mathematics

MDPI

Article

Measure of Similarity between GMMs by Embedding of the Parameter Space that Preserves KL Divergence

Branislav Popović [1], Lenka Cepova [2], Robert Cep [2], Marko Janev [3] and Lidija Krstanović [1,*]

[1] Faculty of Technical Sciences, University of Novi Sad, Trg D. Obradovića 6, 21000 Novi Sad, Serbia; bpopovic@uns.ac.rs
[2] Department of Machining, Faculty of Mechanical Engineering, VSB-Technical University of Ostrava, Assembly and Engineering Metrology, 17. listopadu 2172/15, 708 00 Ostrava Poruba, Czech Republic; lenka.cepova@vsb.cz (L.C.); robert.cep@vsb.cz (R.C.)
[3] Institute of Mathematics, Serbian Academy of Sciences and Arts, Kneza Mihaila 36, 11000 Belgrade, Serbia; markojan@uns.ac.rs
* Correspondence: lidijakrstanovic@uns.ac.rs

Abstract: In this work, we deliver a novel measure of similarity between Gaussian mixture models (GMMs) by neighborhood preserving embedding (NPE) of the parameter space, that projects components of GMMs, which by our assumption lie close to lower dimensional manifold. By doing so, we obtain a transformation from the original high-dimensional parameter space, into a much lower-dimensional resulting parameter space. Therefore, resolving the distance between two GMMs is reduced to (taking the account of the corresponding weights) calculating the distance between sets of lower-dimensional Euclidean vectors. Much better trade-off between the recognition accuracy and the computational complexity is achieved in comparison to measures utilizing distances between Gaussian components evaluated in the original parameter space. The proposed measure is much more efficient in machine learning tasks that operate on large data sets, as in such tasks, the required number of overall Gaussian components is always large. Artificial, as well as real-world experiments are conducted, showing much better trade-off between recognition accuracy and computational complexity of the proposed measure, in comparison to all baseline measures of similarity between GMMs tested in this paper.

Keywords: Gaussian mixture models; similarity measures; dimensionality reduction; KL-divergence

Citation: Popović, B.; Cepova, L.; Cep, R.; Janev, M.; Krstanović, L. Measure of Similarity between GMMs by Embedding of the Parameter Space That Preserves KL Divergence. *Mathematics* **2021**, *9*, 957. https://doi.org/10.3390/math9090957

Academic Editor: David Delgado-Gómez

Received: 23 March 2021
Accepted: 20 April 2021
Published: 25 April 2021

1. Introduction

The Gaussian Mixture Models have been used for many years in pattern recognition, computer vision, and other machine learning systems, due to their vast capability to model arbitrary distributions and their simplicity. The comparison between two GMMs plays an important role in many classification problems in the areas of machine learning and pattern recognition, due to the fact that arbitrary pdf could be successfully modeled by a GMM, knowing the exact number of "modes" of that particular pdf. Those problems include, but are not limited to speaker verification and/or recognition [1], content-based image matching and retrieval [2,3] (also classification [4], segmentation, and tracking), texture recognition [2,3,5–8], genre classification, etc. In the area of Variational Auto-encoders (VAE), extensively used in emerging field of deep learning, GMMs have recently found their gateway (see [9]) with promising results. Many authors considered the problem of developing the efficient similarity measures between GMMs to be applied in such tasks (see for example [1–3,7,10]). The first group of those measures utilize informational distances. In some early works, Chernoff distance, Bhattacharyya distance, and Matusita distance were explored (see [11–13]). Nevertheless, Kullback–Leibler (KL) divergence [14] emerged as the most natural and effective informational distance measure. It is actually an informational distance between two probability distributions p and q. While the solution for the KL divergence between two Gaussian components exists in the analytic, i.e., closed-form, there is no

analytic solution for the KL divergence between arbitrary GMMs, which is very important for various applications. The straight-forward solution of the mentioned problem is to calculate KL divergence between two GMMs via the Monte-Carlo method (see [10]). However, it is almost always an unacceptably computationally expensive solution, especially when dealing with a huge amount of data and large dimensionality of the underlying feature space. Thus, many researchers proposed different approximations for the KL divergence, trying to obtain acceptable precision in recognition tasks of interest. In [2], one such approximation is proposed and applied in image retrieval task as a measure of similarity between images. In [10], lower and upper approximation bounds are delivered by the same authors. Experiments are conducted on synthetic data, as well as in speaker verification task. In [1], accurate approximation built upon Unscented Transform is delivered and applied within a speaker recognition task in a computationally efficient manner. In [15], the authors proposed a novel approach to online estimation of pdf's, based on kernel density estimation. The second group of measures utilize informational geometry. In [16], the authors proposed a metric on the space of multivariate Gaussians by parameterizing that space as the Riemannian symmetric space. In [3], motivated by the mentioned paper and the efficient application of vector-based Earth-Movers Distance (EMD) metrics (see [17]) applied in various recognition tasks (see for example [18]), and their extension to GMMs in texture classification task proposed in [6], the authors proposed sparse EMD methodology for Image Matching based on GMMs. An unsupervised sparse learning methodology is presented in order to construct EMD measure, where the sparse property of the underlying problem is assumed. In experiments, it proved to be more efficient and robust than the conventional EMD measure. Their EMD approach utilizes information geometry based ground distances between component Gaussians, introduced in [16]. On the other hand, their supervised sparse EMD approach uses an effective pair-wise-based method in order to learn GMM EMD metric among GMMs. Both of these methods were evaluated using synthetic as well as real data, as part of texture recognition and image retrieval tasks. Higher recognition accuracy is obtained in comparison to some state-of-the-art methods. In [7], the method proposed in [3] was expanded. A study concerning ground distances and image features such as Local Binary Pattern (LBP) descriptor, SIFT, high-level features generated by deep convolution networks, covariance descriptor, and Gabor filter is also presented.

One of the main issues in pattern recognition and machine learning as a whole is that data are represented in high-dimensional spaces. This problem appears in many applications, such as information retrieval (and especially image retrieval), text categorization, texture recognition, and appearance-based object recognition. Thus, the goal is to develop the appropriate representation for complex data. The variety of dimensionality reduction techniques are designed in order to cope with this issue, targeting problems such as "curse of dimensionality" and computational complexity in the recognition phase of ML task. They tend to increase discrimination of the transformed features, which now lie either on a subspace of the original high dimensional feature space, or more generally, on some lower dimensional manifold embedded into it. Those are the so called *manifold learning* techniques. Some of the most commonly used subspace techniques, such as Linear Discriminant Analysis (LDA) [19] and maximum margin criterion (MMC) [3,20], trained in a supervised manner, or for example Principal Component Analysis (PCA) [21], trained in an unsupervised manner, handle this issue by trying to increase discrimination of the transformed features, and to decrease computational complexity during recognition. Some of the frequently used manifold learning techniques are Isomap [22], Laplacian Eigenmaps (LE) [23], Locality Preserving Projections (LPP) [24] (approach based on LE), and Local Linear Embedding (LLE) [25]. The LE method explores the connection between the graph Laplasian and the Laplace Beltrami operator, in order to project features in a locally-preserving manner. Nevertheless, it is only to be used in various spectral clustering applications, as it cannot deal with unseen data. An approach based on LE, called Locality Preserving Projections (LPP) (see [24]), manages to resolve the previous problem

by learning linear projective map which best "fits" in the manifold, therefore preserving local properties of the data in the transformed space. In this way, we can transform any unseen data into a low-dimensional space, which can be applied in a number of pattern recognition and machine learning tasks. In [26], the authors proposed the Neighborhood Preserving Embedding (NPE) methodology that, similarly to LPP, aims to preserve the local neighborhood structure on data manifold, but it learns not only the projective matrix which projects the original features to lower-dimensional Euclidean feature space, but also, as an intermediate optimization step, the weights that extract the neighborhood information in the original feature space. In [27], some of the previously mentioned methods, such as LE and LLE, are generalized. An example of LE is given for the Riemannian manifold of positive-definite matrices, and applied as part of image segmentation task. Note that the mentioned dimensionality reduction techniques are applicable in many recent engineering and scientific fields, such as social network analysis and intelligent communications (see for example [28,29] , published within a special issue presented in an editorial article [30]).

In many machine learning systems, the trade-off between recognition accuracy and computational efficiency is very important for those to be applicable in real-life. In this work, we construct a novel measure of similarity between arbitrary GMMs, with an emphasis on lowering the complexity of the representation of all GMMs used in a particular system. Our aim is to investigate the assumption that the parameters of full covariance Gaussians, i.e., the components of GMMs, lie close to each other in a lower-dimensional surface embedded in the cone of positive definite matrices for the particular recognition task. Note that this is contrary to the assumption that data themselves lie on the lower-dimensional manifold embedded in the feature space. We actually use the NPE-based idea in order to reduce the projection matrix A, but we apply it on the parameter space of Gaussian components. The matrix A projects the parameters of Gaussian components to a lower-dimensional space. Local neighborhood information from the original parameter space is preserved. Let $\mathcal{N}(\mu_i, \Sigma_i)$, $i = 1, \ldots, M$ be a set of all Gaussian components, and M is the number of Gaussians for the particular task. We assume that parameters of any multivariate Gaussian component $\mathcal{N}(\mu_i, \Sigma_i)$, given as vectorized pair (μ_i, Σ_i), live in a high-dimensional parameter space. Each Gaussian component is then assigned to a node of undirected weighted graph. The graph weights W_{ij} are learned in the intermediate optimization step, forming the weight matrix W, where instead of the Euclidean distance figuring in the particular cost functional that is used in baseline NPE operating on feature space, we use a specified measure of similarity between Gaussian components and plug it into the cost functional. The ground distances between Gaussians $N(\mu_i, \Sigma_i)$ and $N(\mu_j, \Sigma_j)$, proposed in [3,16], are based on information geometry. We name the proposed GMM similarity measure as GMM-NPE.

2. GMM Similarity Measures

KL divergence is the most natural measure between probability distributions p and q. The measure is defined as $KL(p||q) = \int_{\mathbb{R}^d} p(x) \log \frac{p(x)}{q(x)} dx$. However, as mentioned in the previous section, in the case of GMMs, it cannot be expressed as the closed-form solution.

The straightforward, but at the same time the most expensive, is a computation calculated by using the standard Monte-Carlo method (see [31]). The idea is to sample the probability distribution f by using i.i.d. samples x_i, $i = 1, \ldots, N$, such that $E_f\left[\ln \frac{f(x)}{g(x)}\right] = KL(f||g)$. It is given by:

$$KL_{MC}(f||g) \approx \frac{1}{N} \sum_{i=1}^{N} \ln \frac{f(x_i)}{g(x_i)}. \tag{1}$$

Although it is the most accurate, the Monte-Carlo approximation (1) is computationally unacceptably expensive in real world applications, especially in recent years, when there is a huge amount of data present (big data) in almost all potential areas of interest. In order to cope with the mentioned problem, i.e., to obtain fast, but at the same time accurate, approximation, various approximations of the KL-divergence between two

GMMs are proposed in [2–31]. The roughest approximation is based on the convexity of the KL-divergence [32] and for two GMMs $f = \sum_{i=1}^{n} \alpha_i f_i$ and $g = \sum_{j=1}^{m} \beta_j g_j$, it holds

$$KL(f||g) \leq \sum_{i,j} \alpha_i \beta_j KL(f_i||g_j), \tag{2}$$

where $f_i = \mathcal{N}(\Sigma_i, \mu_i)$ and $g_j = \mathcal{N}(\Sigma_j, \mu_j)$ are Gaussian components of the corresponding mixtures, while $\alpha_i > 0$, $\beta_j > 0$ are corresponding weights, satisfying $\sum_i \alpha_i = 1$, $\sum_j \beta_j = 1$. The "roughest" approximation by upper bound (2), yielding the weighted average version given by

$$KL_{WE}(f||g) \approx \sum_{i,j} \alpha_i \beta_j KL(f_i||g_j) \tag{3}$$

plays special role in the case when Gaussians from different GMMs stand far from each other. On the other hand, KL divergence $KL(f_i||g_j)$ between corresponding Gaussians exists in the closed-form given by

$$KL(f_i||g_j) = \ln \frac{|\Sigma_{f_i}|}{|\Sigma_{g_j}|} + Tr\left[\Sigma_{g_j}^{-1}\Sigma_{f_i}\right] + (\mu_{f_i} - \mu_{g_j})^T \Sigma_{g_j}^{-1}(\mu_{f_i} - \mu_{g_j}) - d, \tag{4}$$

so that (3) is computationally much cheaper than the Monte-Carlo approximation (1).

Various approximations of the KL divergence between two GMMs were proposed in [1,2,10] and efficiently applied in real world problems, such as speech recognition, image retrieval, or speaker identification. For example, in [2], the Matching-based Approximation given by

$$KL_{MB}(f||g) \approx \sum_i \alpha_i \left[\min_j KL(f_i||g_j) + \log\left(\frac{\alpha_i}{\beta_j}\right)\right] \tag{5}$$

is proposed, based on the assumption that the element g_j, i.e., the one that is most proximate to f_i, dominates the integral $\int f_i \log g$. Motivated by (5), more efficient matching based approximation is given by

$$KL_{MBS}(f||g) \approx \sum_i \alpha_i \min_j KL(f_i||g_j), \tag{6}$$

showing good performances when the Gaussians figuring in f and those figuring in g are mostly far apart, but shows inappropriate if there is significant overlapping among Gaussian components of f and g. The authors proposed the Unscented Transform-based approximation as a way to deal with those overlapping situations. The Unscented Transformation is a mathematical function used to estimate the statistics of a random variable to which a nonlinear transformation is applied (see [33]). If it holds that $KL(f||g) = \int_{\mathbb{R}^d} f \log f - \int_{\mathbb{R}^d} f \log g$, the unscented transform approach tends to approximate integral $\int_{\mathbb{R}^d} f_i \log g$ as

$$\begin{aligned} &\int_{\mathbb{R}^d} f_i \log g \approx \frac{1}{2d} \sum_{k=1}^{2d} \log g(x_{i,k}) \\ &x_{i,k} = \mu_i + \left(\sqrt{\Sigma_i}\right)_k, \; k = 1, \ldots, d \\ &x_{i,d+k} = \mu_i - \left(\sqrt{\Sigma_i}\right)_k, \; k = 1, \ldots, d, \end{aligned} \tag{7}$$

where $\left(\sqrt{\Sigma_i}\right)_k$ is the k-th column of the matrix square root of Σ_i. Integrals $\int_{\mathbb{R}^d} f \log f$ and $\int_{\mathbb{R}^d} f \log f$ are now approximated in the previous manner, so that for second integral we have

$$\int_{\mathbb{R}^d} f \log g \approx \frac{1}{2d} \sum_{i=1}^{n} \alpha_i \sum_{k=1}^{2d} \log g(x_{i,k})$$

and similarly for the first. Thus, the $KL_{UC}(f||g)$ is obtained as above.

GMM distance which utilizes KL divergence $KL(f_i||g_j)$ between Gaussian components in order to obtain an approximate KL divergence between full GMMs is Variational approximation is proposed in [31] (see also [10]), given by

$$KL_{VAR}(f||g) = \sum_i \alpha_i \frac{\sum_{\hat{i}} \alpha_{\hat{i}} e^{-KL(f_i||f_{\hat{i}})}}{\sum_j \beta_j e^{-KL(f_i||g_j)}} \tag{8}$$

Earth-Movers Distance (EMD) methodology motivated various recognition tasks (see for example [17,18]). Based on that, the authors in [6] proposed EMD to measure the distributional similarity by sets of the Gaussian components representing texture classes. We denote it as EMD-KL measure. In [3], the authors incorporate ground distances between component Gaussians into the unsupervised sparse EMD-based distance metrics between GMMs, using the perspective from the Riemannian geometry and the work delivered in [16]. The first one is based on Lie Groups and it performs better when incorporated into the sparse EMD-based measure of similarity between GMMs than the second one, based on the products of Lie groups. We denote it as SR-EMD measure in the rest of the text.

3. NPE Dimensionality Reduction on Euclidean Data

Unlike PCA, which aims to preserve the global Euclidean structure, and similarly to LPP (see [24]), the nonlinear dimensionality reduction technique NPE [26] aims to preserve the local manifold structure of the input data. Given an embedded set of data points in the configuration space (they lie on a low dimensional manifold, i.e., it is assumed that the samples from the same class probably lie close to each other in the input space), we first build a weight matrix $W \in \mathbb{R}^{m \times d}$, which describes the relationship between data points. Namely, if we assume that data are embedded in the Euclidean $\mathbb{R}^d$ space, each data point $x_i \in \mathbb{R}^d$ is represented as the linear combination of neighboring data points, where for the neighboring data point x_j, the coefficients $w_{ij} \in \mathbb{R}$ in the weight matrix represent the "local proximity" of those two points in the configuration space. The goal is to find the optimal embedding in order to preserve the neighborhood structure in the reduced space. The NPE procedure consists of the following steps:

1. Constructing an adjacency graph: Let us consider a graph with $m \in \mathbb{N}$ nodes, where the i-th node corresponds to the data point x_i. One way to construct the adjacency graph is to use K nearest neighbors (KNN), where we direct an edge from node i to j if x_j is among the K nearest neighbors of x_i. The other one is ϵ neighborhood: Put an edge between nodes i and j if $\|x_i - x_j\| < \epsilon$.
2. Computing the weights: Let W denote the weight matrix with $W_{ij} > 0$ if there is an edge from node i to node j, and $W_{ij} = 0$ if there is no such edge. The weights on the edges can be computed by solving the following minimization problem:

$$\begin{aligned} &\min_W \sum_i \|x_i - W_{ij} x_j\|^2 \\ &\text{s.t} \sum_j W_{ij} = 1,\ i = 1, \ldots, m \end{aligned} \tag{9}$$

3. Computing the projections: In order to compute the projections, we need to solve the following optimization problem:

$$\begin{aligned} &\min_a \sum_i \left(y_i - \sum_j W_{ij} y_j \right) \\ &y = y^T = a^T X \\ &X = [x_1 | \cdots | x_m] \end{aligned} \tag{10}$$

which, by imposing constraint $a^T X X^T a = 1$ and by using the Lagrange multipliers, reduces to the following eigenvalue problem:

$$\begin{aligned} &XMX^T a = \lambda X X^T a \\ &M = (I - W)(I - W)^T. \end{aligned} \tag{11}$$

Since M is symmetric and positive semi-definite, its eigenvalues are real and non-negative. By taking the largest $l \in \mathbb{N}$, $l << d$ eigenvalues $\lambda_0, \ldots, \lambda_{l-1}$, and the corresponding l eigenvectors $a_0, \ldots, a_{l-1}$, we obtain the projection matrix $A = [a_1 | \cdots | a_{l-1}] \in \mathbb{R}^{l \times d}$ and the embedding $x_i \mapsto y_i = Ax_i$, now projecting from the high-dimensional R^d to the low-dimensional R^l Euclidean space. Readers can find more details on the subject in [26].

4. GMM Similarity Measure by the KL Divergence Preserving NPE Embedding of the Parameter Space

We propose a novel measure of similarity between arbitrary GMMs by utilizing the NPE-based technique and the KL divergence type ground distance between the Gaussian embedded components, i.e., their parameters, instead of the Euclidean distance between some observations, as in the standard NPE procedure used as a feature dimensionality reduction technique.

The first step is to learn the projective matrix A in the neighborhood preserving manner with respect to informational ground distance, i.e., the (non-symmetric) KL divergence between Gaussian components of GMMs used, and to project those (vectorized) parameters into the low-dimensional Euclidean parameter space. Our goal is to preserve the local neighborhood information which exists in the original parameter space, while dealing with much lower-dimensional space of transformed parameters. The aim is to obtain the best possible trade-off between the recognition precision and computational efficiency in a particular pattern recognition task. We call it the NPE-based measure of similarity between GMMs and denote it further by GMM-NPE.

The second step is to aggregate the non-negative real value which represents a measure between two particular GMMs. For that purpose, we compare the transformed "clouds" of lower dimensional Euclidean parameter vectors corresponding to the original Gaussian components of GMMs used, pondered by their belonging weights. The first, the simpler technique that we use is based on aggregation operators (the weighted max-min operator and maximum of the weighted sums operator in particular), which we apply on "clouds" of lower dimensional Euclidean parameter vectors in order to aggregate value representing the final measure between two GMMs. Note that, regardless of the usage of non-symmetric KL divergence in the first step, i.e., in the calculation of the projective matrix A, the properties of the invoked measure in terms of the symmetry, satisfying the triangle inequality, etc., depends on the second step, i.e., on the type of aggregation of value of the measure. We will comment later on those properties.

4.1. KL Divergence Type Ground Distance, Forming the NPE-Type Weights and the Projection Matrix

The goal is to use the NPE-like approach in order to obtain the projection matrix A which transforms vectorized representatives of $P_i \in Sym_+(d+1)$ corresponding to Gaussian components $g_i = \mathcal{N}(\Sigma_i, \mu_i), i = 1, \ldots, M$ featuring in GMMs, where M represents the overall number of components and d is the dimension of the underlying feature space. Then, as explained previously, the measure of similarity comparing the "clouds"

of pondered Euclidean vectors is to be used in order to obtain the final value of GMM measure.

To apply an NPE-like approach, we start from the fact that a set of multivariate Gaussians is a Riemannian manifold and that d-dimensional multivariate Gaussian components $g = \mathcal{N}(\mu, \Sigma)$ can be embedded into $Sym_+(d+1)$, i.e., a cone embedded in $n = d(d+1)/2 + d$ Euclidean dimensional space and also a Riemannian manifold [3,16]. It can be conducted as follows:

$$g \hookrightarrow P = |\Sigma|^{-\frac{1}{d+1}} \begin{bmatrix} \Sigma + \mu\mu^T & \mu \\ \mu^T & 1 \end{bmatrix} \tag{12}$$

$|\Sigma| > 0$ denotes the determinant of the covariance matrix of Gaussian component g. For the detailed mathematical theory behind the embedding (12), one can refer to [16]. We invoke the assumption that any representative $P_i \in Sym_+(d+1)$ can be approximated as the non-negative weighted sum of neighbors P_j in the following way:

$$\begin{aligned} &P_i \approx \sum_{j \in \mathcal{N}(i)} W_{ij} P_j = \hat{P}_i \\ &W_{ij} \geq 0, \end{aligned} \tag{13}$$

where $\mathcal{N}(i)$ is the set of indices of neighboring representatives, i.e., the representatives P_j, so that $D(P_i, P_j) \leq T$, where $T > 0$ is a predefined threshold. Recall that if we assign Gaussians $p_i = \mathcal{N}(0, P_i)$, $i = 1, 2$ to non-negative matrices P_i, $i = 1, 2$, the term $D(P_1, P_2)$ is defined as $D(P_1, P_2) = KL(p_1||p_2)$, where $KL(p_1||p_2)$ is given by the expression (4). Thus, we obtain the following optimization problem:

$$\begin{aligned} &\min_{W_{ij}} \sum_{i=1}^{M} D\left(\hat{P}_i, P_i\right) \\ &\hat{P}_i = \sum_{j \in \mathcal{N}(i)} W_{ij} P_j \\ &s.t. \\ &W_{ij} \geq 0,\ i, j = 1, \ldots, M, \\ &\hat{P}_i \preceq P_i \end{aligned}$$

which reduces to M independent optimization problems given below, for $i = 1, \ldots, M$:

$$\begin{aligned} &\min_{W_{ij}} D\left(\hat{P}_i, P_i\right) \\ &\hat{P}_i = \sum_{j \in \mathcal{N}(i)} W_{ij} P_j \\ &i = 1, \ldots, M \\ &s.t. \\ &W_{ij} \geq 0,\ j = 1, \ldots, M, \\ &\hat{P}_i \preceq P_i, \\ &i = 1, \ldots, M, \end{aligned}$$

where the constraint $0 \preceq \hat{P}_i \preceq P_i$ ensures that the residual is positive semi-definite, i.e., $E_i = P_i - \hat{P}_i \succeq 0$. By using (4), we have the following considerations:

$$\begin{aligned} &D(\hat{P}_i, P_i) = tr(\hat{P}_i P_i^{-1}) - \ln\det(\hat{P}_i P_i^{-1}) - (d+1) \\ &= tr(P_i^{-1/2} \hat{P}_i P_i^{-1/2}) - \ln\det\left(P_i^{-1/2} \hat{P}_i P_i^{-1/2}\right) - (d+1) \end{aligned} \tag{14}$$

and thus,

$$D(\hat{P}_i, P_i) = tr \sum_{j \in \mathcal{N}(i)} W_{ij} \tilde{P}_j^{(i)} - \ln \det \sum_{j \in \mathcal{N}(i)} W_{ij} \tilde{P}_j^{(i)},$$
$$\tilde{P}_j^{(i)} = P_i^{-1/2} P_j P_i^{-1/2}. \tag{15}$$

A more efficient way to achieve that only a few "neighbors" effect P_i is to include sparsity constrain in the form of l_1 norm of the weight matrix W (which is the convex relaxation of the l_0 norm). Thus, we include the additional term $\lambda \|W\|_1$ in the penalty function (14), where $\lambda > 0$ is a parameter representing the trade-off between sparser representation and closer approximation. The following sparse convex problem is obtained (similar as in [34]):

$$\min_{W_{ij}} \sum_{j=1}^{M} W_{ij} \left(tr(\tilde{P}_j^{(i)}) + \lambda \right) - \ln \det \sum_{j=1}^{M} W_{ij} \tilde{P}_j^{(i)}$$
$$s.t.$$
$$W_{ij} \geq 0,\ j = 1, \ldots, M,$$
$$\sum_{j=1}^{M} W_{ij} \tilde{P}_j^{(i)} \preceq I_{d+1},$$
$$i = 1, \ldots, M,$$

which is the final problem that we solve in order to obtain the weight matrix W. Note that $W_{ij} \geq 0$ ensures that the following condition is satisfied $\sum_{j=1}^{M} W_{ij} \tilde{P}_j^{(i)} \succeq 0$. The above formulation of tensor sparse coding is associated with the general class of optimization problems denoted as determinant maximization problems, or MAXDET [35], while semi-definite programming (SDP) and linear programming (LP) are its special cases. These problems are convex and could be solved by a class of interior-point methods (see for example [36]). In order to implement the actual optimization, we used CVX [37].

Forming the projection matrix A which projects the vectorized parameters (corresponding to P_i, i.e., the Gaussian representatives p_i), $\tilde{v}_i = (P_i) \in \mathbb{R}^n$, $n = d(d+1)/2$, $i = 1, \ldots, M$, into the lower l-dimensional Euclidean parameter space, with $n \gg l$, is the next step. It is similar to step 3 from Section 3, and thus includes solving the spectral problem (11).

4.2. Constructing the GMM-NPE Similarity Measure

The remaining task in constructing the final GMM-NPE similarity measure is to aggregate the non-negative real value which represents the measure of similarity between two particular GMMs. Actually, we have to compare the transformed "clouds" of lower l-dimensional Euclidean parameter vectors, with $l \ll n$, $n = d(d+1)/2$, corresponding to the original Gaussian components of GMMs used. We also have to encounter the belonging weights into final result. In all approaches that we utilize, for the particular m-component GMM $f = \sum_{i=1}^{m_f} \alpha_i f_i$ with $f_i = \mathcal{N}(\mu_i, \Sigma_i)$, we use the unique representative $F = (v_1, \ldots, v_{m_f}, \alpha_1, \ldots, \alpha_{m_f})$, with $v_i = A\tilde{v}_i \in \mathbb{R}^l$, $\tilde{v}_i = (P_i) \in \mathbb{R}^n$, $i = 1, \ldots, M$, with P_i defined by (12), where we plug μ_i and Σ_i, and where A is the projection matrix obtained as explained in the previous section. Using the above-given representation, the similarity measure between two GMMs given by $f = \sum_{i=1}^{m_f} \alpha_i f_i$, and $g = \sum_{i=1}^{m_g} \beta_j g_j$ can be invoked by simply comparing the corresponding representatives $F = \left(v_1, \ldots, v_{m_f}, \alpha_1, \ldots, \alpha_{m_f} \right)$ and $G = \left(u_1, \ldots, u_{m_g}, \beta_1, \ldots, \beta_{m_g} \right)$ in the transformed space, i.e., by comparing them as weighted low-dimensional Euclidean vectors. Various approaches can be applied to aggregate a single positive scalar value in order to represent a "distance" between F and G and therefore implicitly a "measure" between GMMs f and g. In this work, we use

two essentially different approaches. The first one is simpler and utilizes the arbitrary fuzzy union or intersection in order to extract the mentioned value, given, for example, by various aggregation operators (see, e.g., [38]). The second approach utilizes EMD distance on F and G, and it is based on the work proposed in [17].

For the first approach, we use types of fuzzy aggregation operators, operating on $\|v_i - u_j\|_2$, using α_i and β_j as weights. For the above-mentioned representatives, we apply the weighted max-min operator in the following way:

$$\begin{aligned} p_i &= \min\{\beta_j \|v_i - u_j\|_2 \mid j = 1, \ldots, m_f\} \\ a &= \max\{\alpha_i p_i \mid i = 1, \ldots, m_g\} \\ q_j &= \min\{\alpha_i \|v_i - u_j\|_2 \mid i = 1, \ldots, m_g\} \\ b &= \max\{\beta_j q_j \mid j = 1, \ldots, m_f\} \\ D_1(F, G) &= \frac{1}{2}(a + b), \end{aligned} \tag{16}$$

as well as the maximum of the positive weighted sums

$$\begin{aligned} a &= \max\{\alpha_i \sum_{j=1}^{n} \beta_j \|v_i - u_j\|_2 \mid i = 1, \ldots, m_g\} \\ b &= \max\{\beta_j \sum_{i=1}^{m} \alpha_i \|v_i - u_j\|_2 \mid j = 1, \ldots, m_f\} \\ D_2(F, G) &= \frac{1}{2}(a + b). \end{aligned} \tag{17}$$

We denote the previously invoked GMM measure induced by D_1 by GMM-NPE$_1$, while we denote the GMM measure induced by D_2 by GMM-NPE$_2$. Note that the choice of the particular fuzzy aggregation operator, i.e., the fuzzy measure, determines all the distance-wise properties of the final GMM similarity measure. Those are in our case the properties of D_1 and D_2. It is also interesting to discuss which properties of the KL divergence do GMM-NPE$_1$ and GMM-NPE$_2$ satisfy. Both of them satisfy self similarity and positivity, for arbitrary GMMs f and g, while self-identity is not satisfied. Furthermore, the measures D_1 and D_2 are both symmetric, while KL divergence is not. Nevertheless, note that we could easily obtain non-symmetry by, for example, letting $D_1(F, G) = a$ in (16), and $D_2(F, G) = a$ in (17), but we leave those considerations for some future work.

For the second, i.e., the EMD distance approach, the representatives F and G are interpreted as pondered "clouds" of Euclidean low-dimensional vectors. Thus, the final measure of similarity between GMMs f and g is given (see [17]) as follows:

$$D_{EMD}(F, G) = \frac{\sum_{i=1}^{m_f} \sum_{j=1}^{m_g} d_{ij} \zeta_{ij}}{\sum_{i=1}^{m_f} \sum_{j=1}^{m_g} \zeta_{ij}}, \tag{18}$$

where the flow $[\zeta_{ij}]$ is given as one that solves the following LP type minimization problem:

$$\begin{aligned}
&\min \sum_{i=1}^{m_f} \sum_{j=1}^{m_g} d_{ij}\zeta_{ij}, \\
&s.t. \\
&\zeta_{ij} \geq 0,\ i = 1,\ldots,m_f,\ j = 1,\ldots,m_g, \\
&\sum_{j=1}^{m_g} \zeta_{ij} \leq \alpha_i,\ i = 1,\ldots,m_f, \\
&\sum_{i=1}^{m_f} \zeta_{ij} \leq \beta_j,\ j = 1,\ldots,m_g, \\
&\sum_{i=1}^{m_f} \sum_{j=1}^{m_g} \zeta_{ij} = 1,
\end{aligned} \tag{19}$$

where $[d_{ij}]$ is the matrix of Euclidean distances between v_i and u_j, i.e., $d_{ij} = \|v_i - u_j\|$. Note that the constant 1 which appears in the right hand side of the constraint (19) is due to the fact that α_i, as well as β_j, sum to one. Thus, the term $D_{EMD}(F, G)$ is actually interpreted as the work necessary in order to move, by flow $[\zeta_{ij}]$, the maximum amount of supplies possible, from the "cloud" F to the "cloud" G. Furthermore, note that the fact that EMD distance is a metric (see [17]) implies that the measure of similarity between GMMs D_{EMD} defined by (18) is also a metric. Thus, similarly to the case of D_1 and D_2, it is symmetric. We denote the GMM measure induced by D_{EMD} by GMM-NPE$_3$.

4.3. Computational Complexity

In the given analysis, the computational efficiency of a measure is defined as the efficiency obtained in the testing (not the learning) phase. Let us, for the sake of simplicity and without loss of generality, further assume that GMMs f and g have the same number, $n = m$, and that we treat the full covariance case. Let d denotes the dimension of the original feature space. Let us first elaborate on baseline measures that we use.

The complexity of KL-based measures of similarity between GMMs KL$_{WE}$, KL$_{MB}$, and KL$_{VAR}$ (see [10]) given by (3)–(8), is roughly equivalent and estimated as $O(m^2d^3)$. Namely, as the complexity of calculating the KL divergence between two d-variate Gaussians is approximately equal to the complexity of calculating the inversion of a $d \times d$ matrix and it is of order $O(d^3)$, as there are m^2 such inversions, we obtain the previous estimate for the listed measures.

The Monte-Carlo approximation KL_{MC} (1) is the most computationally demanding. The computational complexity of Monte-Carlo approximation is estimated as $O(Nmd^3)$, where N is the number of samples. The estimate is then obtained using the arguments described above. Furthermore, in order to obtain an efficient approximation, the number of samples N has to be large, i.e., $N >> m$.

For the state-of-the-art EMD-based measures of similarity between GMMs proposed in [3], the computational complexity for SR-EMD measure can be estimated as $O(8m^5d^3)$, as LARS/Homotopy algorithms that are usually used to find a numerical solution of the optimization problem elaborated in SR-EMD converge in about $2m$ iterations (see [39]). Namely, as (19) is a LP problem, in one iteration, the computational complexity is of order $O(n_{const}n_{var})$, where n_{const} is a number of constraints and n_{var} is a number of variables for the particular problem. As it holds $n_{const} = n_{var} = m$ and the complexity of the inversion of $d \times d$ matrix is of order $O(d^3)$, since there are m^2 such inversions at each iteration, we obtain the previously mentioned estimate.

For the proposed similarity measures D_1 and D_2 given by (16) and (17), the analysis is as follows: the computational complexity of comparing F and G rise linearly with l and is given as $O(m^2l)$, where $l \ll d^3$ is delivered a priory on the base of the analysis of the eigenvalues, as explained at the end of Section 4.1. Nevertheless, if we encounter the computational

complexity required to transform the parameters of GMMs to the l dimensional space, there is an additional term $O(md^2l)$. One observes that for small l ($l \sim d$ in our experiments), the overall complexity of the proposed D_1 and D_2 is much smaller then all the baseline measures, and especially for large number of components m. For the EMD-based approach, i.e., the D_{EMD} given by (18), the computational complexity is estimated as the sum of $O(k_{iter}m^4l)$ term and the mentioned term $O(md^2l)$, making it significantly more efficient in comparison to EMD-KL and SR-EMD-M [3], as it holds $l \ll d \ll d^2$. Instead of calculating the KL divergence between two d-variate Gaussians, we calculate the Euclidian distance between two vectors of length l, which is of complexity $O(l)$.

5. Experimental Results

In this section, we present experiments comparing the proposed GMM-NPE measures with the baseline measures presented in Section 2. The experiments were conducted on synthetic as well as real data sets (texture recognition task). For the first case, synthetic data are constructed, satisfying specific assumptions, so that the proposed GMM-NPE measures could demonstrate their effectiveness over the baseline measures in such controlled conditions. In both synthetic and real data case, for the baseline measures, we chose KL_{WE}, KL_{MB}, and KL_{VAR}, defined by (3), (5), and (8), respectively. In the case of real data, we additionally use Earth mover based SR-EMD-M as well as SR-EMD-M-L. In the synthetic data scenario, the computational complexity was largely in favor of the proposed GMM-NPE measures, in all of our experiments. At the same time, the GMM-NPE measures obtained greater recognition precision in comparison to all baseline measures. On real data sets, significantly better trade-off between computational complexity and recognition precision is obtained for the proposed GMM-NPE measures, in comparison to all baseline measures.

5.1. Experiments on Synthetic Data

In order to demonstrate the effectiveness of the proposed method, we use toy examples consisting of two scenarios.

In the first scenario, we set the parameters of the Gaussians to lie on the low dimensional surface embedded in the cone $\mathcal{SPD}_+(d+1) \subset \mathbb{R}^n$, $n = (d+1)(d+2)/2$, where the covariance matrix is of dimension $d \times d$, with various dimensions d (d is also the dimension of the corresponding centroid), as it is given by (12). Dimensions of the surfaces containing data used in experiments are $l = 1$ and $l = 2$.

Mentioned surfaces are formed as follows: For the $l = 1$ case, we randomly generate positive-definite matrices A_1, A_2, both of dimension $d \times d$, in a following way: let $i = 1$ (the procedure is identical for $i = 2$). Firstly we generate a matrix $\tilde{A}_1$ containing independent, identically distributed (i.i.d.) elements, where we set pdf to be $\mathcal{U}([0,1])$. After symmetrization $\tilde{A}_1^{sym} = \frac{1}{2}(\tilde{A}_1 + \tilde{A}_1^T)$, we obtain matrix $\hat{A}_1$ by replacing only the diagonal elements of $\tilde{A}_1^{sym} = \left[\tilde{a}_{ij}\right]_{d\times d}$ with the sum of off-diagonal elements of matrix $\tilde{A}_1^{sym}$, i.e., $\hat{a}_{ii} \leftarrow \sum_{\substack{j=1 \\ j\neq i}}^{d} \tilde{a}_{ij}$ (note that $\tilde{a}_{ij} > 0$) and $\hat{a}_{ij} \leftarrow \tilde{a}_{ij}$ for $i \neq j$. Thus, as $\hat{A}_1$ is a symmetric and diagonally dominant matrix, it is positive semi-definite (see [40]). Finally, we obtain $A_1 = \tilde{A}_1 + \varepsilon I$, for some small $\varepsilon > 0$ (thus A_1 is positive definite), where we chose 0.00001 for all experiments. The same stands for matrix A_2. Finally, the $l = 1$ dimensional manifold in formed in the form of parabolic curve given by:

$$\begin{aligned} &F(t) = at^2\frac{A_2}{\|A_2\|} + bt\frac{A_1}{\|A_1\|} + c \in \mathcal{SPD}_+(d), \\ &t \in [r_1, r_2],\ r_1, r_2 \in \mathbb{R}_+ \cup \{0\},\ r_1 < r_2, \\ &a, b, c \in \mathbb{R}_+ \cup \{0\}, a \neq 0, \end{aligned} \tag{20}$$

and embedded into $\mathbb{R}^n$. For simplicity purposes, $a = 1, b, c = 0$ in all our experiments.

For the case $l = 2$, we form the $l = 2$ dimensional surface given by

$$F(t_1,t_2) = a\left(t_1^2\frac{A_2}{\|A_2\|} + t_2^2\frac{A_2}{\|A_2\|}\right) + b\left(t_1\frac{A_1}{\|A_1\|} + t_2\frac{A_1}{\|A_1\|}\right) + c \in \mathcal{SPD}_+(d),$$
$$t_1,t_2 \in [r_1,r_2],\ r_1,r_2 \in \mathbb{R},\ r_1 < r_2,$$
$$a,b,c \in \mathbb{R}_+ \cup \{0\}, a \neq 0, \quad (21)$$

embedded into $\mathbb{R}^n$. For the same reasons as in the case (20), we chose $a = 1, b, c = 0$ for all experiments.

We uniformly sample $N = 800$ Gaussians directly from the curve (20) for the $l = 1$ or (21) for the $l = 2$ case. From that pool, also by uniform sampling, we obtain M number of GMMs with the predefined size K, where we set all mixture weights to be $1/K$. For the acquired set of GMMs, we conduct "leave 10 percent out" cross-validation for every trial. We find that the estimated number of nonzero eigenvalues in all experiments $\hat{l}$ is fully coherent with the dimension l of the underlying manifolds, i.e., $\hat{l} = 1$ in the $l = 1$ case and $\hat{l} = 2$ in the $l = 2$ case, where the threshold for neglecting the eigenvalues was set to $T = 10^{-3}$. In all experiments, as the proposed method, we use GMM-NPE$_1$, GMM-NPE$_2$, or GMM-NPE$_3$. We vary the parameter K representing the size of a particular GMM used in the training as well as dimension d. We use different values for K, namely $K = 1$ and $K = 5$. In the case $l = 1$, we first set the means of the Gaussians to be zero vectors, where the results of experiments are presented for $[r_1,r_2] = [-3,5]$ and $[r_1,r_2] = [0,5]$ in Tables 1 and 2, respectively. Next, we make the means of Gaussians used in GMMs to be d dimensional vectors (we have $d \in \{10,20,30,50\}$ in all experiments), by setting all means belonging to the first class equal to some fix $m_1 \in \mathbb{R}^d$, and all means belonging to the second class equal to some fix $m_2 \in \mathbb{R}^d$. We set $m_1 = 0 \in \mathbb{R}^d$, $m_2 = 10h$, with $h = [h_1,\ldots,h_d]^T$, $h_i \sim \mathcal{U}([0,1])$, $i = 1,\ldots,d$. The results for $[r_1,r_2] = [-3,5]$ and $[r_1,r_2] = [0,5]$ are presented in Tables 3 and 4, respectively. The same settings as previously described are kept for the $l = 2$ case. The experiments for the case where the means of the Gaussians are set to zero are presented in Tables 5 and 6, while those where the means of Gaussians are non-zero are presented in Tables 7 and 8, respectively.

Table 1. Results in the form of recognition accuracy, obtained on the synthetic data: $l = 1$, $t \in [-3,5]$, $N = 800$, $M = 200$, $m_1, m_2 = 0$.

Type of Measures	$K = 1$				$K = 5$			
	$d = 10$	$d = 20$	$d = 30$	$d = 50$	$d = 10$	$d = 20$	$d = 30$	$d = 50$
$GMM - NPE_1$	0.72	0.77	0.83	0.87	0.87	0.95	0.96	0.96
$GMM - NPE_2$	0.74	0.76	0.85	0.87	0.86	0.96	0.94	0.94
$GMM - NPE_3$	0.78	0.79	0.87	0.90	0.87	0.98	0.95	0.96
KL_{WE}	0.62	0.68	0.79	0.73	0.92	0.86	0.90	0.94
KL_{MB}	0.62	0.68	0.79	0.73	0.91	0.87	0.92	0.93
KL_{VAR}	0.62	0.68	0.79	0.73	0.91	0.87	0.92	0.91

Table 2. Results in the form of recognition accuracy, obtained on the synthetic data: $l = 1$, $t \in [0,5]$, $N = 800$, $M = 200$, $m_1, m_2 = 0$.

Type of Measures	$K = 1$				$K = 5$			
	$d = 10$	$d = 20$	$d = 30$	$d = 50$	$d = 10$	$d = 20$	$d = 30$	$d = 50$
$GMM - NPE_1$	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
$GMM - NPE_2$	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
$GMM - NPE_3$	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
KL_{WE}	0.92	0.92	0.94	0.95	0.99	1.0	0.98	0.99
KL_{MB}	0.92	0.93	0.94	0.95	1.0	0.98	1.0	0.97
KL_{VAR}	0.92	0.93	0.95	0.95	1.0	1.0	0.98	0.97

Table 3. Results in the form of recognition accuracy, obtained on the synthetic data: $l = 1$, $t \in [-3, 5]$, $N = 800$, $M = 200$, $m_1, m_2 = 0$.

Type of Measures	$K = 1$				$K = 5$			
	$d = 10$	$d = 20$	$d = 30$	$d = 50$	$d = 10$	$d = 20$	$d = 30$	$d = 50$
$GMM - NPE_1$	0.73	0.98	0.97	0.98	0.99	1.0	0.97	0.98
$GMM - NPE_2$	0.72	0.97	0.97	0.96	0.97	1.0	0.99	0.99
$GMM - NPE_3$	0.77	1.0	1.0	1.0	1.0	1.0	1.0	1.0
KL_{WE}	0.28	0.38	0.35	0.42	0.46	0.43	0.33	0.41
KL_{MB}								
KL_{VAR}	0.63	1.0	0.98	1.0	1.0	0.98	0.99	1.0

Table 4. Results in the form of recognition accuracy, obtained on the synthetic data: $l = 1$, $t \in [0, 5]$, $N = 800$, $M = 200$, $m_1 = 0$, $m_2 = 10h$, with $h = [h_1, \ldots, h_d]^T$, $h_i \sim \mathcal{U}([0, 1])$, $i = 1, \ldots, d$.

Type of Measures	$K = 1$				$K = 5$			
	$d = 10$	$d = 20$	$d = 30$	$d = 50$	$d = 10$	$d = 20$	$d = 30$	$d = 50$
$GMM - NPE_1$	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
$GMM - NPE_2$	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
$GMM - NPE_3$	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
KL_{WE}	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
KL_{MB}	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
KL_{VAR}	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 5. Results in the form of recognition accuracy, obtained on the synthetic data: $l = 2$, $t_1, t_2 \in [-3, 5]$, $N = 800$, $M = 200$, $m_1, m_2 = 0$.

Type of Measures	$K = 1$				$K = 5$			
	$d = 10$	$d = 20$	$d = 30$	$d = 50$	$d = 10$	$d = 20$	$d = 30$	$d = 50$
$GMM - NPE_1$	0.82	0.84	0.85	0.98	0.99	0.97	0.98	0.99
$GMM - NPE_2$	0.81	0.83	0.85	0.98	0.98	0.97	0.98	0.98
$GMM - NPE_3$	0.84	0.86	0.87	1.0	1.0	1.0	1.0	1.0
KL_{WE}	0.78	0.76	0.75	0.95	0.97	0.94	0.94	0.93
KL_{MB}	0.78	0.76	0.75	0.83	0.94	0.97	0.95	0.94
KL_{VAR}	0.78	0.76	0.75	0.83	0.95	0.97	0.95	0.96

Table 6. Results in the form of recognition accuracy, obtained on the synthetic data: $l = 2$, $t_1, t_2 \in [0, 5]$, $N = 800$, $M = 200$, $m_1, m_2 = 0$.

Type of Measures	$K = 1$				$K = 5$			
	$d = 10$	$d = 20$	$d = 30$	$d = 50$	$d = 10$	$d = 20$	$d = 30$	$d = 50$
$GMM - NPE_1$	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
$GMM - NPE_2$	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
$GMM - NPE_3$	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
KL_{WE}	1.0	0.98	0.97	0.99	0.98	1.0	1.0	1.0
KL_{MB}	1.0	0.98	0.97	0.99	0.97	0.99	1.0	1.0
KL_{VAR}	1.0	0.98	0.97	0.99	0.99	0.98	1.0	1.0

Table 7. Results in the form of recognition accuracy, obtained on the synthetic data: $l = 2$, $t_1, t_2 \in [-3, 5]$, $N = 800$, $M = 200$, $m_1 = 0$, $m_2 = 10h$, with $h = [h_1, \ldots, h_d]^T$, $h_i \sim \mathcal{U}([0, 1])$, $i = 1, \ldots, d$.

Type of Measures	$K = 1$				$K = 5$			
	$d = 10$	$d = 20$	$d = 30$	$d = 50$	$d = 10$	$d = 20$	$d = 30$	$d = 50$
$GMM - NPE_1$	0.92	0.93	0.92	0.89	0.95	0.98	0.98	0.99
$GMM - NPE_2$	0.93	0.93	0.92	0.90	0.97	0.98	1.0	1.0
$GMM - NPE_3$	0.95	0.95	0.95	0.92	0.99	1.0	1.0	1.0
KL_{WE}	0.94	0.94	0.84	0.85	0.96	0.99	0.97	0.98
KL_{MB}	0.86	0.86	0.90	0.76	1.0	0.96	0.99	1.0
KL_{VAR}	0.86	0.85	0.84	0.87	0.98	0.98	0.98	0.97

Table 8. Results in the form of recognition accuracy, obtained on the synthetic data: $l = 2$, $t_1, t_2 \in [0, 5]$, $N = 800$, $M = 200$, $m_1 = 0$, $m_2 = 10h$, with $h = [h_1, \ldots, h_d]^T$, $h_i \sim \mathcal{U}([0, 1])$, $i = 1, \ldots, d$.

Type of Measures	$K = 1$				$K = 5$			
	$d = 10$	$d = 20$	$d = 30$	$d = 50$	$d = 10$	$d = 20$	$d = 30$	$d = 50$
$GMM - NPE_1$	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
$GMM - NPE_2$	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
$GMM - NPE_3$	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
KL_{WE}	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
KL_{MB}	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
KL_{VAR}	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

It can be seen from all the experiments that the recognition accuracy of all three proposed measures is higher than or equal to the recognition accuracy of the baseline measures, while the computational complexity is largely in favor of the proposed measures. Namely, the computational complexity for all baseline measures is $O(K^2 d^3)$, with $d \in \{10, 20, 30, 50\}$, while it is $O(K^2 \hat{l}) + O(K d^2 \hat{l})$, with $\hat{l} \in \{1, 2\}$ estimated l, where we obtained, as we mentioned $\hat{l} = l$, for $l \in \{1, 2\}$. Thus, one could observe that it is largely in favor of all the proposed measures in comparison to all the baseline ones, in all cases.

For the second scenario, $N = 800$ positive-definite matrices A_i are sampled, each one formed in a similar way, previously described for A_1. Thus, we control the sampling process in order to obtain positive-definite matrices "uniformly" distributed in the cone $\mathcal{SPD}_+(d)$, i.e., not lying on any lower dimensional embedded sub-manifold. The set of Gaussians is formed using the set of positive-definite matrices, while all means are set to zero vectors. $\tilde{N}$ different GMMs of size K are formed, their components sampled uniformly from the above-mentioned set of Gaussians ($N = 800$, $\tilde{N} = 200$ and $K = 5$ in the experiment). The proposed GMM-NPE$_i$ ($i = 1, 2, 3$) performs equally well, concerning the recognition precision as well as computational efficiency in comparison to all baseline methods. Estimated number of the non-negligible characteristic values were equal to the dimension of the full space. All the above-mentioned confirms that if data do not lye on the lower dimensional manifold embedded in the cone $\mathcal{SPD}_+(d)$, the proposed method does not provide any benefits in comparison to the baseline methods.

5.2. Experiments on Real Data

In this section, the performances of the proposed method described in Section 4.2, evaluated on real data (texture recognition task), are presented in comparison to baseline methods. As the baseline, we use KL-based KL$_{WE}$, KL$_{MB}$, and KL$_{VAR}$ GMM similarity measures, all described in Section 2. As the baseline, we also use the unsupervised sparse EMD-based measure proposed in [3], denoted by SR-EMD-M measure as well as the supervised sparse EMD-based measure, also proposed in [3], denoted by SR-EMD-M-L.

For a texture recognition task, we conducted experiments on the following databases: UMD [41], containing 25 classes (1000 images); CUReT [42], containing 61 classes (5612 images); KTH-TIPS [43], containing 10 classes (8010 images).

We used covariance descriptors as texture features (see [34,44,45]) in the experiments, as they showed excellent performance in the texture recognition task. We briefly explain how they were formed: For any given textured image, the row features are calculated in a form $[I, |I_x|, |I_y|, |I_{xx}|, |I_{yy}|](x, y)$ (the actual dimension of the vector is $\tilde{d} = 5$), from whom, extracted at the $R \times R$ patch (we used $R = 30$ in all experiments), centered in (x, y), we estimate the covariance matrix, and then finally vectorize its upper triangular into one $d = \tilde{d}(\tilde{d}+1)/2 = 15$ dimensional feature vector. For that particular textured image, the parameters of GMMs are estimated using EM [46] on the pool of feature vectors obtained as previously explained. We note that for every train or test image example, we uniformly divide it into four sub-images and those are used for training/testing. Hence, each image is represented by four GMMs and compared to all GMMs in the training set, while its label is determined using the kNN algorithm ($k = 3$ and class label is obtained by voting). Recognition accuracy of the proposed GMM-NPE measures, in comparison to all baseline measures, for the above-mentioned texture databases, are presented in Figures 1–3. For all databases used, we vary from $l = 30$ to $l = 100$ in order to analyze the trade-off between accuracy and computational efficiency. We kept the number of Gaussian components fixed and equal to $K = 5$. For each class, a fixed number of N examples from the training set is randomly selected (by uniform distribution), keeping the rest for testing. We vary the mentioned number of training instances N across experiments. Final results are averaged over 20 trials. In all experiments, we obtained slightly better results using the GMM-NPE$_2$ measure defined by (18), (19) in comparison to the GMM-NPE$_1$, so we present only the GMM-NPE$_2$ and GMM-NPE$_3$ in our results.

Recall that (see the analysis presented in Section 4.3) the computational complexity of the proposed GMM-NPE$_1$ (as well as GMM-NPE$_2$) is roughly $O(K^2 l) + O(Kd^2 l)$, and for all the KL-based baseline algorithms, i.e., the KL$_{WE}$, KL$_{MB}$, and KL$_{VAR}$, the computational complexity is estimated roughly as $O(K^2 d^3)$. Furthermore, (see Section 4.3), for the EMD-based baseline algorithms, i.e., the EMD-KL and SR-EMD-M, the computational complexity is estimated roughly as $O(8K^5 d^3)$ and $O(k_{iter} K^4 d^3)$, respectively, with $k_{iter} >> K$ (see [3]). It follows that the ratio between the computational complexity of the proposed GMM-NPE$_1$ and GMM-NPE$_2$, and any mentioned baseline KL-based method is estimated roughly as $l/d^3 + l/(Kd)$, while the ratios between the computational complexity of GMM-NPE$_1$ and GMM-NPE$_2$, and the baseline EMD-based measures are estimated as $\frac{l_{max}}{k_{iter} K^2 d^3} + \frac{l_{max}}{8K^4 d}$ and $\frac{l_{max}}{8K^3 d^3} + \frac{l_{max}}{k_{iter} K^3 d}$, respectively. Considering $l_{max} \ll d$ and $k_{iter} >> K$, it can be seen that the computational efficiency is largely in favor of the proposed GMM-NPE$_1$, GMM-NPE$_2$ in comparison to all baseline measures in all experimental cases. Concerning the proposed EMD-based GMM-NPE$_3$ measure with its computational complexity roughly estimated as $O(k_{iter} K^4 l) + O(Kd^2 l)$ (see Section 4.3), we compare its computational complexity with the corresponding EMD-based baseline EMD-KL and SR-EMD-M measures. The complexity ratios are estimated as $\frac{k_{iter} l_{max}}{8Kd} + \frac{l_{max}}{8K^3 d}$ and $\frac{l_{max}}{d} + \frac{l_{max}}{k_{iter} K^3 d}$, again largely in favor of the proposed GMM-NPE$_3$ measure, in comparison to the EMD-KL and SR-EMD-M measures, and especially for smaller values of l. Thus, we conclude that the trade-off between the recognition accuracy and the computational efficiency is in favor of the proposed GMM-NPE measures.

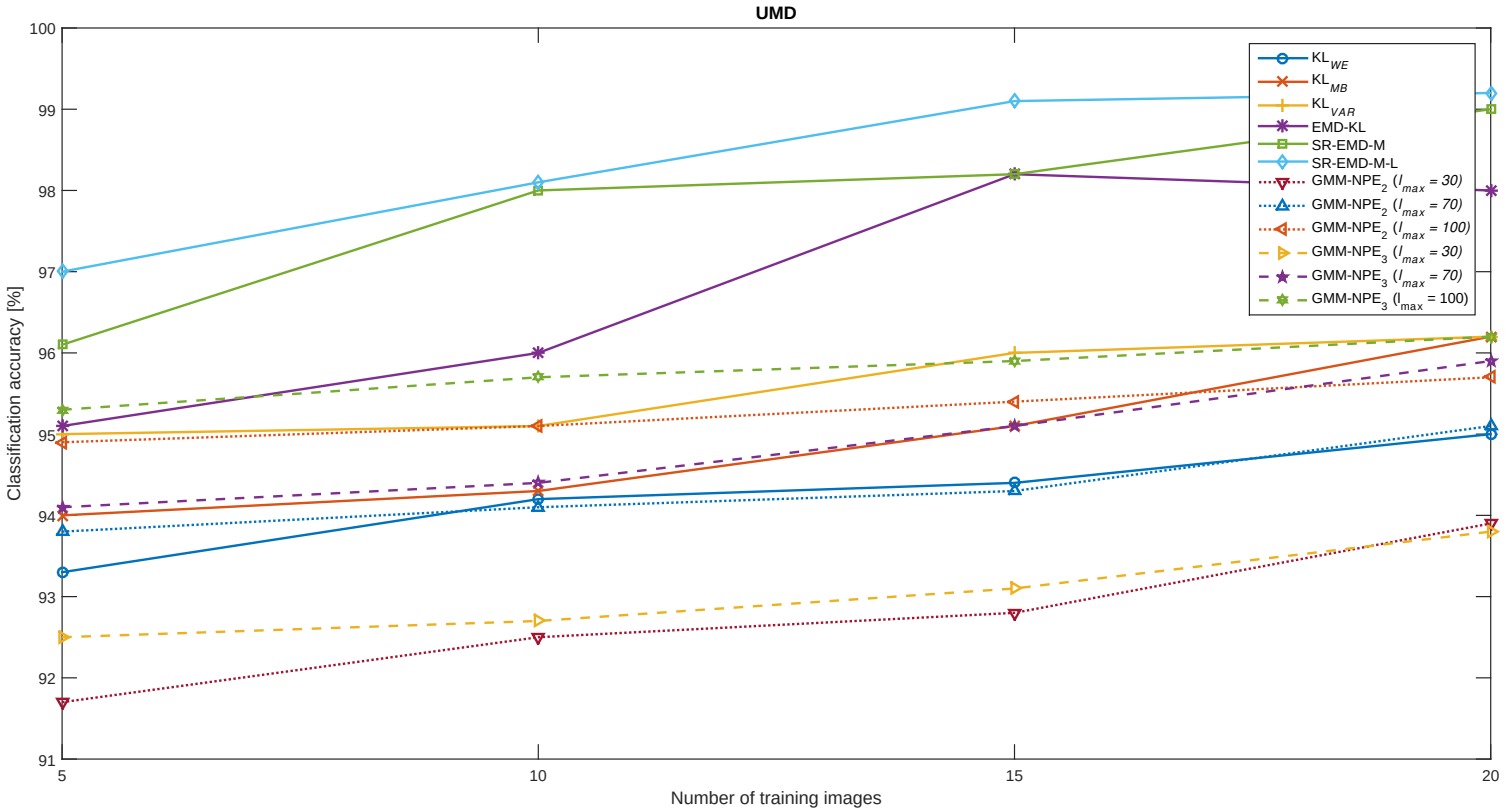

Figure 1. Classification rate vs. the number of training examples for the UMD texture database for the proposed method in comparison to baseline methods.

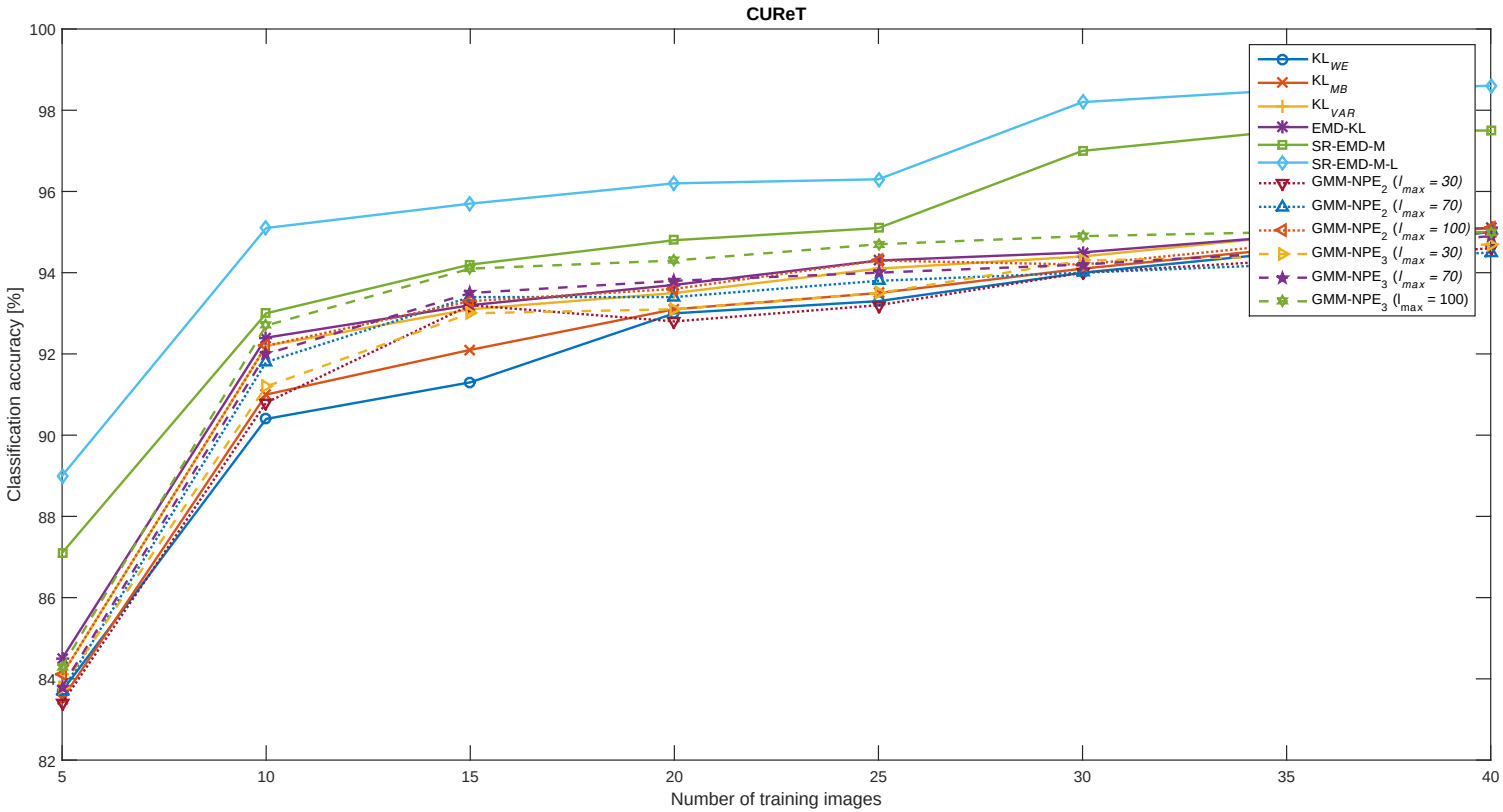

Figure 2. Classification rate vs. the number of training examples for the CUReT texture database for the proposed method in comparison to baseline methods.

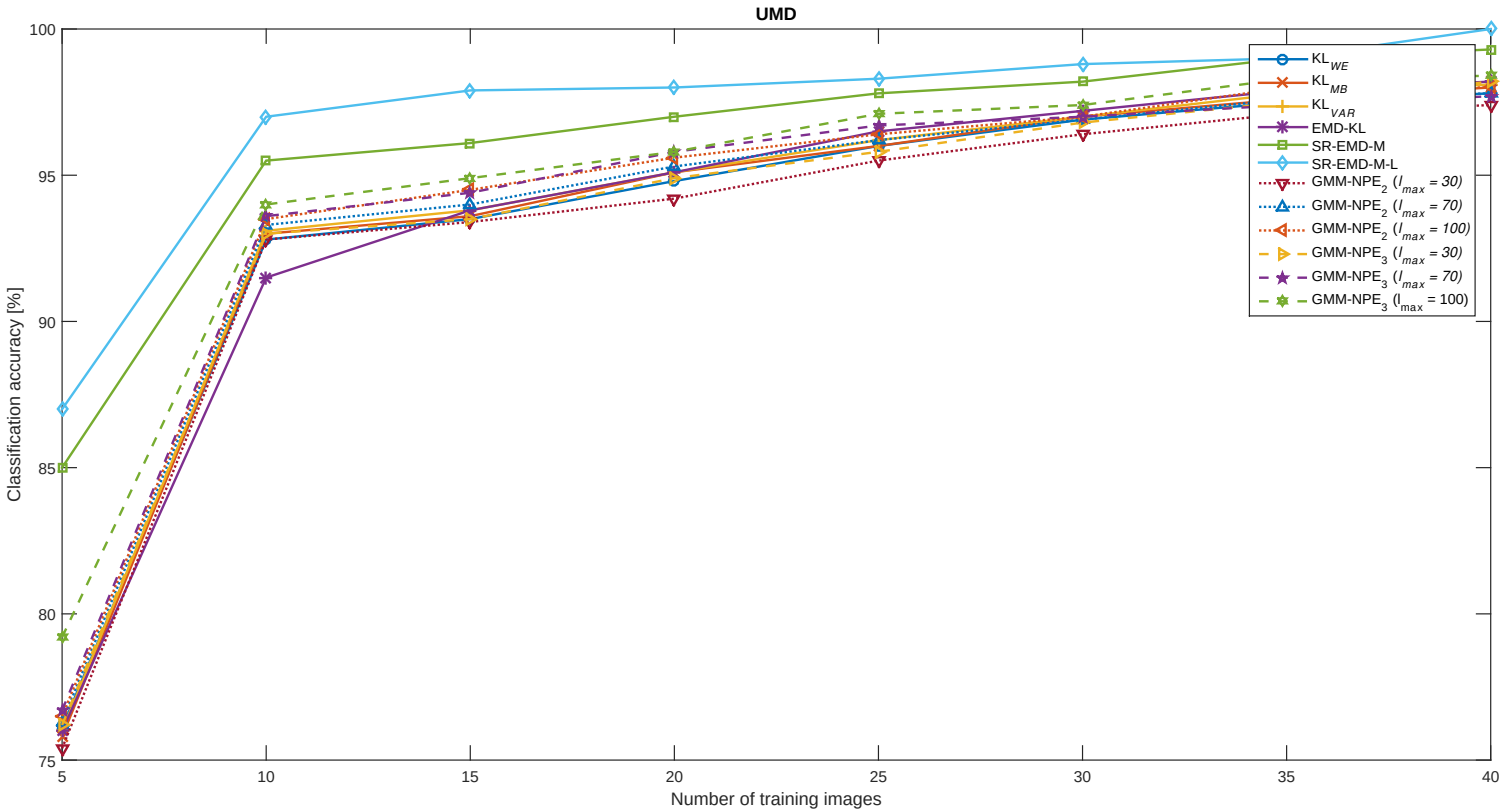

Figure 3. Classification rate vs. the number of training examples for the KTH-TIPS texture database for the proposed method in comparison to baseline methods.

In Table 9, CPU processing times are presented for the proposed GMM-NPE$_1$ and GMM-NPE$_2$ measures, in comparison to the baseline KL$_{WE}$, KL$_{MB}$, KL$_{VAR}$, and EMD-KL measures. The results are obtained as CPU processing times needed for the evaluation of measures of similarity between two GMMs in 100 trials. All GMMs are learned using randomly chosen example images from KTH-TIPS texture classification database. For all the experiments, we set $k_{iter} = 20$, $d = 15$ and $l_{max} = 30$, $l_{max} = 70$, or $l_{max} = 100$. It can be seen that the proposed GMM-NPE measure provides significantly lower CPU processing times in comparison to all baseline measures when there is a significant reduction in dimensionality of the original parameter space, i.e., $l_{max} = 30$ and $l_{max} = 60$, where the original Euclidian parameter space is of dimension $n = 120$. However, in the case of a relatively insignificant reduction in dimensionality, i.e., $l_{max} = 100$, the performances in terms of computational complexity deteriorate significantly for the GMM-NPE measures. These results are consistent concerning the computational bounds given for the proposed and the baseline measures given in Section 4.3. The experiments were conducted on a workstation equipped with one 2.3 GHz CPU and 6 GB RAM.

Table 9. Average processing CPU times for the proposed GMM-NPEs, in comparison to the baseline measures, as a function of number of GMM components K used, as well as dimension of the reduced space l_{max} (unit: [ms]).

K	**5**			**10**			**15**			**20**		
KL$_{WE}$	17.6			70.5			159.2			282.3		
KL$_{MB}$	14.7			80.1			187.3			323.4		
KL$_{VAR}$	32.9			128.0			297.5			528.3		
EMD-KL	49.3			1987			15102			61123		
l_{max}	30	60	100	30	60	100	30	60	100	30	60	100
$GMM-NPE_1$	7.2	14.4	23.9	14.7	29.6	49.3	22.1	46.2	74.8	30.8	62.1	101.6
$GMM-NPE_2$	7.4	14.7	24.2	14.9	30.2	49.6	22.3	46.5	74.9	31.1	62.4	101.9

The proposed methodology could also be applied in realistic personalization and recommendation application scenarios presented in [47]. Namely, user profile features obtained in this process could store history over time, and therefore, the covariance matrix could be estimated in the learning phase. The transformation matrix could be formed as presented in Sections 3 and 4.1, and the covariance which represents any particular user could be projected and represented by low dimensional vector representatives. In the exploitation phase, stored features collected from users in some predefined period of time could also be used in order to form covariances which could then be projected. The measure of similarity between a user and item could then be computed by using similarity measures and the procedure proposed in Section 4.2.

Author Contributions: Conceptualization, M.J., L.K.; data curation, B.P.; formal analysis, L.K.; funding acquisition, L.C., R.C.; investigation, M.J., R.C.; methodology M.J., L.K.; project administration, L.C.; software, B.P.; supervision, L.C., R.C.; visualization, L.K.; writing—original draft, M.J., L.K.; Writing—review and editing, L.K., B.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Science Fund of the Republic of Serbia grant number #6524560, and Serbian Ministry of Education, Science and Technological Development grant number 45103-68/2020-14/200156.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research was supported by the Science Fund of the Republic of Serbia, #6524560, AI-S-ADAPT, and by the Serbian Ministry of Education, Science and Technological Development through the project no. 451 03-68/2020-14/200156: "Innovative Scientific and Artistic Research from the Faculty of Technical Sciences Activity Domain".

Conflicts of Interest: The authors declare that there is no conflict of interest.

References

1. Goldberger, J.; Aronowitz, H. A Distance measure Between GMMs Based on the Unscented Transform and its Application to Speaker Recognition. In Proceedings of the INTERSPEECH 1985–1988, Lisbon, Portugal, 4–8 September 2005.
2. Goldberger, J.; Gordon, S.; Greenspan, H. An Efficient Image Similarity Measure based on Approximations of KL-Divergence Between Two Gaussian Mixtures. In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; Volume 1, pp. 487–493.
3. Li, P.; Wang, Q.; Zhang, L. A Novel Earth Mover Distance Methodology for Image Matching with Gaussian Mixture Models. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 1689–1696.
4. Krstanović, L.; Ralević, N.M.; Zlokolica, V.; Obradović, R.; Mišković, D.; Janev, M.; Popović, B. GMMs Similarity Measure Based on LPP-like Projection of the Parameter Space. *Expert Syst. Appl.* **2016**, *66*, 136–148. [CrossRef]
5. Beecks, C.; Ivanescu, A.M.; Kirchhoff, S.; Seidl, T. Modeling image similarity by Gaussian mixture models and the signature quadratic form distance. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011.
6. Wu, Y.; Chan, K.L.; Huang, Y. Image texture classification based on finite Gaussian mixture models. In Proceedings of the Workshop on Texture Analysis and Synthesis, ICCV, Nice, France, 17 October 2003; pp. 107–112.
7. Hao, H.; Wang, Q.; Li, P.; Zhang, L. Evaluation of ground distances and features in EMD-based GMM matching for texture classification. *Pattern Recognit.* **2016**, *57*, 152–163. [CrossRef]
8. Lazebnik, S.; Schmid, C.; Ponce, J. A Sparse Texture Representation Using Local Affine Regions. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1265–1278. [CrossRef]
9. Dilokthanakul, N.; Mediano, P.A.M.; Garnelo, M.; Lee, M.C.H.; Salimbeni, H.; Arulkumaran, K.; Shanahan, M. Deep unsupervised clustring with Gaussian mixture variational autoencoders. *arXiv* **2017**, arXiv:1611.02648v2.
10. Durrieu, J.-L.; Thiran, J.-P.; Kelly, F. Lower and upper bounds for approximation of the Kullback-Leibler divergence between Gaussian mixture models. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 4833–4836.
11. Chernoff, H. A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the sum of Observations. *Ann. Math. Stat.* **1952**, *23*, 493–507. [CrossRef]
12. Bhattacharyya, A. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.* **1943**, *35*, 99–109.
13. Matusita, K. Decision rules based on the distance for problems of fit two samples and estimation. *Ann. Math. Stat.* **1955**, *26*, 631–640. [CrossRef]
14. Kullback, S. *Information Theory and Statistics*; Dover Publications Inc.: Mineola, NY, USA, 1968.
15. Kristan, M.; Leonardis, A.; Skočaj, D. Multivariate online kernel density estimation with Gaussian kernels. *Pattern Recognit.* **2011**, *44*, 2630–2642. [CrossRef]
16. Lovric, M.; Min-Oo, M.; Ruh, E.A. Multivariate normal distributions parametrized as a Riemannian symmetric space. *JMVA* **2000**, *74*, 36–48. [CrossRef]
17. Rubner, Y.; Tomasi, C.; Guibas, L. The Earth Mover's Distance as a Metric for Image Retrieval. *Int. J. Comput. Vis.* **2000**, *40*, 99–121. [CrossRef]
18. Ling, H.; Okada, K. An efficient Earth Movers Distance algorithm for robust histogram comparison. *PAMI* **2007**, *29*, 840–853 [CrossRef]
19. Belhumeur, P.; Hepanha, J.; Kriegman, D. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 711–720. [CrossRef]
20. Liu, J.; Chen, S.; Tan, X.; Zhang, D. Comments on efficient and robust feature extraction by maximum margin criterion. *IEEE Trans. Neural Netw.* **2007**, *18*, 1862–1864.
21. Turk, M.; Pentland, A. Eigenfaces for recognition. *J. Cogn. Neurosci.* **1991**, *3*, 71–86. [CrossRef]
22. Tenenbaum, J. Mapping a manifold of perceptual observations. *Adv. Neural Inf. Process. Syst.* **1998**, *10*, 682–688.
23. Belkin, M.; Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. *NIPS* **2001**, *14*, 585–591.
24. He, X.; Niyogi, P. Locality preserving projections, Proceedings of Conference on Advances in Neural Information Processing Systems. *NIPS* **2003**, *16*, 153–160.
25. Roweis, S.; Saul, L. Nonlinear dimensionality reduction by locally linear embedding. *Science* **2000**, *290*, 2323–2326. [CrossRef]
26. He, X.; Cai, D.; Yan, S.; Zhang, H.J. Neighborhood Preserving Embedding. In Proceedings of theTenth IEEE International Conference on Computer Vision ICCV'05, Beijing, China, 17–21 October 2005; Volume 1, pp. 17–21.
27. Goh, A.; Vidal, R. Clustering and Dimensionality Reduction on Riemannian Manifolds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR, Anchorage, AK, USA, 23–28 June 2008; pp. 1–7.

28. Li, Z.; Chen, J.; Fu, Y.; Hu, G.; Pan, Z.; Zhang, L. Community Detection Based on Regularized Semi-Nonnegative Matrix Tri-Factorization in Signed Networks. *Mob. Netw. Appl.* **2018**, *23*, 71–79. [CrossRef]
29. Gao, Z.; Wu, Y.; Bu, X.; Yu, T.; Yuang, J.; Jia, Y. Learning a robust representation via a deep network on symmetric positive definite manifolds. *Pattern Recognit.* **2019**, *92*, 1–12. [CrossRef]
30. Huang, X.; Ma, X.; Hu, F. Machine Learning and Intelligent Communications. *Mob. Netw. Appl.* **2018**, *23*, 68–70. [CrossRef]
31. Hershey, J.R.; Olsen, P.A. Aproximating the Kullback Leibler divergence between Gaussian mixture models. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, Honolulu, HI, USA, 15–20 April 2007; Volume 4, pp. IV-317–IV-320.
32. Cover, T.; Thomas, J. *Elements of Information Theory*, 2nd ed.; Wiley Series in Telecommunications; John Wiley and Sons: New York, NY, USA, 1991.
33. Julier, S.; Uhlmann, J.K. *A General Method for Approximating Nonlinear Transformations of Probability Distributions*; Technical Report; RRG, Department of Engineering Science, University of Oxford: Oxford, UK, 1996.
34. Sivalingam, R.; Boley, D.; Morellas, V.; Papanikolopoulos, N. Tensor Sparse Coding for Region Covariances. In *Computer Vision, ECCV*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 722–735.
35. Vandenberghe, L.; Boyd, S.; Wu, S. Determinant Maximization with Linear Matrix Inequality Constraints. *SIAM J. Matrix Anal. Appl.* **1998**, *19*, 499–533. [CrossRef]
36. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambrige University Press, The Edimbrg Building: Cambridge, UK, 2009.
37. Grant, M.; Boyd, S. CVX: Matlab Software for Disciplined Convex Programming. Ver. 1.21. 2010. Available online: http://cvxr.com/cvx (accessed on 1 January 2020).
38. Klir, G.J.; Yuan, B. *Fuzzy Sets and Fuzzy Logic: Theory and Applicaions*; Academic Press: London, UK, 1995.
39. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistial Learning*; Springer: Berlin/Heidelberg, Germany, 2009.
40. Burden, R.L.; Faires, J.D. *Numerical Analysis*, 9th ed.; Springer: London, UK, 2010.
41. Xu, Y.; Ji, H.; Fermuller, C. A projective invariant for texture. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; pp. 1932–1939.
42. Dana, K.J.; Nayar, S.K.; Van Ginneken, B.; Koenderink, J.J. Reflectance and texture of real-world surfaces. *ACM TOG* **1999**, *18*, 1–34. [CrossRef]
43. Hayman, E.; Caputo, B.; Fritz, M.; Eklundh, J.O. On the significance of real-world conditions for material classification. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3024.
44. Jayasumana, S.; Hartley, R.; Salzmann, M.; Li, H.; Harandi, M. Kernel Methods on the Riemannian Manifold of Symmetric Positive Definite Matrices. In Proceedings of theIEEE Computer Vision and Pattern Recognition, CVPR, Portland, OR, USA, 23–28 June 2013; pp. 73–80.
45. Zhang, Y.; Jiang, Z.; Davis, L.S. Discriminative Tensor Sparse Coding for Image Classification. In Proceedings of the British Machine Vision Conference, Bristol, UK, 9–13 September 2013. [CrossRef]
46. Webb, A.R. *Statistical Pattern Recognition*, 2nd ed.; John Wiley and Sons: New York, NY, USA, 2002.
47. Stai, E.; Kafetzoglou, S.; Tsiropoulou, E.E.; Papavassiliou, S. A holistic approach for personalization, relevance feedback & recommendation in enriched multimedia content, Multimed. *Tools Appl.* **2018**, *77*, 283–326. [CrossRef]