Gaussian Mixture Estimation from Weighted Samples

Daniel Frisch and **Uwe D. Hanebeck**

***Abstract*— We consider estimating the parameters of a Gaussian mixture density with a given number of components best representing a given set of weighted samples. We adopt a density interpretation of the samples by viewing them as a discrete Dirac mixture density over a continuous domain with weighted components. Hence, Gaussian mixture fitting is viewed as density re-approximation. In order to speed up computation, an expectation–maximization method is proposed that properly considers not only the sample locations, but also the corresponding weights. It is shown that methods from literature do not treat the weights correctly, resulting in wrong estimates. This is demonstrated with simple counterexamples. The proposed method works in any number of dimensions with the same computational load as standard Gaussian mixture estimators for unweighted samples.**

I. Introduction

Gaussian mixture (GM) estimation is ubiquitous in signal processing and machine learning. Given a set of samples, the parameters of a GM are determined in such a way as to best fit the samples in a maximum likelihood way. Solutions for equally weighted samples are readily available, expectation–maximization (EM) based methods being the most prevalent because of low computational requirements and ease of implementation.

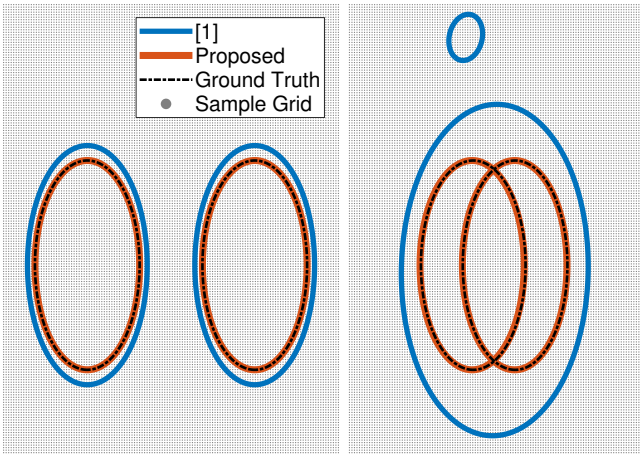
So it comes as a surprise that GM estimation for *weighted samples* is hard to find in literature. It might be even more surprising that the standard reference [1] gives incorrect results, see Fig. 1.

II. Context

Applications for sample-to-density function approximation include clustering of unlabeled data [2], [3], multi-target tracking [4], [5], group tracking [6], multilateration [7], [8], and arbitrary density representation in nonlinear filters [9], [10].

Why may data sets have observational weights associated to their data? Sample weights are used in information fusion methods, especially in the update step of nonlinear filters. Unweighted prior samples are, according to the Bayes rule, re-weighted according to the likelihood function [10]. Thereby the prior information, encoded in sample locations, is fused with the measurement, encoded in the new weights. Cluster analysis in data mining applications does not require an accurate density representation reflecting the data set, but instead to identify and localize all clusters, whether they are represented with many or few samples in the data set.

Daniel Frisch and Uwe D. Hanebeck are with the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany {daniel.frisch, uwe.hanebeck}@ieee.org.



(a) "separated" components (b) "overlapping" components

Fig. 1: Two-dimensional GM parameter estimation using EM from [1] (blue line), and EM according to our proposed method (red line). Compare the ground truth (black line). Equidistant samples (grey dots) were weighted with the GM density function and given to the EM algorithms.

Cluster sizes are sometimes distributed according to the Zipf distribution, i.e., very big and very small clusters are both commonly present. One method to equalize such an imbalanced data set is to discard samples from big clusters (undersampling) and/or generate new samples related to small clusters (oversampling) [11], but the more elegant way is to introduce weights. This is called density-biased sampling [12], [13]. A similar concept is "boosting" from machine learning. Mis-classified samples are assigned higher weights, and after a new training step, the performance of the classificator is improved [14]. Conversely, scattered samples that probably represent noise can be penalized by down-weighting, using prior knowledge [15]. Finally, sample weights may simply be natural numbers stating how many instances of the respective sample were counted [16].

A popular basic solution to this is the k-means algorithm. It does not find a complete density representation, only the means of the individual clusters. The k-means algorithm uses hard sample-to-mean associations, therefore yields merely approximate solutions but can be computationally optimized using k-d trees [17], [18]. Moreover, the global optimum can be found deterministically [19], therefore it can be used to provide an initial guess for more elaborate algorithms. The

k-means algorithm is already available for weighted data [13], [16].

A sample-to-density approximation that is optimal in a maximum likelihood sense can be searched with numerical optimization techniques such as the Newton algorithm that has quadratic convergence but high computational demand per iteration, quasi-Newton methods, the method of scoring, or the conjugate gradient method with slower convergence but less computational effort per iteration [20]. Finally, there is the EM algorithm, which we will focus on in this work.

A. State-of-the-art

The EM algorithm has been used for decades [21], [22] to solve statistical problems. It provides a valid parameter set in every iteration step, i.e., nonnegative and normalized component weights and positive semidefinite covariance matrices, without the need of artificial safeguards. The EM algorithm features good global convergence to some local optimum, is very easy to implement, has low computational cost per iteration when using optimized libraries for standard statistical tasks, and needs little storage [20].

On the downside, EM converges rather slowly, especially if the GM components are poorly separated. Furthermore, for GMs, EM merely finds local optima. Thus, in order to obtain a good result, several optimizations with different initial conditions should be performed. While covariance matrices are guaranteed to be positive semidefinite in every intermediate step, there should be some logic in place that deals with singular covariance matrices.

Extended versions of the EM can automatically determine the optimal number of Gaussian components [23], [24], [25], [26]. In this work we will introduce the basic method and assume the desired number of Gaussian components is given and fixed. However, an extension that includes automatic detection of an optimal number of components similar to one of these methods is possible.

B. Contribution

The contribution of this paper is a fast, simple, and practical EM method for the correct treatment of weighted samples in Gaussian mixture estimation.

III. Problem Formulation

For an observed set of L weighted samples

$$\mathbf{Y} = \{\{\alpha_1, \underline{s}_1\}, \{\alpha_2, \underline{s}_2\}, \ldots, \{\alpha_L, \underline{s}_L\}\}$$

with sample locations $\underline{s}_i$ as vectors in the D-dimensional Euclidean space $\mathbb{R}^D$, and scalar weights α_i, we want to find a GM density function with M Gaussian components

$$f(\underline{x}|\mathbf{\Theta}) = \sum_{m=1}^{M} w_m \, \mathcal{N}\Big(\underline{x} - \underline{\mu}_m,\ \mathbf{C}_m\Big) \quad ,$$

$$\mathcal{N}\big(\underline{x} - \underline{\mu},\ \mathbf{C}\big) = \frac{\exp\Big\{-\frac{1}{2}\left(\underline{x} - \underline{\mu}\right)^{\top} \mathbf{C}^{-1} \left(\underline{x} - \underline{\mu}\right)\Big\}}{\sqrt{|2\pi\mathbf{C}|}} \quad ,$$

with nonnegative component weights $w_m \geq 0$ that are normalized, $\sum_{m=1}^{M} w_m = 1$, component means $\underline{\mu}_m \in \mathbb{R}^D$, and positive definite component covariances $\mathbf{C}_m \in \mathbb{R}^{D \times D}$. The GM should explain the observed samples as good as possible. We thus estimate GM parameters $\mathbf{\Theta}$

$$\mathbf{\Theta} = \Big\{\Big\{w_1, \underline{\mu}_1, \mathbf{C}_1\Big\}, \ \ldots, \ \Big\{w_M, \underline{\mu}_M, \mathbf{C}_M\Big\}\Big\}$$

from the weighted samples $\mathbf{Y}$, ideally in a maximum likelihood sense

$$\widehat{\mathbf{\Theta}}^{\mathrm{ML}} = \arg\max_{\mathbf{\Theta}} \Big\{ f_{\mathbf{Y}|\mathbf{\Theta}}(\mathbf{Y} \,|\, \mathbf{\Theta}) \Big\} \quad .$$

This can be done via numerical optimization or, more efficiently and gradient-free, using the EM algorithm. For the EM algorithm, we additionally consider a hidden variable $\mathbf{H}$. It contains the association probabilities $\eta_{i,m}$ between samples $\{\alpha_i, \underline{s}_i\}$ and GM components $\Big\{w_m, \underline{\mu}_m, \mathbf{C}_m\Big\}$.

IV. Key Idea

We believe that the following two things should give the same contribution to the result: i) one sample with double weight, and ii) two single-weight samples that are arranged with infinitesimally small or zero distance between them. Therefore, we propose to determine the hidden association parameters $\mathbf{H}$ only based on sample locations. In other words, we use the observed sample weights only in the maximization step and not in the expectation step.

For the maximization step, we propose to estimate GM component weights, means, and covariances as a weighted average. Thereby we combine the two weightings from a) the given observed sample weights and b) the sample-to-GM component associations, via multiplication and re-normalization.

V. Implementation of Proposed Method

Associations $\mathbf{H}$ between samples and GM components are unknown but necessary for an EM algorithm in order to independently calculate moments of individual mixtures. Marginalization over all possible associations

$$f_{\mathbf{Y}|\mathbf{\Theta}}(\mathbf{Y}|\mathbf{\Theta}) = \int f_{\mathbf{H},\mathbf{Y}|\mathbf{\Theta}}(\mathbf{H}, \mathbf{Y} \,|\, \mathbf{\Theta}) \, \mathrm{d}\mathbf{H} \quad ,$$

is infeasible, hence the separation into expectation and maximization steps according to the EM algorithm.

A. Expectation Step

Besides the given observed data $\mathbf{Y}$, we assume an estimate $\widehat{\mathbf{\Theta}}^{(r)}$ of the parameter vector containing the GM parameters $\Big\{w_m^{(r)}, \underline{\mu}_m^{(r)}, \mathbf{C}_m^{(r)}\Big\}$, $m \in \{1, \ldots, M\}$, with iteration index (r), to obtain a new estimate of the hidden data $\widehat{\mathbf{H}}^{(r+1)}$

$$\eta_{i,m}^{(r+1)} = \frac{w_m^{(r)} \, \mathcal{N}\Big(\underline{s}_i - \underline{\mu}_m^{(r)},\ \mathbf{C}_m^{(r)}\Big)}{\sum_{\tilde{m}=1}^{M} w_{\tilde{m}}^{(r)} \, \mathcal{N}\Big(\underline{s}_i - \underline{\mu}_{\tilde{m}}^{(r)},\ \mathbf{C}_{\tilde{m}}^{(r)}\Big)} \quad , \qquad (1)$$

with matrix elements $\Big[\widehat{\mathbf{H}}^{(r+1)}\Big]_{i,m} = \eta_{i,m}^{(r+1)}$. Due to the normalization such that the row sum is equal to one, $\widehat{\mathbf{H}}^{(r+1)}$ describes a "probability of association" for each sample i to each component m of the GM.

B. Maximization Step

Using said estimate of the hidden data $\widehat{\mathbf{H}}^{(r+1)}$ and also the observed data $\mathbf{Y}$, i.e., sample locations $\underline{s}_i$ and sample weights α_i, we obtain a new estimate of the parameter vector $\widehat{\boldsymbol{\Theta}}^{(r+1)}$

$$w_m^{(r+1)} = \frac{\sum_{i=1}^{L} \eta_{i,m}^{(r+1)} \alpha_i}{\sum_{\tilde{m}=1}^{M} \sum_{\tilde{i}=1}^{L} \eta_{\tilde{i},\tilde{m}}^{(r+1)} \alpha_{\tilde{i}}} = \frac{\sum_{i=1}^{L} \eta_{i,m}^{(r+1)} \alpha_i}{\sum_{\tilde{i}=1}^{L} \alpha_{\tilde{i}}} \ , \quad (2)$$

$$\underline{\mu}_m^{(r+1)} = \frac{\sum_{i=1}^{L} \eta_{i,m}^{(r+1)} \alpha_i \, \underline{s}_i}{\sum_{\tilde{i}=1}^{L} \eta_{\tilde{i},m}^{(r+1)} \alpha_{\tilde{i}}} \ , \quad (3)$$

$$\mathbf{C}_m^{(r+1)} = \frac{\sum_{i=1}^{L} \eta_{i,m}^{(r+1)} \alpha_i \left(\underline{s}_i - \underline{\mu}_m^{(r+1)}\right) \left(\underline{s}_i - \underline{\mu}_m^{(r+1)}\right)^{\top}}{\sum_{\tilde{i}=1}^{L} \eta_{\tilde{i},m}^{(r+1)} \alpha_{\tilde{i}}} \ . \quad (4)$$

C. Split Sample Linearity

The obtained parameter estimate $\widehat{\boldsymbol{\Theta}}^{(r+1)}$, after performing one expectation and maximization step for some given prior parameter estimate $\widehat{\boldsymbol{\Theta}}^{(r)}$, is identical whether we have a set of weighted samples

$$\mathbf{Y} = \{\{\alpha_1, \underline{s}_1\}, \{\alpha_2, \underline{s}_2\}, \ldots, \{\alpha_L, \underline{s}_L\}\} \ ,$$

or "split samples" with, e.g., two samples and two weights at each sample location, i.e.,

$$\widetilde{\mathbf{Y}} = \Big\{ \left\{\alpha_1^{(1)}, \underline{s}_1\right\}, \left\{\alpha_1^{(2)}, \underline{s}_1\right\}, \left\{\alpha_2^{(1)}, \underline{s}_2\right\}, \left\{\alpha_2^{(2)}, \underline{s}_2\right\}, \ldots, \left\{\alpha_L^{(1)}, \underline{s}_L\right\}, \left\{\alpha_L^{(2)}, \underline{s}_L\right\} \Big\} \ , \quad (5)$$

with $\alpha_i = \alpha_i^{(1)} + \alpha_i^{(2)} \ \forall \ i \in \{1, \ldots, L\}$. This is because association probabilities $\eta_{i,m}^{(r+1)}$ in the expectation step do not depend on sample weights α_i, and for the maximization step due to its linearity it does not matter whether there are two samples with weights $\alpha_i^{(1)}$, $\alpha_i^{(2)}$ at the same location $\underline{s}_i$, or only one sample that contains their combined weight α_i.

Note that the same holds for any other linear combination of more than two weights and samples at each same sample location, moreover not all but only a few sample locations may exhibit "split samples". We see this invariance against "split samples" as a logical sanity check the method should pass in order to be consistent.

VI. Implementation in [1]

For comparison, we reproduce the implementation from [1], [27] and highlight the differences to what we propose.

A. Expectation Step in [1]

For estimating the associations $\eta_{i,m}^{(r+1)}$ between samples $\underline{s}_i$ and GM components $\left\{w_m^{(r)}, \underline{\mu}_m^{(r)}, \mathbf{C}_m^{(r)}\right\}$, the covariances $\mathbf{C}_m^{(r)}$ of the individual Gaussian components are scaled in [1] based on the sample weights α_i

$$\eta_{i,m}^{(r+1)} = \frac{w_m^{(r)} \, \mathcal{N}\left(\underline{s}_i - \underline{\mu}_m^{(r)}, \ \mathbf{C}_m^{(r)}/\alpha_i\right)}{\sum_{\tilde{m}=1}^{M} w_{\tilde{m}}^{(r)} \, \mathcal{N}\left(\underline{s}_i - \underline{\mu}_{\tilde{m}}^{(r)}, \ \mathbf{C}_{\tilde{m}}^{(r)}/\alpha_i\right)} \ .$$

We however propose to use the GM covariances $\mathbf{C}_m^{(r)}$ without any sample-specific adaptions (1).

B. Maximization Step in [1]

In [1], sample weights α_i are not considered when calculating the Gaussian mixture component weights $w_m^{(r+1)}$

$$w_m^{(r+1)} = \frac{1}{L} \sum_{i=1}^{L} \eta_{i,m}^{(r+1)} \ ,$$

compare (2). Calculation of component means $\underline{\mu}_m^{(r+1)}$

$$\underline{\mu}_m^{(r+1)} = \frac{\sum_{i=1}^{L} \eta_{i,m}^{(r+1)} \alpha_i \underline{s}_i}{\sum_{\tilde{i}=1}^{L} \eta_{\tilde{i},m}^{(r+1)} \alpha_{\tilde{i}}}$$

is identical to our proposed method (3). For component covariance estimation $\mathbf{C}_m^{(r+1)}$, the difference between [1] and our proposed method (4) is that sample weights α_i are not considered for normalization

$$\mathbf{C}_m^{(r+1)} = \frac{\sum_{i=1}^{L} \eta_{i,m}^{(r+1)} \alpha_i \left(\underline{s}_i - \underline{\mu}_m^{(r+1)}\right) \left(\underline{s}_i - \underline{\mu}_m^{(r+1)}\right)^{\top}}{\sum_{\tilde{i}=1}^{L} \eta_{\tilde{i},m}^{(r+1)}} \ .$$

C. Split Sample Linearity in [1]

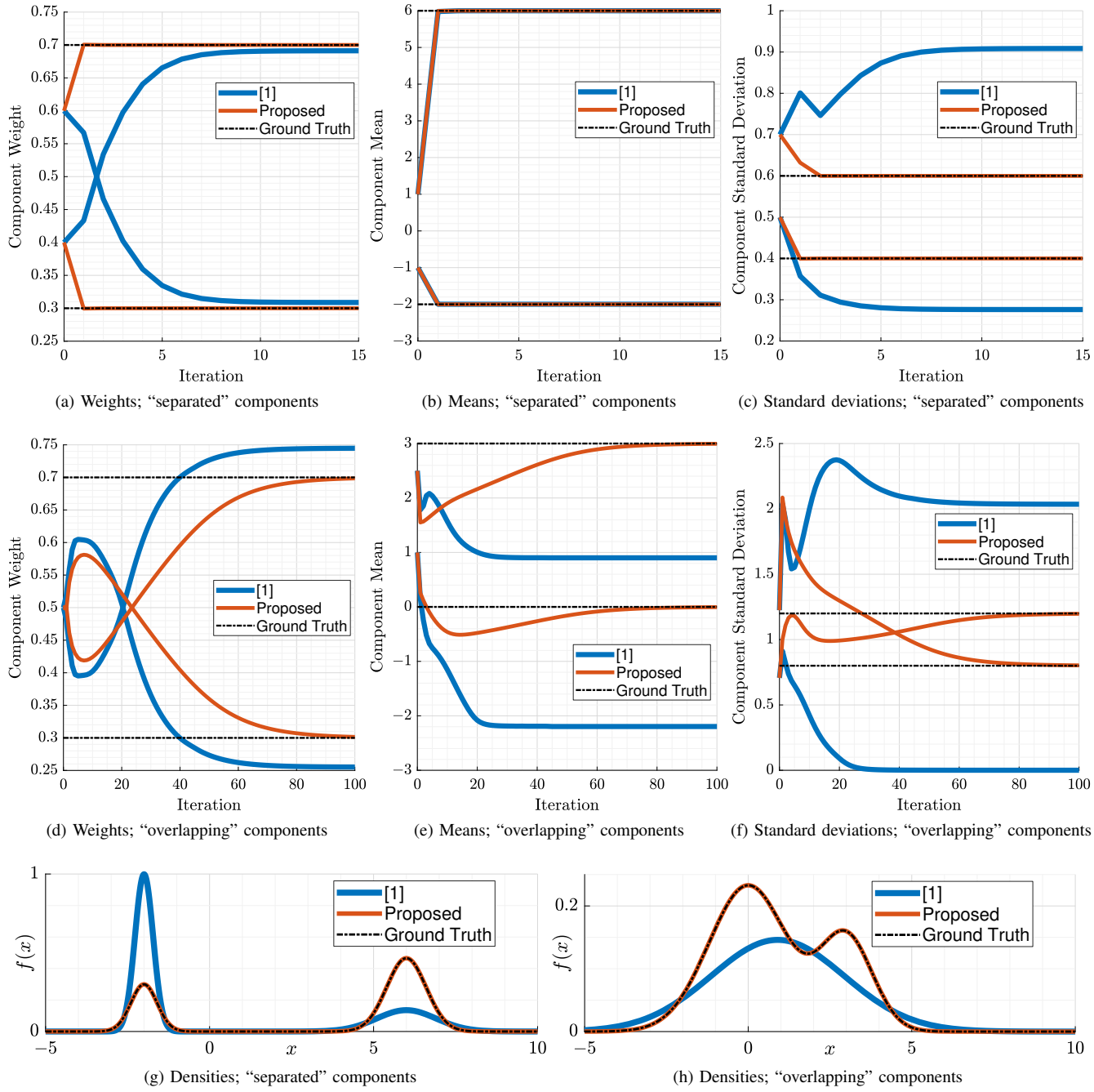
For the EM method according to [1], the result of each iteration is different when we "split" some samples in different ways, e.g., in two parts (5). Therefore, a double sample weight is *not* equivalent with two samples at the same location. The evaluation section will demonstrate that not only the individual iteration results, but also the final result differs from our proposed method, and from the ground truth.

VII. Evaluation and Comparison with [1]

As the simplest example, we define a one-dimensional GM with two components. A large number of equidistant samples is placed in the relevant region, and the GM density function at each sample location is used as the respective sample weight. Furthermore, some random initial guess of the GM parameters is given. Two algorithms are compared in solving this density estimation problem. First, our proposed method as defined in Sec. V, and second, the method as proposed in [1] and replicated here in Sec. VI.

One setup is defined where the two Gaussian components are rather "separated", this can be solved with about 15 iteration steps, see Fig. 2 (a, b, c). A second setup has Gaussian components that are closer together and exhibit some "overlap" of probability mass. Both EM algorithms need much more iteration steps to converge here, see Fig. 2 (d, e, f).

For the "separated" Gaussian components we find that all algorithms provide a very good estimate of the GM component means after about three iterations. The weighting factor estimates need more iterations to converge and are slightly off with the algorithm from [1]. Standard deviations from [1] are not reliable at all. Only our proposed method provides accurate results here. In the "overlapping" setup,



(a) Weights; "separated" components (b) Means; "separated" components (c) Standard deviations; "separated" components

(d) Weights; "overlapping" components (e) Means; "overlapping" components (f) Standard deviations; "overlapping" components

(g) Densities; "separated" components (h) Densities; "overlapping" components

Fig. 2: A simple scalar example with two GM components. Equidistant samples were weighted with the ground truth probability density function, and the GM parameters (component weights, means, and variances) were estimated with our proposed method (red lines) and the method from [1] (blue lines). Ideally, the estimations should converge to the ground truth (black dashed lines) after some iterations.

the GM component weight, mean, and variance estimates converge to solutions that are significantly off the ground truth when using the method from [1]. Our proposed method needs more iterations to converge but finds the accurate solution in the end.

Matlab source code with an implementation of the presented GM estimation from weighted samples is provided alongside this paper in the IEEE Xplore Code Ocean system. The code for the existing method has been taken from [28].

VIII. Conclusions

Considering weighted samples opens new applications for GM estimation, e.g., in the field of Bayesian estimation, see [10]. The correct treatment of weighted samples in GM estimation was derived. It was shown that current approaches have a serious flaw that leads to wrong estimates. The proper modifications can simply be added to existing GM estimation code to extend its applicability to weighted samples. The proposed method is also a plugin replacement for standard GM estimators as it is backwards compatible for unweighted samples.

Often the optimal number of Gaussian components describing a set of weighted samples is not known beforehand. Therefore, in future work, the algorithm should be extended to automatically detect an optimal number of Gaussian components.

References

[1] I. D. Gebru, X. Alameda-Pineda, F. Forbes, and R. Horaud, "EM Algorithms for Weighted-Data Clustering with Application to Audio-Visual Scene Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 12, pp. 2402–2415, 2016.

[2] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data Clustering: A Review," *ACM Comput. Surv.*, vol. 31, no. 3, p. 264–323, Sep. 1999. [Online]. Available: https://doi.org/10.1145/331499.331504

[3] J. Larsen, A. Szymkowiak, and L. K. Hansen, "Probabilistic Hierarchical Clustering With Labeled and Unlabeled Data," *International Journal of Knowledge Based Intelligent Engineering Systems*, vol. 6, no. 1, pp. 56–63, 2002.

[4] C. Premebida and U. Nunes, "A Multi-Target Tracking and GMM-Classifier for Intelligent Vehicles," in *2006 IEEE Intelligent Transportation Systems Conference*, 2006, pp. 313–318.

[5] D. E. Clark and J. Bell, "Multi-Target State Estimation and Track Continuity for the Particle PHD Filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 4, pp. 1441–1453, 2007.

[6] S. K. Pang, J. Li, and S. J. Godsill, "Detection and Tracking of Coordinated Groups," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 1, pp. 472–502, 2011.

[7] E. Tzoreff and A. J. Weiss, "Expectation-Maximization Algorithm for Direct Position Determination," *Signal Processing*, vol. 133, pp. 32–39, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168416302821

[8] H. Li, J. K. Ng, V. C. Cheng, and W. K. Cheung, "Fast Indoor Localization for Exhibition Venues With Calibrating Heterogeneous Mobile Devices," *Internet of Things*, vol. 3-4, pp. 175–186, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660518300611

[9] L. Dovera and E. Della Rossa, "Multimodal Ensemble Kalman Filtering Using Gaussian Mixture Models," *Computational Geosciences*, vol. 15, no. 2, pp. 307–323, Mar 2011. [Online]. Available: https://doi.org/10.1007/s10596-010-9205-3

[10] D. Frisch and U. D. Hanebeck, "Progressive Bayesian Update Using Interleaved Gaussian Mixture and Dirac Mixture," in *Proceedings of the 23rd International Conference on Information Fusion (Fusion 2020)*, Virtual, Jul. 2020.

[14] S. Kim, "Weighted K-Means Support Vector Machine for Cancer Prediction," *SpringerPlus*, vol. 5, no. 1, p. 1162, Jul 2016. [Online]. Available: https://doi.org/10.1186/s40064-016-2677-4

[11] C. Drummond, R. C. Holte *et al.*, "C4. 5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling Beats Over-Sampling," in *Workshop on learning from imbalanced datasets II*, vol. 11. Citeseer, 2003, pp. 1–8.

[12] C. R. Palmer and C. Faloutsos, "Density Biased Sampling: An Improved Method for Data Mining and Clustering," *SIGMOD Rec.*, vol. 29, no. 2, p. 82–92, May 2000. [Online]. Available: https://doi.org/10.1145/335191.335384

[13] K. Kerdprasop, N. Kerdprasop, and P. Sattayatham, "Weighted K-Means for Density-Biased Clustering," in *Data Warehousing and Knowledge Discovery*, A. M. Tjoa and J. Trujillo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 488–497.

[15] G. C. Tseng, "Penalized and Weighted K-Means for Clustering With Scattered Objects and Prior Information in High-Throughput Biological Data," *Bioinformatics*, vol. 23, no. 17, pp. 2247–2255, 06 2007. [Online]. Available: https://doi.org/10.1093/bioinformatics/btm320

[16] G. N. Abras and V. L. Ballarín, "A Weighted K-Means Algorithm Applied to Brain Tissue Classification," *Journal of Computer Science & Technology*, vol. 5, 2005.

[17] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An Efficient K-Means Clustering Algorithm: Analysis and Implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[18] G. Hamerly and C. Elkan, "Alternatives to the K-Means Algorithm That Find Better Clusterings," in *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, ser. CIKM '02. New York, NY, USA: Association for Computing Machinery, 2002, p. 600–607. [Online]. Available: https://doi.org/10.1145/584792.584890

[19] A. Likas, N. Vlassis, and J. J. Verbeek, "The Global K-Means Clustering Algorithm," *Pattern Recognition*, vol. 36, no. 2, pp. 451–461, 2003, biometrics. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320302000602

[20] R. A. Redner and H. F. Walker, "Mixture Densities, Maximum Likelihood and the EM Algorithm," *SIAM Review*, vol. 26, no. 2, pp. 195–239, 1984.

[21] 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)*, vol. 39, no. 1, pp. 1–22, 1977.

[22] C. F. J. Wu, "On the Convergence Properties of the EM Algorithm," *The Annals of Statistics*, vol. 11, no. 1, pp. 95–103, 1983.

[23] M. A. T. Figueiredo and A. K. Jain, "Unsupervised Learning of Finite Mixture Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, 2002.

[24] B. Zhang, C. Zhang, and X. Yi, "Competitive EM Algorithm for Finite Mixture Models," *Pattern Recognition*, vol. 37, no. 1, pp. 131–144, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320303001407

[25] P. Paalanen, J.-K. Kamarainen, J. Ilonen, and H. Kälviäinen, "Feature representation and discrimination based on Gaussian mixture model probability densities—Practices and algorithms," *Pattern Recognition*, vol. 39, no. 7, pp. 1346 – 1358, 2006.

[26] V. Melnykov and I. Melnykov, "Initializing the EM Algorithm in Gaussian Mixture Models with an Unknown Number of Components," *Computational Statistics & Data Analysis*, vol. 56, no. 6, pp. 1381–1395, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167947311003963

[27] I. D. Gebru, X. Alameda-Pineda, R. Horaud, and F. Forbes, "Audio-Visual Speaker Localization Via Weighted Clustering," in *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2014, pp. 1–6.

[28] I. D. Gebru, "EM_WD," https://github.com/isrish/EM_WD, 2016.