I.J. Intelligent Systems and Applications, **2024, 5, 10-21**
Published Online on October 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2024.05.02

Data-driven Approximation of Cumulative Distribution Function Using Particle Swarm Optimization based Finite Mixtures of Logistic Distribution

Rajasekharreddy Poreddy*
Department Electronics and Communication Engineering, National Institute of Technology Tiruchirappalli-620015, Tamil Nadu, India
E-mail: sekharpraja@gmail.com
ORCID iD: https://orcid.org/0000-0002-7004-4805
*Corresponding Author

Gopi E. S.
Department Electronics and Communication Engineering, National Institute of Technology Tiruchirappalli-620015, Tamil Nadu, India
E-mail: esgopi@nitt.edu
ORCID iD: https://orcid.org/0000-0001-7502-6756

Received: 20 November 2023; Revised: 14 January 2024; Accepted: 10 September 2024; Published: 08 October 2024

Abstract: This paper proposes a data-driven approximation of the Cumulative Distribution Function using the Finite Mixtures of the Cumulative Distribution Function of Logistic distribution. Since it is not possible to solve the logistic mixture model using the Maximum likelihood method, the mixture model is modeled to approximate the empirical cumulative distribution function using the computational intelligence algorithms. The Probability Density Function is obtained by differentiating the estimate of the Cumulative Distribution Function. The proposed technique estimates the Cumulative Distribution Function of different benchmark distributions. Also, the performance of the proposed technique is compared with the state-of-the-art kernel density estimator and the Gaussian Mixture Model. Experimental results on κ−μ distribution show that the proposed technique performs equally well in estimating the probability density function. In contrast, the proposed technique outperforms in estimating the cumulative distribution function. Also, it is evident from the experimental results that the proposed technique outperforms the state-of-the-art Gaussian Mixture model and kernel density estimation techniques with less training data.

Index Terms: Cumulative Distribution Function (CDF), Probability Density Function (PDF), Logistic Mixture Model (LMM), Particle Swarm Optimization (PSO), Extreme Learning Machine (ELM), Interior-point Method.

1. Introduction

Density estimation plays a crucial role in understanding and characterizing the behavior of signals in fading wireless channels. By accurately estimating the density of signal strength variations, we gain valuable insights into the statistical properties of the channel. This knowledge is instrumental in designing robust communication systems that can adapt to changing channel conditions. Density estimation enables us to model the fading phenomena, predict signal behavior, and optimize system parameters to ensure reliable and efficient wireless communication. It also aids in the development of advanced modulation, coding, and equalization techniques that mitigate the effects of fading, ultimately enhancing the quality and performance of wireless networks.

Most density estimation techniques attempt to approximate the Probability Density Function (PDF), which is then summed or integrated to yield the Cumulative Distribution Function (CDF). As a result, in most cases, a closed-form expression for the CDF cannot be obtained. The closed form expression of the CDF is critical in calculating performance metrics such as outage probability [1, 2], detection probability, False Alarm Probability, and Bit Error Probability in wireless fading environments [3]. As a result, studies have been carried out to obtain the closed-form expressions for the

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

PDF and CDF of fading distributions [3–6]. In [3], authors have expressed the CDF of "quadratic-form" receivers over generalized fading channels with tone interference in the form of a single integral with finite limits and integrand composed of elementary functions. The authors of [4] derived the exact closed-form expressions for the bivariate Nakagami-m cumulative distribution function (CDF) with positive integer fading severity index m in terms of a class of hypergeometric functions. Obtaining the closed form expression of CDF of Bivariate Gamma Distribution with arbitrary parameters is studied in [5]. Authors of [6] used Berry-Esseen theorem and the method of tilted distributions to derive a simple tight closed-form approximation for the tail probabilities of a sum of independent but not necessarily identically distributed random variables. Also, CDF can be used to generate the samples of the distribution of the data which can be used for data augmentation. Authors of [7] and [8] have proposed nonparametric and bivariate nonparametric random variate generation and using a piecewise-linear CDF to generate the data. The closed-form expressions of the PDF and CDF of the many fading distributions are proposed as a function of complex functions (such as hypergeometric functions, q-functions, error functions, etc.,). For example, the PDF of the $\kappa - \mu$ distribution is expressed in terms of a modified Bessel function, and the CDF is expressed in terms of a generalized Marcum-Q function in [9]. Hence, we considered a novel technique of obtaining the closed form approximate of CDF using the finite mixtures of Logistic Distribution. The finite mixtures of Logistic distribution are referred to as Logistic Mixture Model (LMM) in the rest of the paper. Since the LMM cannot be modeled using the Maximum likelihood estimator, we proposed two computational intelligence techniques.

The main contributions of this paper are

- To estimate the CDF using finite mixtures of the logistic distribution. This results in a closed-form expression for the estimated CDF, which is expressed in terms of simple exponential functions.
- To use computational intelligence techniques such as particle swarm optimization or extreme learning machine to determine the parameters of the mixtures of the logistic distribution.

The rest of the paper is organized as follows. Section 2 introduces the LMM using the Multivariate Logistic distribution. Section 3 describes the proposed algorithms to approximate the CDF. Section 4 presents the experimental results on the effectiveness of the Mixtures of the Logistic distribution using the proposed algorithms. Section 5 provides the conclusion on the observations.

2. Related Works

Researchers have been continuously developing methods for data-driven probability distribution estimation. Addressing the computational challenges of large, multidimensional datasets, one approach focuses on efficient calculation of the empirical cumulative distribution function (Empirical CDF) [10]. This technique, employing lexicographical summation and divide-and-conquer methods, paves the way for faster kernel density estimation for complex data.

Beyond efficiency improvements, alternative estimation techniques are being explored. Moving Extreme Ranked Set Sampling is a method used to estimate both the CDF and a reliability parameter, offering advantages over traditional approaches like simple random sampling [11]. Extreme Learning Machines present a competitive alternative for PDF and CDF estimation. Leveraging their low computational cost and random hidden layer assignment, ELMs are particularly attractive for large datasets [12]. A fast kernel density estimator based local linear estimate have shown their efficiency in estimating the conditional CDF in dependent functional data case [13].

Looking at more recent advancements, data-driven deep density estimation emerges as a powerful tool. Promising accurate and efficient PDF estimation regardless of data dimensionality or size, DDE offers advantages but may require access to the original PDF in some form during the estimation process [14]. Another approach tackles classification problems in a unique way. Instead of directly analyzing empirical data, this method estimates the underlying probability distribution through the Fredholm equation [15]. This approach offers a distinct perspective on data-driven probability estimation.

3. LMM Using Multivariate Logistic Distribution

Different variants of the multivariate logistic distribution were proposed in many research papers. In this paper, the multivariate logistic distribution of the type I proposed by Malik et. al. [16], with some arbitrary center μ and scale parameters Σ is considered to build the mixture model. The joint cumulative distribution function of $x_1, x_2, \ldots, x_n$ which follow logistic distribution is defined as

$$L(x_1, x_2, \ldots, x_n | \mu, \Sigma) = \frac{1}{1+\sum_{j=1}^{n} \exp\left(-\frac{(x_j-\mu^j)}{2\sigma^j}\right)} \quad (1)$$

The corresponding joint probability density function is

$$l(x_1, x_2, \ldots, x_n|\mu, \Sigma) = \frac{n!\exp\left(-\sum_{j=1}^{n}\frac{(x_j-\mu^j)}{2\sigma^j}\right)}{\prod_{j=1}^{n}\sigma^j\left[1+\sum_{j=1}^{n}\exp\left(-\frac{(x_j-\mu^j)}{2\sigma^j}\right)\right]^{n+1}} \quad (2)$$

where $\mu = (\mu_1, \mu_2, \ldots, \mu_n)$ and $\Sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$. Let $\mathrm{x} = [x_1, x_2, \ldots, x_n]^T$ be the random vector with the CDF $F_X(\mathrm{x};\ \Theta)$, then $F_X(\mathrm{x};\ \Theta)$ can be written as linear superposition of the multivariate logistic distributions in the following form

$$F_X(\mathrm{x}|\Theta) = \sum_{k=1}^{K}\pi_k L(\mathrm{x}; \mu_k, \Sigma_k) \quad (3)$$

where K is the number of mixture components, $\Theta = [\pi, \mu, \Sigma]$, $\pi = [\pi_1, \pi_2, \ldots, \pi_n]$, $0 < \pi_k < 1$ and $\sum_{k=1}^{K}\pi_k = 1$.

For $k = 1, 2, \ldots, K, \mu = [\mu_1, \mu_2, \ldots, \mu_K]$, $\Sigma = [\Sigma_1, \Sigma_2, \ldots, \Sigma_K]$ and $\mu_k = (\mu_k^1, \mu_k^2, \ldots, \mu_k^n) \in \Re^n$, $\Sigma_k = (\sigma_k^1, \sigma_k^2, \ldots, \sigma_k^n) \in (0, \infty)^n$. The corresponding PDF of X, which can be obtained by differentiating $F_X(\mathrm{x};\ \Theta)$, is

$$f_X(\mathrm{x}|\Theta) = \sum_{k=1}^{K}\pi_k l(\mathrm{x}; \mu_k, \Sigma_k) \quad (4)$$

Z.Q. Shi et al. [17] showed that the LMM mentioned in equation (3) is shown to be identifiable. But the maximum likelihood estimator cannot be used to model the parameters of the (3), since the closed-form expressions cannot be obtained for μ and Σ. Hence, (3) is solved in a semi-supervised manner. The proposed algorithms to model the parameters of LMM are explained in section 3. The modeled parameters can be used to estimate the cumulative distribution function of x.

4. Proposed Algorithms to Model the Lmm

In this section, we present two algorithms to model the parameters of the LMM. They are

- Extreme Learning Machine (ELM) framework using the interior method.
- Multilevel Particle Swarm Optimization (Mu-PSO).

The parameters of the LMM are computed by the two algorithms in a semi-supervised fashion. The basic idea of the proposed algorithms is to use the empirical approximation of the CDF (also known as empirical CDF) as the target value and consider the Mean Square Error (MSE) as the objective function which must be minimized. The empirical CDF of some arbitrarily uniformly distributed points is computed based on the available data. These arbitrary points

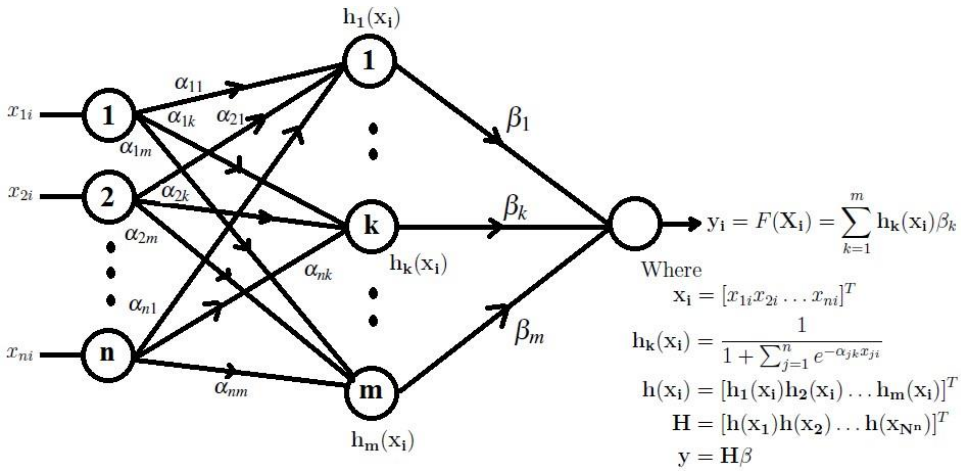

Fig.1. Single hidden layer feed-forward network (SLFN)

and the obtained Empirical CDF values are considered input-target pairs to compute the parameters (output weights) of the ELM and the optimum particle in the swarm of the Mu-PSO.

The Empirical CDF of an arbitrary data point x is obtained by

$$I(x) = \frac{\text{Number of available data samples} \le x}{\text{Total number of available data samples}} \quad (5)$$

The *x* and *I(x)* pair are assumed as the input and target values of the proposed algorithms.

4.1. Modeling LMM Using Interior Point Method in ELM Framework

In this section, (3) is modeled using the interior point method in the ELM framework. The individual mixtures $L(\mathrm{x};\mu_k,\Sigma_k)$ in (3) are considered as the hidden layer activations and the mixing probabilities π_k as the output layer weights of a Single hidden Layer Feed-forward Network. The Empirical CDF values *I(x)* of the corresponding input data points x are considered as the target of the SLFN. The details of the ELM algorithm are presented in section 4.1.1.

A. Extreme Learning Machine (ELM)

An Extreme Learning Machine (ELM) is a Single hidden Layer Feed-forward Network that randomly chooses hidden node weights and analytically determines the output weights [18]. Fig. 1 shows the network structure of SLFN where $\alpha_{jk}, j = 1,\dots,n, k = 1,\dots,K$ are hidden layer weights, $\beta_k, k = 1,2,\dots,K$ are output layer weights. The output $y_i, i = 1,\dots,N$ corresponding to the input $\mathrm{x}_i = [x_{1i},\dots,x_{ni}]$, for $i = 1,\dots,N$ is obtained by $y_i = \sum_{k=1}^{K} h_k(\mathrm{x}_i)\beta_k$. The output can be represented in matrix form for all inputs $\mathrm{x}_i, i = 1,\dots,N$ as,

$$\mathrm{y} = \mathrm{H}\beta \tag{6}$$

Algorithm 1: Extreme Learning Machine (ELM)

Given a training set $\aleph = \{(\mathrm{x}_i;\ t_i)|\mathrm{x}_i \in \Re^n;\ t_i \in \Re;\ i = 1,\dots,N^n\}$, activation function $h_k(\mathrm{x}_i)$, number of hidden nodes *K* and a target $T = [t_1,\dots,t_N]^T$.

Step 1: Randomly assign input weights $\alpha_{jk}, j = 1,\dots,n, k = 1,\dots,K$.

Step 2: Compute the hidden layer output matrix, H.

Step 3: Find the output weight vector, β as

$$\beta = \mathrm{H}^{\dagger}T$$

Step 4: Compute the output of SLFN for a test data, $\mathrm{x}_{\mathrm{test}}$ can be found by

$$y(\mathrm{x}_{\mathrm{test}}) = h(\mathrm{x}_{\mathrm{test}})\mathrm{H}^{\dagger}T = h(\mathrm{x}_{\mathrm{test}})(\mathrm{H}^T\mathrm{H}) - \mathrm{H}^T T$$

where $\mathrm{H} = [h(\mathrm{x}_1)h(\mathrm{x}_2)\dots h(\mathrm{x}_N)]^T$ is the hidden layer activation matrix of all the input samples, $h(\mathrm{x}_i) = [h_1(\mathrm{x}_i),\dots,h_K(\mathrm{x}_i)]^T$ is the hidden layer activation vector of the input sample x_i, $h_k(\mathrm{x}_i)$ is the activation of the kth hidden node, $\beta = [\beta_1,\dots,\beta_K]^T$, and $T = [t_1,\dots,t_N]^T$. The values of α_{jk} are randomly assigned and the values of βk are computed by minimizing the MSE,

$$MSE = ||\mathrm{H}\beta - T||^2 \tag{7}$$

The values of β can be computed by solving (7) using the Least Square method as

$$\beta = \mathrm{H}^{\dagger}T \tag{8}$$

where $\mathrm{H}^{\dagger} = (\mathrm{H}^T\mathrm{H})^{-1}\mathrm{H}^T$ is the pseudo-inverse of the matrix H.

B. Interior Point Method

The ELM solves its objective function using the Least Squares (LS) method, but the proposed algorithm uses the interior point method to solve the least squares problem, to satisfy the constraints of the probability. The objective function of the ELM algorithm in equation (7) does not impose constraints on the mixing probabilities π_k of (3). Hence the π_k obtained in the section 4.1.1 may be negative or greater than one [19], which violates the axioms of the probability. To avoid this, equation 7 is solved using the interior point method by imposing the axioms of the probability as constraints. Hence the objective of the ELM approach can be modified by imposing the constraints as follows

$$J = \min_{\pi} MSE = \min_{\pi} ||L(\mathrm{X};\ \mu,\Sigma)\pi - T||^2 \tag{9}$$

subject to the constraints, $0 < \pi_{\mathrm{k}} < 1, \mathrm{k} = 1,2,\dots,\mathrm{K}$ and $\sum_{k=1}^{K}\pi_k = 1$. Where $\mathrm{X} = [\mathrm{x}_1,\mathrm{x}_2,\dots,\mathrm{x}_N]$, $\mu = (\mu_1,\mu_2,\dots,\mu_K)$, $\Sigma = (\Sigma_1,\dots,\Sigma_N)$ and $\pi = [\pi_1,\pi_2,\dots,\pi_N]^T$.

The objective function in (9) is a linear programming problem with inequality constraints. This problem can be solved using the interior-point methods [20]. Once the parameters μ,Σ, and π are computed using the interior-point method, the CDF of a data point x can be computed using (3). The algorithmic framework of the interior-point method in ELM fashion to model the parameters of (3) is presented in Algorithm 2.

Algorithm 2: Modeling LMM using Interior Point method in ELM framework

Given a training set $\aleph = \{(x_i;\ t_i)|x_i \in \Re^n;\ t_i \in \Re;\ i = 1,\ldots,N^n\}$, activation function $L(x_i|\mu_k,\Sigma_k) = \frac{1}{1+\sum_{j=1}^{n}\exp\left(-\frac{\left(x_{ji}-\mu_k^j\right)}{2\sigma_k^j}\right)}$, number of hidden nodes K and a target $T = [t_1,\ldots,t_N]^T$.

Step 1: Randomly assign μ_k^j and $\sigma_k^j, \forall j = 1,\ldots,n$, and $k = 1,2,\ldots,K$.

Step 2: Calculate the matrix $L(X;\ \mu,\Sigma) = [L(x_1;\ \mu,\Sigma),\ldots,L(x_N;\ \mu,\Sigma)]^T$,

Where $L(x_i;\ \mu,\Sigma) = [L(x_i;\ \mu_1,\Sigma_1),\ldots,L(x_i;\ \mu_K,\Sigma_K)]^T$.

Step 3: Formulate the Optimization Problem as

$$\min_\pi ||L(X;\ \mu,\Sigma)\pi - T||^2$$

such that $0 < \pi_k < 1$ and $\sum_{k=1}^{K}\pi_k = 1$.

Step 4: Solve for π by minimizing the MSE using Interior Point Method.

Step 5: CDF of a test data, x_{test} can be found by $F_X(x_{test}) = L(x_{test};\ \mu,\Sigma)\pi$.

4.2. Modeling LMM using Mu-PSO

The Mu-PSO proposed in this section is inspired by the Expectation-Maximization technique [21] where the maximum likelihood of a model is obtained by first finding the expectation and then maximizing the model. The objective function mentioned in (9) is solved using the Mu-PSO algorithm in Expectation-Maximization fashion. The detailed algorithm of the Mu-PSO is presented in Algorithm 3. As in the Expectation-Maximization technique,

Algorithm 3: Mu-PSO to model LMM

Initialize the parameters μ and Σ randomly in $(-\infty,\infty)$ and $(0,\infty)$ respectively.

num_iter - The number of iterations.

for *l* = 1 to *num_iter*

Step 1 - Expectation stage: Compute π_l as described in Algorithm 4. The obtained π_l will be used as one of the particles of Algorithm 4 in the next iteration.

Step 2 - Maximization stage: Compute μ_l and Σ_l as described in Algorithm 5. This μ_l and Σ_l will be included as one of the particles of the respective swarms in the next iteration.

Step 3: Compute the objective function in, $J_l = J(\mu_l,\Sigma_l,\pi_l)$

Step 4: if $(J_{l-1} - J_l) < 10^{-6}$

$\mu = \mu_l, \Sigma = \Sigma_l, \pi = \pi_l$ and break the loop and go to Step 5.

end if

end for

Step 5: Use the values of μ,Σ and π to compute the CDF of a test sample x.

- Expectation: The values of π are computed using PSO [22-24] algorithm by randomly initializing the μ and Σ.
- Maximization: Once π are computed, then the values of μ and Σ can be obtained using PSO.
- This process is repeated until convergence of either the parameters or the objective function in (9).

The values of π, μ and Σ obtained in the previous iteration are used as one of the particles in the corresponding PSO algorithms. To make sure that π follow the axioms of probability, the absolute value of the next movement of the particles of the PSO used to compute the π are taken, and then normalized so that the sum of each particle equals to one in Algorithm 4. The values of Σ must range in $(0,\infty)$, hence the values of Σ are made positive by taking the absolute values while updating the next position of the population in Algorithm 5.

Algorithm 4: Expectation stage

Initialization:

- Assume the number of particles in the swarm is *M*, the dimension of each particle is *n*, *c*1, *c*2 are the individual and group learning rates, *r*1 and *r*2 are uniformly distributed random numbers between 0 and 1, |.| denotes the modulus and the minimum value of the objective function *J* is saved in variable min_*J* to track the convergence of the algorithm.
- Initialize the *M* particle positions of the mixing probabilities, π randomly as $\pi_{p_1},\pi_{p_2},\ldots,\pi_{p_M}$ and the tentative local decisions taken by the particles randomly as $\pi_{ld_1},\pi_{ld_2},\ldots,\pi_{ld_M}$. The parameters μ and Σ are fixed with the values obtained in maximization stage.

for iteration = 1 to 25

Step 1: Compute the functional values of $J(\pi_{ld_m}), \forall m$.

Step 2: Find the minimum among $J(\mu,\Sigma,\pi_{ld_m}), \forall m$ and declare the particle position as global best, π_{ld_g} . i.e.,

$\pi_{ld_g} = \arg\min_{l=1,\dots,L} J(\mu, \Sigma, \pi_{ld_m})$ and $J(\mu, \Sigma, \pi_{ld_g}) = \arg\min_{l=1,\dots,L} J(\mu, \Sigma, \pi_{ld_m})$.

Step 3: Identify the moved next position of the particles.

for m = 1 to M

$\pi_{p_m}_next = |\pi_{p_m} + c1\ r1(\pi_{ld_m} - \pi_{p_m}) + c2\ r2(\pi_{ld_g} - \pi_{p_m})|$

$\pi_{p_m}_next = \pi_{p_m}_next / sum(\pi_{p_m}_next)$

end for

Step 4: Assign the current positions of the M particles as $\pi_{p_m} = \pi_{p_m}_next, \forall m$ and obtain the tentative decision taken by each particle for further movement is obtained as

for m = 1 to M

if $J(\mu, \Sigma, \pi_{p_m}_next) < J(\mu, \Sigma, \pi_{ld_m})$

$\pi_{ld_m} = \pi_{p_m}_next$

end if

end for

end for

Step 5: Compute the functional values $J(\mu, \Sigma, \pi_{p_m}), \forall m$ and find the $\pi = \arg\min_{m=1,\dots,M} J(\pi_{p_m})$

Algorithm 5: Maximization stage

Initialization:

- Assume the number of particles in the swarm is L, the dimension of each particle is n, $c3$, $c4$ are the individual and group learning rates, $r3$ and $r4$ are uniformly distributed random numbers between 0 and 1, $|.|$ denotes the modulus and The minimum value of the objective function J is saved in the variable min_J to track the convergence of the algorithm.
- Initialize the L particle positions of the parameters μ and Σ randomly in $(-\infty, \infty)$ and $(0, \infty)$, respectively, as $\mu_{p_1}, \mu_{p_2}, \dots, \mu_{p_L}$ and $\Sigma_{p_1}, \Sigma_{p_2}, \dots, \Sigma_{p_L}$, the tentative local decisions taken by the particles randomly as $\mu_{ld_1}, \mu_{ld_2}, \dots, \mu_{ld_L}$ and $\Sigma_{ld_1}, \Sigma_{ld_2}, \dots, \Sigma_{ld_L}$. The mixing probabilities π obtained in the expectation stage are used.

for iteration = 1 to 25

Step 1: Compute the functional values of $J(\mu_{ld_l}, \Sigma_{ld_l}, \pi), \forall l$.

Step 2: Find the minimum among $J(\mu_{ld_l}, \Sigma_{ld_l}, \pi), \forall l$ and declare the particle position as global best, μ_{ld_g} and Σ_{ld_g}.

$\mu_{ld_g}, \Sigma_{ld_g} = \arg\min_{l=1,\dots,L} J(\mu_{ld_l}, \Sigma_{ld_l}, \pi)$ and $J(\mu_{ld_g}, \Sigma_{ld_g}, \pi) = \min_{l=1,\dots,L} J(\mu_{ld_l}, \Sigma_{ld_l}, \pi)$.

Step 3: Identify the moved next position of the particles.

for l = 1 to L

$\mu_{p_l}_next = \mu_{p_l} + c3r3(\mu_{ld_l} - \mu_{p_l}) + c4r4(\mu_{ld_g} - \mu_{p_l})$

$\Sigma_{p_l}_next = |\Sigma_{p_l} + c3r3(\Sigma_{ld_l} - \Sigma_{p_l}) + c4r4(\Sigma_{ld_g} - \Sigma_{p_l})|$

end for

Step 4: Assign the current positions of the L particles as $\mu_{p_l} = \mu_{p_l}_next$ and $\Sigma_{p_l} = \Sigma_{p_l}_next, \forall l$. Obtain the tentative decision taken by each particle for further movement is obtained as

for l = 1 to L

if $J(\mu_{p_l}_next, \Sigma_{p_l}_next, \pi) < J(\mu_{ld_l}, \Sigma_{ld_l}, \pi)$

$\mu_{ld_l} = \mu_{p_l}_next$

$\Sigma_{ld_l} = \Sigma_{p_l}_next$

end if

end for

end for

Step5: Compute the functional values $J(\mu_{p_l}, \Sigma_{p_l}, \pi), \forall l$ and find $\mu, \Sigma = \arg\min_{l=1,2,\dots L} J(\mu_{p_l}, \Sigma_{p_l}, \pi)$

5. Experimental Results

In this section, the effectiveness of the proposed algorithms to model the LMM for approximating the CDF is demonstrated using the $\kappa - \mu$ fading wireless environment. The envelope probability density function of the $\kappa - \mu$ distribution is defined using the modified Bessel functions as follows [9],

$$f_R(r) = \frac{2\mu(1+\kappa)^{\frac{\mu+1}{2}}}{\kappa^{\frac{\mu-1}{2}} exp\,(\mu\kappa)} r^{\mu} \exp(-\mu(1 + \kappa)r^2)\, I_{\mu-1}\left[2\mu\sqrt{\kappa(1 + \kappa)r}\right] \tag{10}$$

where $\kappa > 0$, $\mu > 0$, and $I_v[.]$ is the modified Bessel function of the first kind and order v. The cumulative distribution function of the κ − μ distribution is defined as

$$F_R(r) = 1 - Q_v\left[\sqrt{2\kappa\mu}, \sqrt{2(1 + \kappa)\mu r}\right] \tag{11}$$

where

$$Q_v(a, b) = \frac{1}{a^{v-1}} \int_b^{\infty} x^v \exp\left(-\frac{x^2+a^2}{2}\right) I_{v-1}[ax]\, dx \tag{12}$$

is the generalized Marcum Q function [9]. The outcomes of the envelope of the kappa-mu distribution are generated using the in-phase and quadrature-phase components as [9]

$$R = \sqrt{\sum_{i=1}^{n}(X_i + p_i)^2 + \sum_{i=1}^{n}(Y_i + q_i)^2} \tag{13}$$

where X_i and Y_i are mutually independent Gaussian processes with $E[X_i] = E[Y_i] = 0$, $E[X_i^2] = E[Y_i^2] = \sigma^2$; p_i and q_i are respectively the mean values of the in-phase and quadrature components of the multipath waves of i^{th} cluster, and n is the number of clusters of multipath. The approximated values of n, p_i, q_i and σ^2 in terms of κ and μ can be written as follows.

$$\begin{aligned} n &= \lceil \mu \rceil \\ p_i &= \sqrt{\frac{\kappa}{2n(1+\kappa)}} \\ q_i &= p_i \\ \sigma^2 &= \frac{1}{2n(1+\kappa)} \end{aligned} \tag{14}$$

The outcomes of the Gaussian processes X_i and Y_i are generated with zero mean and variance σ^2. These generated outcomes and the values of n, p_i, and q_i computed using (14) are used to obtain the envelope R of κ−μ distribution that is defined in (13). For the demonstration purpose of the effectiveness of the proposed techniques, the results are presented for different values of κ and μ. The generated data is divided into train and test data. The train data is used to model the proposed methods to approximate the CDF of the data. The test data is used to check the efficiency of the proposed methods in approximating the CDF of the data and hence the PDF. The performance of the proposed methods is compared with the theoretical values computed using the method proposed by M. D. Yacoub [9], and the estimated values by using the state-of-the-art techniques Kernel Density Estimator (KDE), and Gaussian Mixture Model (GMM).

The training of the estimation methods is performed with 500, 100, and2000 numbers of the samples, and the trained algorithms are tested with 100 samples. The number of mixtures used for the proposed interior-point method is 100 since some of the mixing probabilities becomes 0 while solving the equation in (9). The parameters of PSO used in the Algorithm 4 are M = 30, $c1$ = 0.3, and $c2$ = 1. The parameters of PSO used in the Algorithm 5 are L = 30, $c3$ = 0.7, and $c4$ = 1.

We have showcased our results across various combinations of κ and μ values. These different parameter settings yield distinct probability distributions. Specifically, when κ = 1 and μ = 1, it corresponds to the Rician distribution. Conversely, when κ = 0, it transforms into a Nakagami-m distribution with m equal to μ. Fig.2(a), (b), and (c) shows the approximation of the CDF and PDF of the κ−μ distribution for κ = 10 and μ = 1 with the training samples of 500, 1000, and 2000 respectively. Similarly, Fig.3., Fig.4., and Fig.5. demonstrate the approximation of CDF and PDF with different training samples for κ = 5, μ = 1, κ = 1, μ = 1 and κ = 0.55, μ = 1, respectively. It is evident from Fig.2.- 5. (a) that the proposed methods approximate both the CDF and PDF better than compared to the KDE and GMM even when the number of training samples is less. Also, one can see from Fig.2.- 5. (b) and (c), that, as the number of samples increases the performance of the proposed methods is like that of KDE and GMM. However, the proposed method offers a distinct advantage by providing a closed-form expression for the CDF.

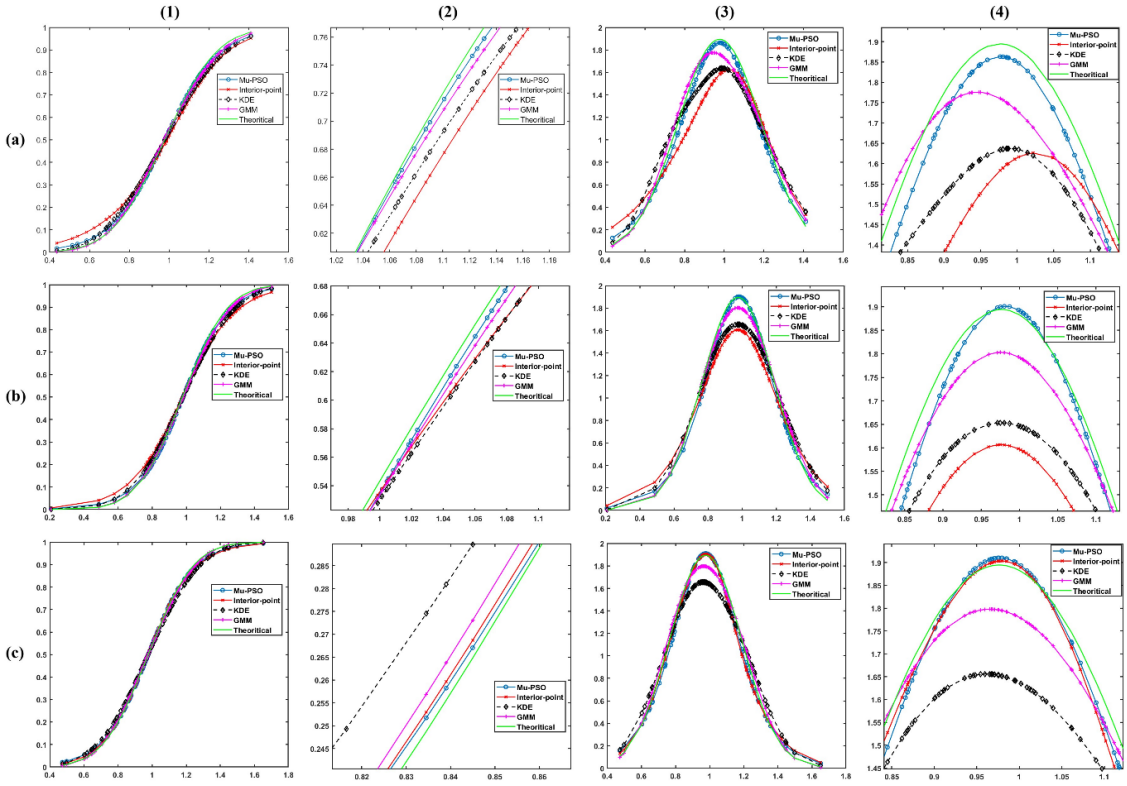

Fig.2. Comparison of the CDF and PDF estimated using different methods for $\kappa = 10$ and $\mu = 1$ with the number of training samples of (a) 500, (b) 1000, and (c) 2000. In each row (1) CDF, (2) CDF zoomed, (3) PDF, and (4) PDF zoomed

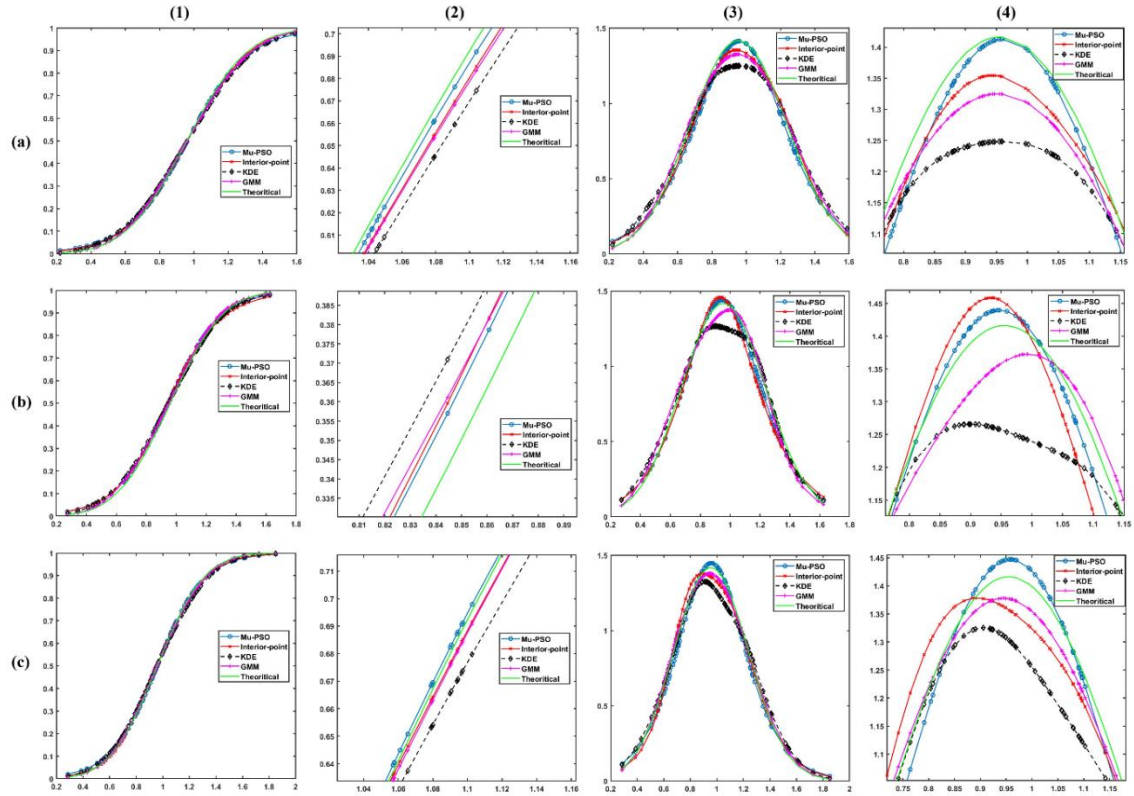

Fig.3. Comparison of the CDF and PDF estimated using different methods for $\kappa = 5$ and $\mu = 1$ with the number of training samples of (a) 500, (b) 1000, and (c) 2000. In each row (1) CDF, (2) CDF zoomed, (3) PDF, and (4) PDF zoomed

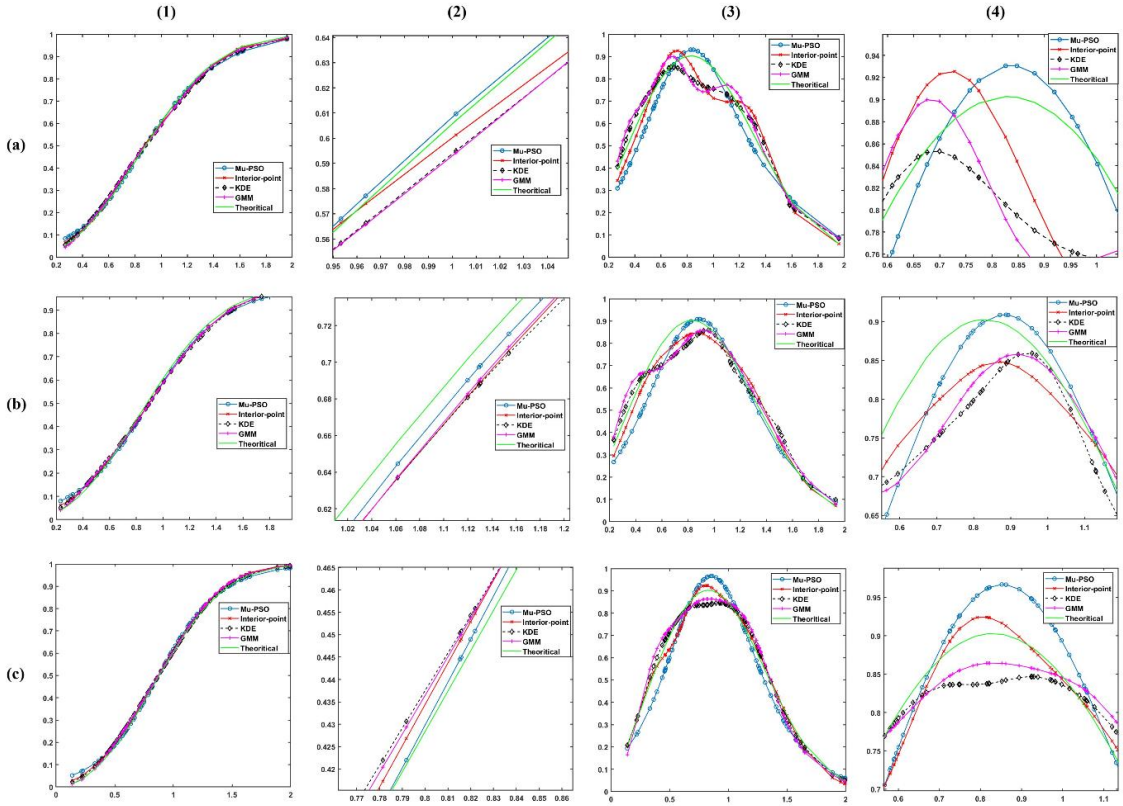

Fig.4. Comparison of the CDF and PDF estimated using different methods for κ = 1 and μ = 1 with the number of training samples of (a) 500, (b) 1000, and (c) 2000. In each row (1) CDF, (2) CDF zoomed, (3) PDF, and (4) PDF zoomed

This closed-form CDF can serve multiple purposes, including calculating outage probabilities and generating data using the inverse of the CDF. The variation in the MSE of the estimated CDF and PDF when compared to the theoretical values with different numbers of mixtures is shown in Fig.6 for $\kappa = 10$ and, $\mu = 1$ with 2000 training samples.

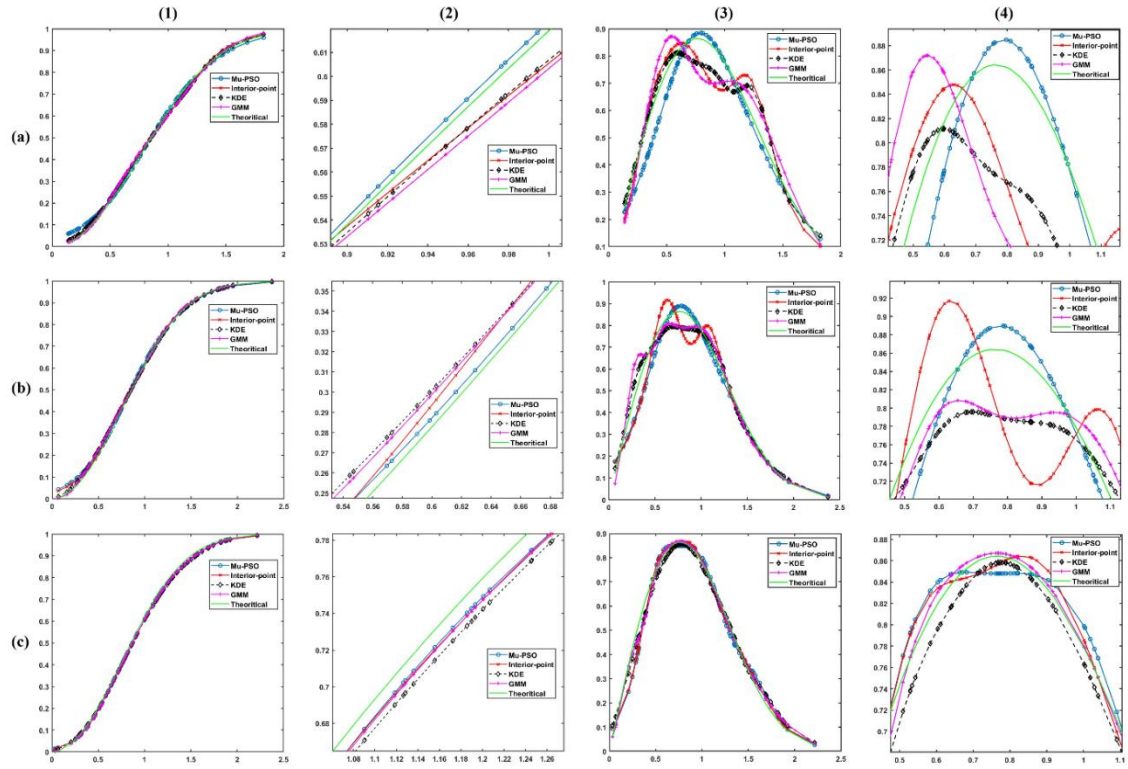

Fig.5. Comparison of the CDF and PDF estimated using different methods for κ = 0.55 and μ = 1 with the number of training samples of (a) 500, (b) 1000, and (c) 2000. In each row (1) CDF, (2) CDF zoomed, (3) PDF, and (4) PDF zoomed

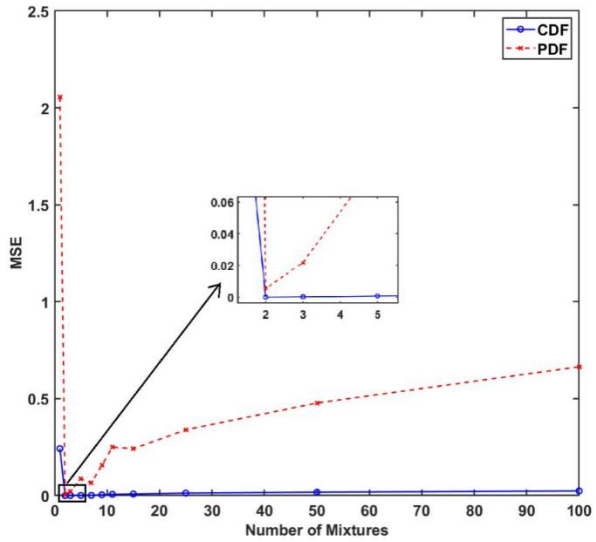

Fig.6. Comparison of the MSE of the estimated CDF and PDF with different numbers of mixtures using Mu-PSO algorithm

The optimum number of mixtures that are needed to estimate the CDF of the $\kappa-\mu$ distribution with κ = 10 and, μ = 1 is 2 because the $\kappa-\mu$ distributed data is generated with two independent and identically distributed Gaussian distributions with different variances when μ = 1. Since the closed-form expression of the CDF approximation is a simple exponential function, the outage probability, P_{out} [1, 2] can be easily computed as

$$P_{out} = P(x < \gamma_{th}) = F_X(\gamma_{th}) = \sum_{k=1}^{K} \pi_k L(\gamma_{th};\ \mu_k, \Sigma_k) \quad (15)$$

The outage probability computed using the proposed methods is compared with the theoretical values proposed in [9]. The outage probability of the κ – μ distribution as a function of the γth (dB) for κ = 0.55 and μ = 1, κ = 5 and μ = 1, κ = 10 and μ = 1 is plotted in Fig.7. 2000 samples of the kappa-mu distribution are used to train the proposed methods in this instance. It can be observed from Fig.7., how close the outage probabilities obtained by the proposed methods are to the theoretical values.

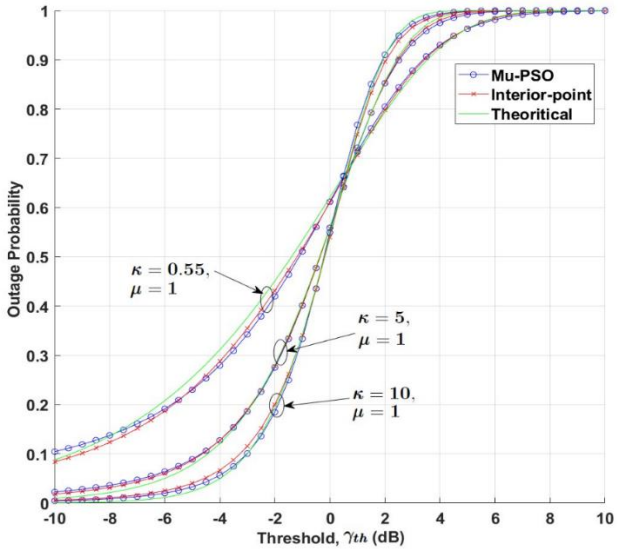

Fig.7. Outage Probability at three different combinations of κ and μ

6. Limitations

Since the interior-point based ELM method takes the parameters of Logistic distribution randomly, it needs a greater number of mixtures to exactly model the LMM in approximating CDF. The Mu-PSO method needs a smaller number of mixtures when compared to interior point method since it tries to obtain all the parameters of LMM using PSO. However, Mu-PSO method takes more time when compared to interior point-based ELM.

7. Conclusions

We have successfully developed closed-form expressions to approximate the CDF and PDF of the κ−μ distribution using LMM. Our comprehensive evaluation reveals that these new approximations outperform current state-of-the-art methods in accurately approximating the CDF and PDF of the κ−μ distribution. Notably, these methods demonstrated impressive accuracy even with a relatively small dataset of just 500 samples.

Furthermore, it's worth highlighting that our presented approaches can be extended beyond univariate data. They can be adapted to estimate the CDF and PDF of multivariate data, offering broader applicability. Additionally, exploring the potential of these techniques to approximate the CDF and PDF of other probability distributions presents an exciting direction for future research.

References

[1] D. Morales-Jimenez and J.F. Paris, "*Outage probability analysis for η − μ fading channels*", IEEE Communications Letters 14 (6) (2010), 521–523.

[2] M. Milisic, M. Hamza and M. Hadzialic, "*Outage Performance of L-Branch Maximal-Ratio Combiner for Generalized κ – μ Fading*", in VTC Spring 2008 - IEEE Vehicular Technology Conference, 2008, pp. 716–731, [10.1109/VETECS.2008.79].

[3] M.D. Renzo, F. Graziosi and F. Santucci, "*On the cumulative distribution function of quadratic-form receivers over generalized fading channels with tone interference*", IEEE Trans. On Communications 57 (7) (2009), 2122–2137.

[4] F.J. Lopez-Martinez, D. Morales-Jimenez, E. Martos-Naya and J.F. Paris, "*On the Bivariate Nakagami-m Cumulative Distribution Function: Closed-Form Expression and Applications*", IEEE Trans. on Communications 61 (4) (2013), 1404–1414.

[5] N.Y. Ermolova and O. Tirkkonen, "*Cumulative Distribution Function of Bivariate Gamma Distribution With Arbitrary Parameters and Applications*", IEEE Communications Letters 19(2) (2015), 167–170.

[6] J.A. Maya, L.R. Vega and C.G. Galarza, "*A Closed-Form Approximation for the CDF of the Sum of Independent Random Variables*", IEEE Signal Processing Letters 24 (1) (2017), 121–125.

[7] W. Kaczynski, L. Leemis, N. Loehr and J. McQueston, "*Nonparametric Random Variate Generation Using a Piecewise Linear Cumulative Distribution Function*", Communications in Statistics - Simulation and Computation 41(4) (2012), 449–468. doi:10.1080/03610918.2011.606947.

[8] W. Kaczynski, L. Leemis, N. Loehr and J. McQueston, "*Bivariate Nonparametric Random Variate Generation Using a Piecewise-Linear Cumulative Distribution Function*", Communications in Statistics - Simulation and Computation 41(4) (2012), 469–496. doi:10.1080/03610918.2011.594532.

[9] M. D. Yacoub, "*The κ−μ distribution and the η−μ distribution*", IEEE Antennas and Propagation Magazine 49 (1) (2007), 68–81.

[10] L. Nicolas and W. Xavier, "Fast multivariate empirical cumulative distribution function with connection to kernel density estimation", Computational Statistics and Data Analysis, 162 (2021), 1-16.

[11] E. Zamanzade, M. Mahdizadeh and H.M. Samawi, "Efficient estimation of cumulative distribution function using moving extreme ranked set sampling with application to reliability", AStA Advances in Statistical Analysis, 104 (2020), 485-502.

[12] C. Cristiano and M. Danilo, "An Extreme Learning Machine Approach to Density Estimation Problems", IEEE Transactions on Cybernetics, 47 (10) (2017), 3254-3265.

[13] M. A. Ibrahim, C. E. Zouaoui, L. Ali and R. Mustapha, "*kNN local linear estimation of the conditional cumulative distribution function: Dependent functional data cas*e", Comptes Rendus Mathematique, 356 (10) (2018), 1036-1039.

[14] P. Puchert, P. Hermosilla, T. Ritschel and T. Ropinski, "Data-driven deep density estimation", Neural Computing and Applications, 33 (2021), 16773-16807.

[15] M. X. Zhu and Y. H. Shao, "*Classification by Estimating the Cumulative Distribution Function for Small Data*", IEEE Access, 11 (2023), 41142-41152.

[16] H. J. Malik and B. Abraham, "*Multivariate Logistic Distributions*", The Annals of Statistics 1 (3) (1973), 588–590.

[17] Z. Q. Shi, T. R. Zheng and J. Q. Han, "*Identifiability of multivariate logistic mixture models*", 2014, Preprint at https://arxiv.org/abs/1208.3546.

[18] Q.Y.Z. G.B. Huang and C.K. Siew, "*Extreme learning machine: Theory and applications*", Neurocomputing 70 (2006), 489–501.

[19] C. Cervellera and D. Macciò, "*An Extreme Learning Machine Approach to Density Estimation Problems*", IEEE Trans. on Cybernetics 47(10) (2017), 3254–3265.

[20] S.J.W. F.A. Potra, "*Interior-point methods*", Journal of Comp. and Appl. Mathematics 124(1) (2000), 281–302.

[21] C. Bishop, "*Mixture models and EM*", in Pattern Recognition and Machine Learning, Springer, New York, 2006, pp. 423–459.

[22] J. Kennedy and R.C. Eberhart, "*Particle swarm optimization*", in IEEE International Conference on Neural Networks, Vol. 4, IEEE Service Center, Piscataway, NJ, 1995, pp. 1942–1948.

[23] E.S. Gopi, "*Artificial Intelligence*", in Algorithm Collections for Digital Signal Processing Applications using Matlab, Springer Dordrecht, Netherlands, 2007, pp. 1–8.

[24] P. Rajasekharreddy and E.S. Gopi, "*Improvement of accuracy of under-performing classifier in decision making using discrete memoryless channel model and Particle Swarm Optimization*", Expert Systems with Applications 213, pp.1-12.

Authors' Profiles

Rajasekharreddy Poreddy received the B.Tech. degree in electronics and communication engineering from G Pulla Reddy Engineering College, Kurnool, Andhra Pradesh, India, and the M.Tech. degree in communication systems from National Institute of Technology Tiruchirappalli, Tamil Nadu, India. He is currently pursuing Ph.D. degree in computational intelligence for pattern recognition and applications in the Department of Electronics and Communication Engineering, National Institute of Technology Tiruchirappalli, Tamil Nadu, India. His research interests include pattern recognition, computational intelligence, machine learning and deep learning.

Dr. E. S. Gopi received his Ph.D. degree from the National Institute of Technology Tiruchirappalli (NITT). He is currently an Associate Professor, coordinator, and head of the Pattern Recognition and Computational Intelligence Laboratory at NITT. Dr. Gopi is the author of seven books and more than 50 journal articles and conference papers in the area of pattern recognition, machine learning, deep learning, computational intelligence, and digital signal processing. He is currently an IEEE senior member of the Computational Intelligence Society and the Signal Processing Society. He is one of the editors for the book series signals and communication technology, Springer publications. He is also serving as one of the officers for Emerging Technology Initiative Machine Learning for Communications Emerging Technology Initiative of IEEE Communications Society.

How to cite this paper: Rajasekharreddy Poreddy, Gopi E. S., "Data-driven Approximation of Cumulative Distribution Function Using Particle Swarm Optimization based Finite Mixtures of Logistic Distribution", International Journal of Intelligent Systems and Applications(IJISA), Vol.16, No.5, pp.10-21, 2024. DOI:10.5815/ijisa.2024.05.02