SELF-PACED MIXTURE OF T DISTRIBUTION MODEL

Yang Zhang[†,*] *, Qingtao Tang*[†,*]*, Li Niu* [‡]*,Tao Dai*[†]*, Xi Xiao*[†]*, Shu-Tao Xia*[†]

[†] Department of Computer Science and Technology, Tsinghua University, China
[‡] Department of Electrical and Computer Engineering, Rice University, U.S.

ABSTRACT

Gaussian mixture model (GMM) is a powerful probabilistic model for representing the probability distribution of observations in the population. However, the fitness of Gaussian mixture model can be significantly degraded when the data contain a certain amount of outliers. Although there are certain variants of GMM (*e.g.*, mixture of Laplace, mixture of t distribution) attempting to handle outliers, none of them can sufficiently mitigate the effect of outliers if the outliers are far from the centroids. Aiming to remove the effect of outliers further, this paper introduces a Self-Paced Learning mechanism into mixture of t distribution, which leads to Self-Paced Mixture of t distribution model (SPTMM). We derive an Expectation-Maximization based algorithm to train SPTMM and show SPTMM is able to screen the outliers. To demonstrate the effectiveness of SPTMM, we apply the model to density estimation and clustering. Finally, the results indicate that SPTMM outperforms other methods, especially on the data with outliers.

Index Terms— Self-Paced Learning, Robustness, Gaussian Mixture Model, Mixture of t distribution

1. INTRODUCTION

Gaussian Mixture Model (GMM) [1] is a powerful probabilistic model for representing a population consisting of several subpopulations. Due to its satisfactory flexibility, good interpretability, and simple parameter learning, GMM has been widely used in many fields, including data mining, pattern recognition [2], machine learning, and statistical analysis.

In GMM, the basic model is $p(\boldsymbol{x}) = \sum_{j=1}^{g} \pi_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, where each Gaussian density $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ is a component of the mixture and has its own mean $\boldsymbol{\mu}_j$ and covariance $\boldsymbol{\Sigma}_j$, and the parameters $\pi_j \geq 0$ are mixing coefficients satisfying $\sum_{j=1}^{g} \pi_j = 1$. However, due to the thin-tailed property of the Gaussian distribution, GMM may perform poorly on the data which contain a group or groups of observations with heavy tails or outliers [3].

In order to handle the data with heavy tails or outliers, heavy-tailed distributions have been introduced into mixture model. For instance, Laplace distribution has heavier tails than Gaussian distribution, and t distribution is a robust generalization of Gaussian distribution [4–6]. In particular, mixture of Laplace distribution is proposed by [7] while Peel *et al.* [3] introduced mixture of t distribution (TMM) with each component assumed to be a t distribution. These mixture models alleviate the influence of outliers to a certain degree.

Unfortunately, the aforementioned approaches cannot achieve satisfactory performance when the outliers stay distant from the centroids [8]. The main reason is that these methods focus on alleviating the influence of outliers based on heavy-tailed distribution while the outliers are still included in the learning procedure. With that reason, these methods cannot remove the effect of outliers adequately. In fact, the performance of these models can be significantly compromised when the outliers are far from the centroids.

To further reduce the influence of outliers, we introduce the Self-Paced Learning (SPL) [9] mechanism into mixture model, and develop a novel method named Self-Paced Mixture of t distribution model (SPTMM). Analogous to human learning procedure (*e.g.*, a pupil is supposed to understand elementary algebra before he/she can move on to advanced algebra topic), SPL starts with the easiest samples and then gradually includes harder samples while keeping the outliers off the learning procedure. In the last few years, the effectiveness of SPL has been validated in many tasks, such as event detection [10], co-saliency detection [11] and mixture of regression [12]. In this paper, we apply SPL to address the outliers in mixture model by screening the outliers during the learning procedure, which has never been explored before.

In summary, this paper presents three major contributions:

- This is the first work of employing Self-Paced Learning (SPL) to mixture model[1], with the aim to effectively remove the influence of outliers.
- We propose our SPTMM method which integrates SPL with TMM, and develop an EM based algorithm to solve the corresponding optimization problem.
- Extensive experiments demonstrate the superiority of our SPTMM method for density estimation and clustering.

[*]Equal Contribution. (Corresponding author: Qingtao Tang.)

[1]Here, we adopt the definition of mixture model in statistics, which corresponds to the mixture distribution [13].

2. MIXTURE OF T DISTRIBUTION

The t distribution is defined as follows.

Definition 1. A p-dim random vector $\boldsymbol{x} \in \mathbb{R}^p$ follows the p-variate t distribution with degrees of freedom $\nu \in \mathbb{R}_+$, mean $\boldsymbol{\mu} \in \mathbb{R}^n$, and correlation matrix $\boldsymbol{\Sigma} \in \Pi(p)$ if its joint probability density function (PDF) is given by

$$t\left(\boldsymbol{x}|\nu,\boldsymbol{\mu},\boldsymbol{\Sigma}\right) = \frac{\Gamma[(\nu+p)/2]}{\Gamma(\nu/2)\nu^{p/2}\pi^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \cdot \left[1+\frac{1}{\nu}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]^{-\frac{\nu+p}{2}}. \quad (1)$$

The mixture of t distribution model (TMM) is a linear superposition of g-component t distribution, *i.e.*,

$$\phi(\boldsymbol{x};\boldsymbol{\Psi}) = \sum_{j=1}^{g} \pi_j t(\boldsymbol{x};\nu_j,\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j),$$

where π_j is the mixing coefficient of the j-th component and $\boldsymbol{\Psi} = \{\boldsymbol{\pi},\boldsymbol{\nu},\boldsymbol{\mu},\boldsymbol{\Sigma}\}$, in which $\boldsymbol{\pi} = (\pi_1,\pi_2,\ldots,\pi_g)^{\mathrm{T}}$, $\boldsymbol{\nu} = (\nu_1,\nu_2,\ldots,\nu_g)^{\mathrm{T}}$, $\boldsymbol{\mu} = (\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\ldots,\boldsymbol{\mu}_g)$, and $\boldsymbol{\Sigma} = (\boldsymbol{\Sigma}_1,\boldsymbol{\Sigma}_2,\ldots,\boldsymbol{\Sigma}_g)$.

Given the dataset $\mathcal{D} = \{\boldsymbol{x}_i|i=1,2,\ldots,n\}$, where $\boldsymbol{x}_i \in \mathbb{R}^p$ denotes a p-dim sample, the model parameters of TMM $\boldsymbol{\Psi}$ can be estimated by minimum the negative log likelihood, *i.e.*,

$$\min_{\boldsymbol{\Psi}} -\sum_{i=1}^{n}\log\sum_{j=1}^{g}\pi_j t(\boldsymbol{x}_i;\nu_j,\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j), \quad (2)$$

which can be solved by EM algorithm [3].

3. SELF-PACED MIXTURE OF T DISTRIBUTION

3.1. Introduction of Our model

Compared with GMM, TMM is able to mitigate the defect of outliers and thus is more robust to outliers, but it cannot remove the influence completely [8]. In fact, TMM is still prone to outliers, especially when the outliers are not close to the major data samples. To further reduce the influence of outliers, we introduce the Self-Paced Learning to TMM, leading to our Self-Paced Learning mixture of t distribution model (SPTMM). Specifically, based on (2), we introduce a latent binary variable v_i to indicate whether the sample $\boldsymbol{x}_i$ is an outlier, and add a sparse regularizer of v_i:

$$E(\boldsymbol{v};\boldsymbol{\Psi},\lambda) = -\sum_{i=1}^{n} v_i \log\sum_{j=1}^{g}\pi_j t(\boldsymbol{x}_i;\nu_j,\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j) - \lambda||\boldsymbol{v}||_1, \quad (3)$$

where λ is a hyper-parameter, $\boldsymbol{v} = (v_1,\ldots,v_n)^T$ is a binary outlier indicator with $v_i \in \{0,1\}$, and $||\boldsymbol{v}||_1$ enforces the sparsity of $\boldsymbol{v}$ since there exist only a few outliers in the training samples. We expect $v_i = 0$ if $\boldsymbol{x}_i$ is an outlier and $v_i = 1$ otherwise, so that only clean samples contribute to the learning procedure. Thus, we also refer to v_i as learning weight for $\boldsymbol{x}_i$. Note that $\boldsymbol{\Psi}$ and $\boldsymbol{v}$ are unknown and need to be learnt iteratively.

One interesting observation is that when $\boldsymbol{\Psi}$ are fixed, $\boldsymbol{v}$ can be easily learnt by using λ as threshold. Specifically, when $-\log\sum_{j=1}^{g}\pi_j\, t(\boldsymbol{x};\nu_j,\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)$ is larger than λ, the learnt v_i will be 0 (see Section 3.2.1 for technical derivation) and thus the corresponding $\boldsymbol{x}_i$ will make no contribution when updating $\boldsymbol{\Psi}$ next time. This meets our expectation because the outliers are usually far away from the mean of distributions with large negative log likelihood $-\log\sum_{j=1}^{g}\pi_j t(\boldsymbol{x};\nu_j,\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)$.

3.2. EM algorithm for SPTMM

Inspired by Self-Paced Learning (SPL), we tend to start the learning procedure with the easiest samples and then gradually use harder samples while preventing outliers from being used. Hence, in our learning procedure, we initialize λ to a small value and enlarge λ gradually, until the learnt model parameters $\boldsymbol{\Psi}$ remain unchanged, which is shown in Algorithm 1. With each fixed λ, we optimize (3) over $\boldsymbol{v}$ and $\boldsymbol{\Psi}$ iteratively until the objective in (3) converges. In particular, in each iteration with fixed λ, we update $\boldsymbol{v}$ when fixing $\boldsymbol{\Psi}$, and then update $\boldsymbol{\Psi}$ when fixing $\boldsymbol{v}$. In the following, we elaborate on the optimization *w.r.t.* $\boldsymbol{v}$ and $\boldsymbol{\Psi}$ in detail.

3.2.1. Optimization of $\boldsymbol{v}$

When the model parameters $\boldsymbol{\Psi}$ are fixed, we estimate $\boldsymbol{v}$ by solving the following problem:

$$\min_{\boldsymbol{v}\in\{0,1\}} E(\boldsymbol{v};\boldsymbol{\Psi},\lambda) = \sum_{i=1}^{n} v_i\ell_i - \lambda||\boldsymbol{v}||_1, \quad (4)$$

where $\ell_i = -\log\phi(\boldsymbol{x}_i;\boldsymbol{\Psi}) = -\log\sum_{j=1}^{g}\pi_j t(\boldsymbol{x}_i;\nu_j,\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)$. Considering $v_i \in \{0,1\}$, the problem in (4) can be written as

$$\min_{\boldsymbol{v}\in\{0,1\}} \sum_{i=1}^{n} v_i(\ell_i - \lambda). \quad (5)$$

It is obvious that the solution to (5) is

$$v_i = \begin{cases} 0 & \ell_i > \lambda, \\ 1 & \ell_i \le \lambda. \end{cases} \quad (6)$$

It is worth noting that the training samples $\boldsymbol{x}_i$ with larger ℓ_i are more likely to be outliers. To explain more, ℓ_i is an increasing function of $(\boldsymbol{x}_i-\boldsymbol{\mu}_j)^{\mathrm{T}}{\boldsymbol{\Sigma}_j}^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu}_j)$ (see (1)), which is the Mahalanobis distance between sample $\boldsymbol{x}_i$ and component centroid $\boldsymbol{\mu}_j$. Therefore, ℓ_i can be used to measure the distance between sample $\boldsymbol{x}_i$ and component centroids. As outliers are far from the component centroids, the training sample $\boldsymbol{x}_i$ with larger ℓ_i is more likely to be an outlier. According to (6), the

learning weight v_i of outliers would be set to 0, which can allow us to use only clean samples for learning model parameters $\mathbf{\Psi}$.

3.2.2. Optimization of $\mathbf{\Psi}$

Fixing $\boldsymbol{v}$, we estimate $\mathbf{\Psi}$ by minimizing the negative log marginal likelihood (3):

$$\min_{\mathbf{\Psi}} E(\mathbf{\Psi}; \lambda, \boldsymbol{v}) \Leftrightarrow \min_{\mathbf{\Psi}} \sum_{i=1}^{n} v_i \ell_i - \lambda ||\boldsymbol{v}||_1 \Leftrightarrow \min_{\mathbf{\Psi}} \sum_{i=1}^{n} v_i \ell_i. \quad (7)$$

As $v_i \in \{0, 1\}$, (7) is the same optimization problem as (2). Thus, a similar EM algorithm can be applied to solve (7). For parameters $\mathbf{\Psi} = \{\boldsymbol{\pi}, \boldsymbol{\nu}, \boldsymbol{\mu}, \mathbf{\Sigma}\}$, we set the degrees of freedom ν_j $(j = 1, 2, \ldots, g)$ to 4 following the practice in [14] and update $\{\boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{\Sigma}\}$ using the following equations:

$$\pi_j^\star = \frac{1}{n} \sum_{i=1}^{n} v_i \hat{\tau}_{ij}^\star, \quad (8)$$

$$\boldsymbol{\mu}_j^\star = \frac{\sum_{i=1}^{n} v_i \hat{\tau}_{ij}^\star \hat{u}_{ij}^\star \boldsymbol{x}_i}{\sum_{i=1}^{n} v_i \hat{\tau}_{ij}^\star \hat{u}_{ij}^\star}, \quad (9)$$

$$\mathbf{\Sigma}_j^\star = \frac{\sum_{i=1}^{n} v_i \hat{\tau}_{ij}^\star \hat{u}_{ij}^\star (\boldsymbol{x}_i - \boldsymbol{\mu}_j^\star)(\boldsymbol{x}_i - \boldsymbol{\mu}_j^\star)^{\mathrm{T}}}{\sum_{i=1}^{n} v_i \hat{\tau}_{ij}^\star}, \quad (10)$$

where

$$\hat{\tau}_{ij}^\star = \frac{\pi_i t(\boldsymbol{x}_i; \nu_j, \boldsymbol{\mu}_j, \mathbf{\Sigma}_j)}{\sum_{j=1}^{g} \pi_j t(\boldsymbol{x}_i; \nu_j, \boldsymbol{\mu}_j, \mathbf{\Sigma}_j)}, \quad (11)$$

$$\hat{u}_{ij}^\star = \frac{\nu_j + p}{\nu_j + (\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}} {\mathbf{\Sigma}_j}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)}. \quad (12)$$

The definitions of $\hat{\tau}$ and $\hat{\tau}$ are given by Peel *et al.* [3]. Note that the above two sets of variables $\{\pi_j^\star, \boldsymbol{\mu}_j^\star, \mathbf{\Sigma}_j^\star\}$ and $\{\hat{\tau}_{ij}^\star, \hat{u}_{ij}^\star\}$ are updated in an iterative fashion based on EM algorithm. More precisely, we update $\{\hat{\tau}_{ij}^\star, \hat{u}_{ij}^\star\}$ in E-step and update $\{\pi_j^\star, \boldsymbol{\mu}_j^\star, \mathbf{\Sigma}_j^\star\}$ in M-step.

3.2.3. Summary of the Algorithm

The whole optimization algorithm is summarized in Algorithm 1, which has an outer loop and an inner loop. In the outer loop, we start with small λ and increase λ by $\lambda \leftarrow a\lambda$ in each iteration until the model parameters $\mathbf{\Psi}$ remain unchanged in several steps , which follows the strategy of SPL. In the inner loop, we update the binary outlier indicator $\boldsymbol{v}$ and the model parameters $\mathbf{\Psi}$ iteratively until the objective in (3) converges.

Algorithm 1: Self-paced Mixture of t distribution

Input: dataset $\mathcal{D} = \{\boldsymbol{x}_i, i = 1, 2, \ldots, n\}$, learning pace $a > 1$, number of components g
Output: $\mathbf{\Psi}_j (j = 1, 2, \ldots, g)$
Initialize $\mathbf{\Psi}$ by the result of k-means.
Initialize λ to the median of $\ell_i, i = 1, 2, \ldots, n$.
while $\Delta\mathbf{\Psi} \simeq \mathbf{0}$ **do**
  while *not converged* **do**
    Update $\boldsymbol{v} = \arg\max_{\boldsymbol{v} \in \{0,1\}} E(\boldsymbol{v}; \mathbf{\Psi}, \lambda)$ by (6).
    while *not converged* **do**
      E-step
      Update $\hat{\tau}_{ij}$ and $\hat{u}_{ij}$ by (11) and (12), respectively.
      M-step
      Update $\mathbf{\Psi}$ by (8), (9) and (10).
    end
  end
  $\lambda \leftarrow a\lambda$.
end

4. EXPERIMENTS

In this section, we evaluate our SPTMM model on two tasks: density estimation and clustering. For density estimation, we use a synthetic dataset, and compare our SPTMM method with GMM and TMM. For clustering, we use real-world datasets and additionally include k-means algorithm as a baseline besides GMM and TMM.

4.1. Experimental results in density estimation

This section is devoted to comparing GMM, TMM and SPTMM *w.r.t.* estimating the density of a synthetic dataset.

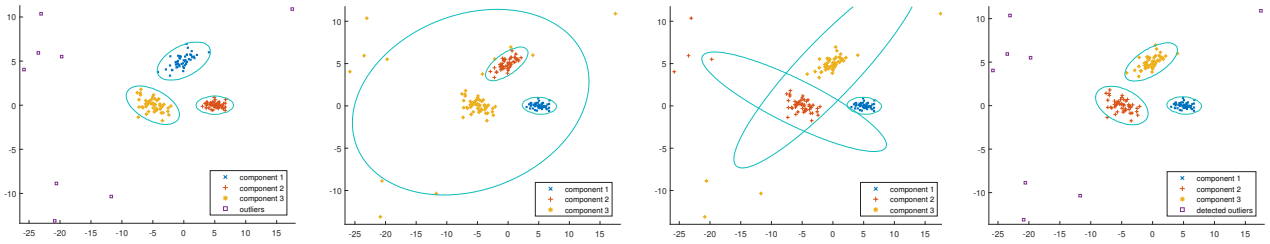
Firstly, a synthetic dataset is generated with a 3-component GMM. To evaluate the robustness of GMM, TMM and SPTMM to outliers, we add 5% outliers[2] to the synthetic dataset. The synthetic dataset is shown in **Fig.** 1 (a).

The experimental results of GMM, TMM and SPTMM on the synthetic dataset are illustrated in **Fig.** 1 (b) (c) (d), respectively. It is obvious that the covariance matrix of GMM is severely influenced by the outliers. Although TMM reduces the influence to a degree, the covariance of TMM is still affected heavily. In contrast, our SPTMM model gives much better estimation for the covariance matrix, since all the identified outliers are not used in the fitting procedure.

4.2. Experimental results in clustering

In this section, we evaluate the performance of our SPTMM model and other baselines, including k-means, GMM and TMM, for clustering on 4 real-world datasets, *i.e.*, Bezdekiris, Iris, Seeds and Thyroid [16]. Besides, to validate the robustness of these algorithms, we add 5% outliers to each real dataset. The real datasets without and with 5% outliers are referred to as Clean and Noisy respectively in Table 1.

[2]In this paper, we adopt the definition of outliers in [15], *i.e.*, $[Q_1 - \alpha(Q_3 - Q_1), Q_3 + \alpha(Q_3 - Q_1)]$.



(a) Synthetic dataset (b) GMM (c) TMM (d) SPTMM

Fig. 1: (a) is Synthetic dataset with outliers. (b), (c) and (d) are fitting results of GMM, TMM, SPTMM, respectively. The colored lines represent 99% confidence ellipses.

(a) Bezdekiris

	Clean				Noisy			
	MSE	W/B	DB	Dunn	MSE	W/B	DB	Dunn
k-means	**0.93**	**0.30**	0.62	2.03	2.48	0.75	0.53	1.73
GMM	1.12	0.39	0.74	1.88	2.53	0.78	0.84	0.93
TMM	1.43	0.61	**0.29**	1.94	2.66	0.88	0.59	1.67
SPTMM	1.51	0.65	**0.29**	**2.78**	**1.51**	**0.61**	**0.26**	**3.69**

(b) Iris

	Clean				Noisy			
	MSE	W/B	DB	Dunn	MSE	W/B	DB	Dunn
k-means	**0.93**	**0.31**	0.57	2.03	2.48	0.72	0.54	1.76
GMM	1.13	0.40	0.74	1.88	2.95	0.98	1.12	1.03
TMM	1.45	0.62	**0.28**	1.95	2.92	1.07	0.73	1.18
SPTMM	1.45	0.62	**0.28**	**2.68**	**1.45**	**0.62**	**0.28**	**2.89**

(c) Seeds

	Clean				Noisy			
	MSE	W/B	DB	Dunn	MSE	W/B	DB	Dunn
k-means	2.80	**0.28**	**0.59**	2.35	3.55	0.69	0.72	2.06
GMM	2.87	**0.28**	0.60	**2.37**	4.67	1.16	1.81	0.63
TMM	2.93	0.29	0.61	**2.37**	4.66	0.81	1.41	0.89
SPTMM	**2.55**	0.29	0.61	**2.37**	**2.94**	**0.53**	**0.34**	**2.30**

(d) Thyroid

	Clean				Noisy			
	MSE	W/B	DB	Dunn	MSE	W/B	DB	Dunn
k-means	3.44	2.23	0.69	2.66	3.64	2.50	0.72	2.61
GMM	3.54	2.48	0.59	3.34	3.75	2.79	1.14	1.35
TMM	3.62	2.74	0.57	3.33	3.80	3.02	0.58	3.46
SPTMM	**0.98**	**0.99**	**0.43**	**3.80**	**1.67**	**1.05**	**0.43**	**3.66**

Table 1: MSE, W/B, DBI and Dunn of the clustering results on real datasets.

We evaluate the clustering performance of all these algorithms with internal clustering evaluation metrics [17], including clustering mean squared error (MSE) [18], $\frac{WSS}{BSS}$ (WSS: within-cluster sum of squares; BSS: between-cluster sum of squares) [19], Davies-Bouldin index (DB) [20] and Dunn index (Dunn) [21]. Note that for MSE, W/B, DB, smaller value indicates better performance while for Dunn, larger value indicates better performance.

Clustering results reported in Table 1, show that on the Clean Bezdekiris, Iris and Seeds datasets, our SPTMM method performs best with half of the evaluation metrics and on the Clean Thyroid dataset, SPTMM achieves the best performance with all the evaluation metrics. Besides, we can observe that on all Noisy datasets, SPTMM significantly outperforms all the baselines with all the evaluation metrics. This attributes to that SPTMM can eliminate most outliers from the fitting procedure.

5. CONCLUSION

In this paper, we depicted a novel model SPTMM which integrates the Self-Paced Learning mechanism into mixture of t distribution, in order to improve the mixture models' ability of handling outliers. Given the model, we developed an EM based algorithm that can solve the optimization problem in SPTMM efficiently. In addition to the mathematical justification, the experiments also display the value of the model. The results demonstrated that SPTMM clearly outperforms K-means, GMM and TMM for estimating the covariance matrix in the distributions. With respect to clustering, SPTMM is shown to be the best performer in most cases, in particular for the data with outliers. In the future, we would like to assess if SPTMM can be improved to perform better in a clean environment.

6. ACKNOWLEDGMENTS

We thank Doctor Weipeng Huang, Insight Centre for Data Analytics, University College Dublin, for comments that greatly improved the manuscript. This work is supported by the National Natural Science Foundation of China under grant Nos. 61371078, 61771273, and the R&D Program of Shenzhen under grant Nos. JCYJ20140509172959977, JSGG20150512162853495, ZDSYS20140509172959989, JCYJ20160331184440545.

7. REFERENCES

[1] Geoffrey J Mclachlan and Kaye E Basford, "Mixture models. inference and applications to clustering," *Applied Statistics*, vol. 38, no. 2, 1988.

[2] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert, "Robust monte carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2001.

[3] David Peel and Geoffrey J Mclachlan, "Robust mixture modelling using the t distribution," *Statistics and Computing*, vol. 10, no. 4, pp. 339–348, 2000.

[4] Qingtao Tang, Li Niu, Yisen Wang, Tao Dai, Wangpeng An, Jianfei Cai, and Shu-Tao Xia, "Student-t process regression with student-t likelihood," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, 2017, pp. 2822–2828.

[5] Qingtao Tang, Tao Dai, Li Niu, Yisen Wang, Shu-Tao Xia, and Jianfei Cai, "Robust survey aggregation with student-t distribution and sparse representation," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, 2017, pp. 2829–2835.

[6] Qingtao Tang, Yisen Wang, and Shu-Tao Xia, "Student-t process regression with dependent student-t noise," in *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, 2016, pp. 82–89.

[7] Debjani Bhowmick, Ac Davison, Darlene R Goldstein, and Yann Ruffieux, "A laplace mixture model for identification of differential expression in microarray experiments," *Biostatistics*, vol. 7, no. 4, pp. 630, 2006.

[8] G. J. Mclachlan, S K Ng, and R. W. Bean, "Robust cluster analysis via mixture models," *Austrian Journal of Statistics*, vol. 35, pp. 157–174, 2006.

[9] M. Pawan Kumar, Benjamin Packer, and Daphne Koller, "Self-paced learning for latent variable models," in *International Conference on Neural Information Processing Systems*, 2010, pp. 1189–1197.

[10] Jiang Lu, Deyu Meng, Teruko Mitamura, and Alexander G. Hauptmann, "Easy samples first: Self-paced reranking for zero-example multimedia search," in *ACM International Conference on Multimedia*, 2014, pp. 547–556.

[11] D. Zhang, D. Meng, and J. Han, "Co-saliency detection via a self-paced multiple-instance learning framework," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 5, pp. 865–878, 2016.

[12] Longfei Han, Dingwen Zhang, Dong Huang, Xiaojun Chang, Jun Ren, Senlin Luo, and Junwei Han, "Self-paced mixture of regressions," in *Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017, pp. 1816–1822.

[13] Bruce G. Lindsay, *Mixture Models: Theory, Geometry, and Applications*, Institute of Mathematical Statistics , American Statistical Association, 1995.

[14] Jarno Vanhatalo, Pasi Jylänki, and Aki Vehtari, "Gaussian process regression with student-t likelihood," in *Annual Conference on Neural Information Processing Systems*, 2009, pp. 1910–1918.

[15] J W Tuckey, *Exploratory Data Analysis*, Addison-Wesley Pub. Co., 1977.

[16] M. Lichman, "UCI machine learning repository," 2013.

[17] Yanchi Liu, Zhongmou Li, Hui Xiong, Xuedong Gao, and Junjie Wu, "Understanding of internal clustering validation measures," in *IEEE International Conference on Data Mining*, 2011, pp. 911–916.

[18] Pang Ning Tan, Michael Steinbach, and Vipin Kumar, *Introduction to Data Mining, (First Edition)*, Addison-Wesley Longman Publishing Co., Inc., 2005.

[19] Hans-Peter Kriegel, Erich Schubert, and Arthur Zimek, "The (black) art of runtime evaluation: Are we comparing algorithms or implementations?," *Knowledge and Information Systems*, vol. 52, no. 2, pp. 341–378, Aug 2017.

[20] David L. Davies and Donald W. Bouldin, *A Cluster Separation Measure*, IEEE Computer Society, 1979.

[21] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1974.