a singular covariance matrix. When the sample number N_u approaches infinite, the ICS can well approximate squared cosine similarity. However, the sample number N_u in implementation is limited, indicating that the obtained $1/ICS$ has an estimation error. We may use the averaged $1/ICS$ as the ground truth. Figure 4 shows that the obtained $1/ICS$ fluctuates around the averaged value. As the averaged $1/ICS$ converges, the estimated density by G^{2}M^2 converges to the true density.

Robustness Analysis. The robustness of the estimation algorithm is very important. In this experiment, the robustness of the proposed method was evaluated in two aspects, including the sensitivity of ICS to noise and the sensitivity of G^{2}M^2 to parameter initialization. The similarity between the estimated density and the true density is evaluated by ALL (average log likelihood) [Kolouri *et al.*, 2018] which is commonly employed by EM-based methods. Besides, as indicated in Section 3.2, Eq. (4) can evaluate the similarity between two

[1] The implementation code is available at https://github.com/media-sec-lab/G2M2

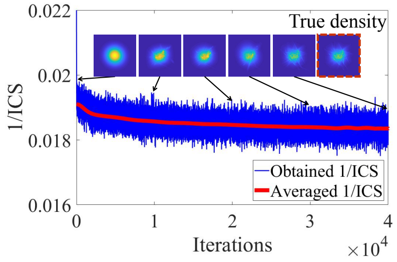


Figure 4: The learning process of $\mathrm{G^2M^2}$ ($M = 200$) with respect to 1/ICS. Each averaged 1/ICS value was obtained by 2000 times of calculation.

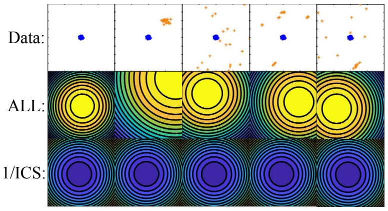


Figure 5: The contour map of ALL and 1/ICS for fitting data. In the first row, the blue points in the center are normal samples, and the orange ones deviated from the center are noisy samples. In the second row, the brighter the yellow, the closer to the global optimal value of ALL. In the third row, the darker the blue, the closer to the optimal value of 1/ICS.

arbitrary functions; therefore we can use the ICS as a kind of metric by replacing the UGMF in Eq. (8) with the density function. Without confusion, we still call it as ICS.

Robustness to Noise: To get a sense of the robustness to noise, we studied a simple scenario by using 1000 clean samples randomly drawn from $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu} = [0, 0]$, and $\boldsymbol{\Sigma} = \begin{bmatrix} 0.02 & 0 \\ 0 & 0.02 \end{bmatrix}$. We set $M = 1$ and assume $\boldsymbol{\Sigma}$ is known. As a result, there is only one unknown parameter $\boldsymbol{\mu}$ in Eq. (8). In this way, the $1/ICS$ value is a function of $\boldsymbol{\mu}$ and it can be visualized. So is ALL. It can be seen from Fig. 5 that for clean data, the global optimum was the same for both metrics. It can also be seen that when there is some noise, the contour map of ALL changes significantly, which leads to its global optimum shifts greatly from the ground-truth of $\boldsymbol{\mu}$. In contrast, the contour map of ICS was less sensitive to noise. The reason is that ALL focuses on fitting every single sample, including a noisy sample. For example, if a sample $\boldsymbol{x}_i$ leads to $p(\boldsymbol{x}_i) = 0$, then $log(p(\boldsymbol{x}_i)) = -\infty$, which indicates that the penalty for ALL is infinite. In this case, even the change of a few samples may cause a significant change in ALL. However, ICS pays more attention to the overall sample distribution. According to (8), only when $\tilde{p}(\boldsymbol{x}_i) = 0$ holds for all samples $\boldsymbol{x}_i$, the value of $1/ICS$ will become infinite, making it be more robust to noise.

Insensitivity to Parameter Initialization: We performed an experiment to demonstrate $\mathrm{G^2M^2}$ is insensitive to random initialization of parameters. Our method was compared with EM-based methods, including EM-GMM-R by adding regularization values to the diagonal of the covariance matrix, and EM-GMM-D by setting the covariance matrix to be diagonal. The regularization parameter of EM-GMM-R was empirically selected from 10^{-6} to 10^{-1} and we found 10^{-4} achieved the best performance. In this scenario, the 2-D Gaussian mixture data polluted with noise, which was previous used and shown in the top-right corner of Fig. 4, was used again. We learned the optimized parameters from each run of random initialization using $\mathrm{G^2M^2}$ and EM-based algorithms. We repeated 30 runs. Some estimated density functions are illustrated in Fig. 6. Moreover, we show the histograms in terms of ALL and $1/ICS$ to evaluate the discrepancy between the fitted model and the true data distribution. It can be seen that $\mathrm{G^2M^2}$ achieved better performance in most runs of initialization, showing its good robustness against parameter randomization.

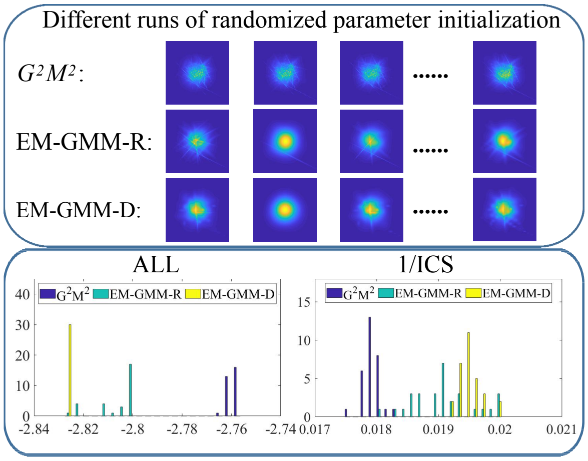


Figure 6: The results of 30 runs of randomized parameter initialization for EM-GMM-R, EM-GMM-D, and $\mathrm{G^2M^2}$ in fitting Gaussian mixture data ($M = 200$). The top shows the estimated density for four different runs of random initialization. The bottom shows the histograms of all 30 runs in term of ALL and 1/ICS.

4.2 Unsupervised Anomaly Detection

In the application of unsupervised anomaly detection, a GMM model is used to learn the distribution of normal samples, while anomalies are identified as low probability samples. To evaluate the unsupervised anomaly detection performance, we conducted experiments on the datasets for outlier detection[2], including Lympho, Cardio, Annthyroid, Shuttle, Pima, Pendigits, Satimage2, Arrthymia, Musk, Mnist, and Optdigits. Our method was compared with conventional and state-of-the-art methods, including EM-GMM-R, EM-GMM-D, iForest [Liu *et al.*, 2012], AvgKNN [Angiulli and Pizzuti, 2002], FOD [Kriegel *et al.*, 2008], MOGAAL [Liu *et al.*, 2020], and DAGMM [Zong *et al.*, 2018]. The performance of these methods was evaluated by the area under curve (AUC). For both of $\mathrm{G^2M^2}$ and EM-based GMM, the training data were normalized to $[-1, 1]$ and the same scaling factor was used for the testing data. The parameters of $\mathrm{G^2M^2}$ were set as follows: $N_u = 10^6$, $A = -1$, $B = 1$, and $M = 8$.

[2] http://odds.cs.stonybrook.edu/

Dataset	Num	Dim	iForest	AvgKNN	FOD	MOGAAL	EM-GMM-R	EM-GMM-D	G^2M^2	DAGMM	DA-G^2M^2
Lympho	148	18	0.993	0.974	0.911	0.910	0.935	0.916	**1.000**	0.996	**1.000**
Pima	768	8	0.676	0.708	0.679	0.758	0.723	0.653	**0.781**	0.568	0.667
Cardio	1831	21	0.924	0.724	0.569	0.914	0.841	0.861	**0.978**	0.878	0.952
Satimage2	5803	36	0.996	0.954	0.819	0.961	0.997	0.995	0.993	0.996	**0.997**
Pendigits	6870	16	0.950	0.749	0.688	0.976	0.992	0.959	0.992	0.963	**0.993**
Annthyroid	7200	6	0.632	0.693	0.730	0.690	0.935	**0.954**	0.934	0.634	0.803
Shuttle	49097	9	0.997	0.654	0.623	0.907	0.991	0.999	0.999	0.985	**1.000**
Optdigits	5216	64	0.699	0.629	0.816	0.690	-	-	-	0.757	**0.931**
Mnist	7603	100	0.801	0.848	0.782	0.909	-	-	-	0.784	**0.927**
Musk	3062	166	0.999	0.799	0.816	0.798	-	-	-	1.000	**1.000**
Arrhythmia	452	274	0.820	0.786	0.769	0.751	-	-	-	0.643	**0.804**
Average	-	-	0.862	0.774	0.746	0.842	0.916	0.905	**0.954**	0.837	0.916

Table 1: The performance (AUC) of unsupervised anomaly detection (Num: Number of feature, Dim: Feature dimension).

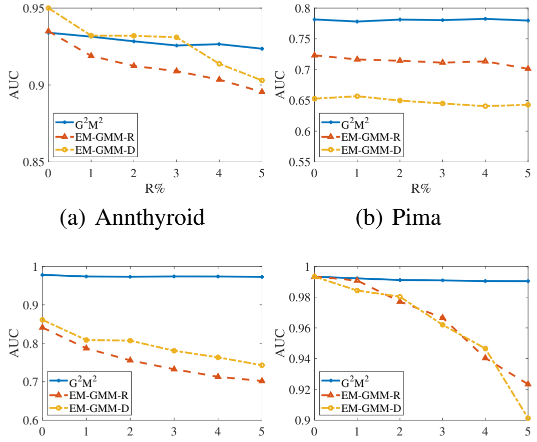
Detection Performance. In the first set of experiments, the parameter setting in [Liu *et al.*, 2020] with completely clean training data was adopted. In each run, 60% of the data taken by random sampling were used for training, and the remaining data were used for testing. It should be noted that only the data samples from the normal class were used to train the models. Table 1 shows the AUC of G^2M^2 and its counterparts. It can be seen that the performance of G^2M^2 was higher than that of the EM-based methods by $4\% \sim 5\%$ on average and $13\% \sim 14\%$ at most. Thus, G^2M^2 is more competitive than EM-based methods.

Since GMM is too complex to handle high-dimensional data, some methods such as DAGMM [Zong *et al.*, 2018] use an autoencoder to reduce feature dimension, and optimize the GMM jointly with the autoencoder. In this study, the EM learning process in DAGMM was replaced with G^2M^2. Specifically, the output of the last module of autoencoder, i.e., the estimation network in DAGMM, is the input of G^2M^2, and the EM-based optimization term was replaced with (11). The adapted method was called DA-G^2M^2. It can be seen in Table 1 that the performance of DA-G^2M^2 was significantly improved. In this case, the AUC was improved by nearly 8% on average compared to DAGMM, showing DA-G^2M^2 is highly competitive as compared with the-state-of-art methods.

Robustness to Noise. In the second set of experiment, the sensitivity of G^2M^2 to contaminated training data was investigated. In each run, 60% of the samples from the normal class were used. In addition, some samples from the anomaly class, of which the amount was equal to $R\%$ of the amount of normal training samples, were mixed into the training set as contamination noise. Fig. 7 illustrates the AUC of EM-based GMM and G^2M^2. It can be seen that G^2M^2 achieved a stable detection accuracy when the contamination rate was increased from 1% to 5%. The results indicate traditional methods require a high quality training dataset (i.e., clean data with no contamination) to achieve good detection accuracy, while our new method G^2M^2 is much robust and less sensitive to the contamination of abnormal data.

5 Concluding Remarks

In this paper, a novel approach called G^2M^2 is proposed to overcome the singularity problem of conventional learning methods for GMM. Unlike traditional methods by transforming the data or constraining the form of the estimated covariance matrix, this paper presents a general GMM learning method that can relax the assumption of nonsingularity. Meanwhile, an optimization function based on a new metric called ICS is proposed to facilitate the parameter estimation in G^2M^2. The effectiveness and robustness of G^2M^2 have been verified through experiments. While the new method, equipped with a Siamese neural network for learning parameters, is simple and not complicated compared to many existing deep learning architectures such as ResNet, Transformer, etc., it is computationally more demanding than EM-based solutions. In future work, we will try to overcome this problem especially for high-dimensional data.



(a) Annthyroid

(b) Pima

(c) Carido

(d) Shuttle

Figure 7: Unsupervised anomaly detection results when the training set is contaminated by some samples from the anomaly class.

Acknowledgments

This work was supported in part by NSFC (Grant 61872244, 62072313), Guangdong Basic and Applied Basic Research Foundation (Grant 2019B151502001), Shenzhen Program (Grant JCYJ20200109105008228), and Alibaba Innovative Research (AIR) Program.

References

[Angiulli and Pizzuti, 2002] Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. In *European conference on principles of data mining and knowledge discovery*, pages 15–27. Springer, 2002.

[Ban *et al.*, 2018] Zhihua Ban, Jianguo Liu, and Li Cao. Superpixel segmentation using gaussian mixture model. *IEEE Transactions on Image Processing*, 27(8):4105–4117, 2018.

[Cho *et al.*, 2019] Byung Joon Cho, Jun-Min Lee, and Hyung-Min Park. A beamforming algorithm based on maximum likelihood of a complex gaussian distribution with time-varying variances for robust speech recognition. *IEEE Signal Processing Letters*, 26(9):1398–1402, 2019.

[Davari *et al.*, 2018] Amirabbas Davari, Erchan Aptoula, Berrin Yanikoglu, Andreas Maier, and Christian Riess. Gmm-based synthetic samples for classification of hyperspectral images with limited training data. *IEEE Geoscience and Remote Sensing Letters*, 15(6):942–946, 2018.

[Dempster *et al.*, 1977] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[Do and Ohsaki, 2021] Bach Do and Makoto Ohsaki. Gaussian mixture model for robust design optimization of planar steel frames. *Structural and Multidisciplinary Optimization*, 63(1):137–160, 2021.

[Duda *et al.*, 1973] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Kolouri *et al.*, 2018] Soheil Kolouri, Gustavo K Rohde, and Heiko Hoffmann. Sliced wasserstein distance for learning gaussian mixture models. In *the 30th IEEE Conference on Computer Vision and Pattern Recognition*, pages 3427–3436, 2018.

[Kriegel *et al.*, 2008] Hans-Peter Kriegel, Matthias Schubert, and Arthur Zimek. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 444–452, 2008.

[Liu *et al.*, 2012] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1):1–39, 2012.

[Liu *et al.*, 2020] Yezheng Liu, Zhe Li, Chong Zhou, Yuanchun Jiang, Jianshan Sun, Meng Wang, and Xiangnan He. Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1517–1528, 2020.

[Ma *et al.*, 2019] Jiayi Ma, Xingyu Jiang, Junjun Jiang, and Yuan Gao. Feature-guided gaussian mixture model for image matching. *Pattern Recognition*, 92:231–245, 2019.

[McLachlan and Krishnan, 1997] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, New York, 1997.

[Moon, 1996] T.K. Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, 1996.

[Ng, 2000] Andrew Ng. Cs229 lecture notes. *CS229 Lecture notes*, 1(1):1–3, 2000.

[Qu *et al.*, 2020] Jiahui Qu, Qian Du, Yunsong Li, Long Tian, and Haoming Xia. Anomaly detection in hyperspectral imagery based on gaussian mixture model. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–14, 2020.

[Rousseeuw and Hubert, 2011] Peter J Rousseeuw and Mia Hubert. Robust statistics for outlier detection. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, 1(1):73–79, 2011.

[Sorenson and Alspach, 1971] Harold W Sorenson and Daniel L Alspach. Recursive bayesian estimation using gaussian sums. *Automatica*, 7(4):465–479, 1971.

[Venegas *et al.*, 2019] Pablo Venegas, Noel Pérez, Diego Benítez, Román Lara-Cueva, and Mario Ruiz. Combining filter-based feature selection methods and gaussian mixture model for the classification of seismic events from cotopaxi volcano. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(6):1991–2003, 2019.

[Xu and Wunsch, 2005] Rui Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.

[Xu *et al.*, 2014] Xianzhen Xu, Lei Xie, and Shuqing Wang. Multimode process monitoring with pca mixture model. *Computers & Electrical Engineering*, 40(7):2101–2112, 2014.

[Yang *et al.*, 2019] Linxiao Yang, Ngai-Man Cheung, Jiaying Li, and Jun Fang. Deep clustering by gaussian mixture variational autoencoders with graph embedding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6440–6449, 2019.

[Zhou *et al.*, 2020] Xiaokang Zhou, Yiyong Hu, Wei Liang, Jianhua Ma, and Qun Jin. Variational lstm enhanced anomaly detection for industrial big data. *IEEE Transactions on Industrial Informatics*, 17(5):3469–3477, 2020.

[Zong *et al.*, 2018] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.

[Zoran *et al.*, 2012] Daniel Zoran, Yair Weiss, et al. Natural images, gaussian mixtures and dead leaves. In *the 25th International Conference on Neural Information Processing Systems*, volume 2, pages 1745–1753, 2012.