ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10094719

HEMPPCAT: MIXTURES OF PROBABILISTIC PRINCIPAL COMPONENT ANALYSERS FOR DATA WITH HETEROSCEDASTIC NOISE

Alec S. Xu, Laura Balzano, Jeffrey A. Fessler

EECS Department, University of Michigan-Ann Arbor

ABSTRACT

Mixtures of probabilistic principal component analysis (MPPCA) is a well-known mixture model extension of principal component analysis (PCA). Similar to PCA, MPPCA assumes the data samples in each mixture contain homoscedastic noise. However, datasets with heterogeneous noise across samples are becoming increasingly common, as larger datasets are generated by collecting samples from several sources with varying noise profiles. The performance of MPPCA is suboptimal for data with heteroscedastic noise across samples. This paper proposes a heteroscedastic mixtures of probabilistic PCA technique (HeMPPCAT) that uses a generalized expectation-maximization (GEM) algorithm to jointly estimate the unknown underlying factors, means, and noise variances under a heteroscedastic noise setting. Simulation results illustrate the improved factor estimates and clustering accuracies of HeMPPCAT compared to MPPCA.

Index Terms— Heterogeneous data, latent factors, expectation maximization.

1. INTRODUCTION

PCA is a well-known unsupervised dimensionality reduction method for high-dimensional data analysis. It has been extended to capture a mixture of low-dimensional affine subspaces. When this mixture model is derived through a probabilistic perspective, it is called Mixtures of Probabilistic PCA (MPPCA) [1]. MPPCA models are a statistical extension of union-of-subspace models [2, 3, 4] and are also related to subspace clustering methods [5, 6]. One can apply MPPCA to many engineering and machine learning tasks, such as image compression and handwritten digit classification.

However, a limitation of PPCA and MPPCA is that they both model the noise as independent and identically distributed (IID) with a shared variance, i.e., homoscedastic. Consequently, the performance of PPCA and MPPCA can be suboptimal for heteroscedastic noise conditions. Heterogeneous datasets are increasingly common, e.g., when combining samples from several sources [7], or samples collected under varying ambient conditions [8]. Recently an algorithm called HePPCAT was developed to extend PPCA to data with heteroscedastic noise across samples [9], but no corresponding methods exist for a *mixture* of PPCA models. This paper generalizes MPPCA to introduce a **He**teroscedastic **MPPCA** **T**echnique (HeMPPCAT): an MPPCA method for data with heteroscedastic noise. This paper presents the statistical data model, a GEM algorithm, and results with both synthetic data and motion segmentation from real video data.

Supported in part by NSF Grant IIS 1838179.

2. RELATED WORK

2.1. MPPCA

The original MPPCA approach [1] models n data samples in $\mathbb{R}^d$ as arising from J affine subspaces:

$$\boldsymbol{y}_i = \boldsymbol{F}_j \boldsymbol{z}_i + \boldsymbol{\mu}_j + \boldsymbol{\epsilon}_i \tag{1}$$

for all $i \in \{1, \ldots, n\}$ and some class index $j = j_i \in \{1, \ldots, J\}$. Here $\boldsymbol{F}_1, \ldots, \boldsymbol{F}_J \in \mathbb{R}^{d \times k}$ are deterministic factor matrices to estimate, $\boldsymbol{z}_i \sim \mathcal{N}(\mathbf{0}_k, \boldsymbol{I}_k)$ are IID factor coefficients, $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_J \in \mathbb{R}^d$ are unknown deterministic mean vectors to estimate, $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}_d, v_j \boldsymbol{I}_d)$ are IID noise vectors, and $v_1, \ldots, v_J$ are unknown noise variances to estimate.

MPPCA assumes all samples from mixture component j have the same noise variance v_j. In contrast, our proposed HeMPPCAT allows each sample from mixture j to come from one of $L \leq n$ noise groups, where $L \neq J$ in general.

2.2. Probabilistic PCA for Heteroscedastic Signals

The authors of [10] developed a Riemannian optimization method to perform an MPPCA-like method on signals with heterogeneous power levels. They model n data samples in $\mathbb{C}^d$ as

$$\boldsymbol{y}_i = \sqrt{\tau_j}\boldsymbol{U}_j\boldsymbol{z}_i + \boldsymbol{\epsilon}_i \tag{2}$$

for all $i = \{1, \ldots, n\}$ and some $j = j_i \in \{1, \ldots, J\}$. The signal powers $\tau_1, \ldots, \tau_J \in \mathbb{R}^+$, known as *signal textures*, are factors to estimate, $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_J \in \mathrm{St}_{d,k}$ are orthonormal subspace bases to estimate ($\mathrm{St}_{d,k}$ denotes the $d \times k$ Stiefel manifold), $\boldsymbol{z}_i \sim \mathbb{C}\mathcal{N}(\boldsymbol{0}_k, \boldsymbol{I}_k)$ are IID subspace coefficients, and $\boldsymbol{\epsilon}_i \sim \mathbb{C}\mathcal{N}(\boldsymbol{0}_d, \boldsymbol{I}_d)$ are IID noise vectors.

The model (2) assumes all signals in subspace j have the same texture value τ_j. That assumption is somewhat analogous to how the MPPCA model assumes all samples in mixture j have the same noise variance v_j. Our proposed HeMPPCAT model instead allows samples in the same mixture component to have different noise variances, and allows different signal components to have different signal strengths, rather than a common scaling factor $\sqrt{\tau}$.

3. HEMPPCAT DATA MODEL

We assume there are $n_1 + \cdots + n_L = n$ data samples in $\mathbb{R}^d$ from L different noise groups with model

$$\boldsymbol{y}_{\ell,i} = \boldsymbol{F}_j\boldsymbol{z}_{\ell,i} + \boldsymbol{\mu}_j + \boldsymbol{\epsilon}_{\ell,i} \tag{3}$$

for all $i \in \{1, \ldots, n_\ell\}$, $\ell \in \{1, \ldots, L\}$, and some $j = j_i \in \{1, \ldots, J\}$. $\boldsymbol{F}_1, \ldots, \boldsymbol{F}_J \in \mathbb{R}^{d\times k}$ are unknown factor matrices to estimate (*not* constrained to the Stiefel manifold, so different signal components can have different amplitudes), $\boldsymbol{z}_{\ell,i} \sim \mathcal{N}(\boldsymbol{0}_k, \boldsymbol{I}_k)$ are IID coefficients, and $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_J \in \mathbb{R}^d$ are unknown mean vectors to estimate. To model heteroscedastic noise, we assume $\boldsymbol{\epsilon}_{\ell,i} \sim \mathcal{N}(\boldsymbol{0}_d, v_\ell\boldsymbol{I}_d)$, where $v_1, \ldots, v_L$ are unknown noise variances to estimate. Importantly, the noise model associates noise variance with data sample, regardless of the underlying affine subspace.

The joint log-likelihood of the samples is

$$\begin{aligned}
\mathcal{L}(\boldsymbol{F}, \boldsymbol{\mu}, \boldsymbol{v}, \boldsymbol{p}) &= \sum_{\ell=1}^{L}\sum_{i=1}^{n_\ell} \ln\{p(\boldsymbol{y}_{\ell,i})\} \\
&= \sum_{\ell=1}^{L}\sum_{i=1}^{n_\ell} \ln\left\{\sum_{j=1}^{J} \pi_j p(\boldsymbol{y}_{\ell,i} \mid j)\right\}, \quad (4)
\end{aligned}$$

$$\begin{aligned}
p(\boldsymbol{y}_{\ell,i} \mid j) &= (2\pi)^{-d/2}\det(\boldsymbol{C}_{\ell,j})^{-1/2}\exp(-E_{\ell ij}^2/2), \\
E_{\ell ij}^2 &= (\boldsymbol{y}_{\ell,i} - \boldsymbol{\mu}_j)^\top \boldsymbol{C}_{\ell,j}^{-1}(\boldsymbol{y}_{\ell,i} - \boldsymbol{\mu}_j), \\
\boldsymbol{C}_{\ell,j} &= \boldsymbol{F}_j\boldsymbol{F}_j^\top + v_\ell\boldsymbol{I}_d,
\end{aligned}$$

where $\boldsymbol{F} = [\boldsymbol{F}_1, \ldots, \boldsymbol{F}_J]$, $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_J]$, $\boldsymbol{v} = [v_1, \ldots, v_L]$, and $\boldsymbol{p} = [\pi_1, \ldots, \pi_J]$, with π_j being the jth *mixing proportion*, such that $\pi_j \geq 0$ and $\sum_j \pi_j = 1$. Conceptually, to draw samples from this model, one first picks a class j according to the categorical distribution with parameters $\boldsymbol{p}$, generates random coefficients to multiply with the factor matrix $\boldsymbol{F}_j$, and then adds noise, where the noise variance depends on the noise group.

4. GEM ALGORITHM

The joint log-likelihood (4) is nonconvex with respect to the parameters of interest. MPPCA [1] maximized a similar expression using an EM method. We derived a GEM algorithm to maximize (4) with respect to the parameters $\boldsymbol{F}$, $\boldsymbol{\mu}$, $\boldsymbol{v}$, and $\boldsymbol{p}$, using a *complete-data log-likelihood* formulation.

Let $\boldsymbol{z}_{\ell ij}$ denote the coefficients associated with mixture j for sample $\boldsymbol{y}_{\ell,i}$, and let $g_{\ell ij}$ be random variables that follow a categorical distribution, where $g_{\ell ij} = 1$ indicates "mixture j generated sample $\boldsymbol{y}_{\ell,i}$." Treating $\boldsymbol{z}_{\ell ij}$ and $g_{\ell ij}$ as missing data, the complete-data log-likelihood is

$$\begin{aligned}
\mathcal{L}_C(\boldsymbol{\theta}) &= \sum_{\ell=1}^{L}\sum_{i=1}^{n_\ell}\sum_{j=1}^{J} g_{\ell ij}\ln\{\pi_j p(\boldsymbol{y}_{\ell,i}, \boldsymbol{z}_{\ell ij})\}, \\
p(\boldsymbol{y}_{\ell,i}, \boldsymbol{z}_{\ell ij}) &= p(\boldsymbol{y}_{\ell,i}|\boldsymbol{z}_{\ell ij})p(\boldsymbol{z}_{\ell ij}), \quad (5)\\
\boldsymbol{y}_{\ell,i}|\boldsymbol{z}_{\ell ij} &\sim \mathcal{N}(\boldsymbol{F}_j\boldsymbol{z}_{\ell ij} + \boldsymbol{\mu}_j, v_\ell\boldsymbol{I}), \\
p(\boldsymbol{z}_{\ell ij}) &= (2\pi)^{-k/2}\exp\{-\frac{1}{2}\boldsymbol{z}_{\ell ij}^\top\boldsymbol{z}_{\ell ij}\},
\end{aligned}$$

where $\boldsymbol{\theta} = [\boldsymbol{F}, \boldsymbol{\mu}, \boldsymbol{v}, \boldsymbol{p}]$ is shorthand for all the parameters to estimate. The derivations for the GEM algorithm can be found in [11].

4.1. Expectation step

For the E-step of the GEM algorithm, we compute the expectation of (5) conditioned on the observed data samples. Ignoring irrelevant constants, one can show this expression is

$$\begin{aligned}
\left\langle \mathcal{L}_C(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) \right\rangle = \sum_{\ell=1}^{L}\sum_{i=1}^{n_\ell}\sum_{j=1}^{J} R_{\ell ij}^{(t)}\Big\{&\ln(\pi_j) - \frac{d}{2}\ln(v_\ell) \\
&- \frac{1}{2}\mathrm{tr}\Big(\langle \boldsymbol{z}_{\ell ij}\boldsymbol{z}_{\ell ij}^\top \rangle\Big) - \frac{1}{2v_\ell}\|\boldsymbol{y}_{\ell,i} - \boldsymbol{\mu}_j\|_2^2 \\
&+ \frac{1}{v_\ell}\langle \boldsymbol{z}_{\ell ij}\rangle^\top \boldsymbol{F}_j^\top(\boldsymbol{y}_{\ell,i} - \boldsymbol{\mu}_j) \\
&- \frac{1}{2v_\ell}\mathrm{tr}\Big(\boldsymbol{F}_j^\top\boldsymbol{F}_j\langle \boldsymbol{z}_{\ell ij}\boldsymbol{z}_{\ell ij}^\top\rangle\Big)\Big\}, \quad (6)
\end{aligned}$$

where $\boldsymbol{\theta}^{(t)} = [\boldsymbol{F}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{v}^{(t)}, \boldsymbol{p}^{(t)}]$ denotes the parameter estimates at iteration t, and $\langle\cdot\rangle$ denotes conditional expectation. The conditional moments $\langle \boldsymbol{z}_{\ell ij}\rangle$ and

$\langle \boldsymbol{z}_{\ell ij}\boldsymbol{z}_{\ell ij}^\top \rangle$ are

$$\begin{aligned} \langle \boldsymbol{z}_{\ell ij} \rangle &= \boldsymbol{M}_{\ell,j}^{(t)\,-1} \boldsymbol{F}_j^{(t)\,\top} (\boldsymbol{y}_{\ell,i} - \boldsymbol{\mu}_j^{(t)}), \\ \langle \boldsymbol{z}_{\ell ij}\boldsymbol{z}_{\ell ij}^\top \rangle &= v_\ell^{(t)} \boldsymbol{M}_{\ell,j}^{(t)\,-1} + \langle \boldsymbol{z}_{\ell ij} \rangle \langle \boldsymbol{z}_{\ell ij} \rangle^\top, \\ \boldsymbol{M}_{\ell,j}^{(t)} &= v_\ell^{(t)} \boldsymbol{I}_k + \boldsymbol{F}_j^{(t)\,\top} \boldsymbol{F}_j^{(t)}, \end{aligned} \tag{7}$$

and the conditional expectation of $g_{\ell ij}$, denoted as $R_{\ell ij}$, is the posterior *mixing responsibility* of mixture component j generating sample $\boldsymbol{y}_{\ell,i}$:

$$R_{\ell ij}^{(t)} = \frac{p(\boldsymbol{y}_{\ell,i} \mid j)\, \pi_j^{(t)}}{p(\boldsymbol{y}_{\ell,i})}. \tag{8}$$

The probabilities are evaluated at the current parameter estimates $\boldsymbol{\theta}^{(t)}$.

4.2. Maximization step

For the M-step, it appears impractical to maximize (6) over all parameters simultaneously, so we adopt a GEM approach [12] where we update subsets of parameters in sequence.

Maximizing (6) with respect to π_j, v_ℓ, $\boldsymbol{\mu}_j$, and $\boldsymbol{F}_j$, in that sequence, results in the following M-step update expressions:

$$\pi_j^{(t+1)} = \frac{1}{n} \sum_{\ell=1}^{L} \sum_{i=1}^{n_\ell} R_{\ell ij}^{(t)},$$

$$\begin{aligned} v_\ell^{(t+1)} = \frac{1}{d \sum_{i=1}^{n_\ell} \sum_{j=1}^{J} R_{\ell ij}^{(t)}} \Bigg[&\sum_{i=1}^{n_\ell} \sum_{j=1}^{J} R_{\ell ij}^{(t)} \|\boldsymbol{y}_{\ell,i} - \boldsymbol{\mu}_j^{(t)}\|_2^2 \\ &- 2 \sum_{i=1}^{n_\ell} \sum_{j=1}^{J} R_{\ell ij}^{(t)} \langle \boldsymbol{z}_{\ell ij} \rangle^\top \boldsymbol{F}_j^{(t)\,\top} (\boldsymbol{y}_{\ell,i} - \boldsymbol{\mu}_j^{(t)}) \\ &+ \sum_{i=1}^{n_\ell} \sum_{j=1}^{J} R_{\ell ij}^{(t)} \mathrm{tr}\Big(\langle \boldsymbol{z}_{\ell ij}\boldsymbol{z}_{\ell ij}^\top \rangle \boldsymbol{F}_j^{(t)\,\top} \boldsymbol{F}_j^{(t)} \Big) \Bigg], \end{aligned}$$

$$\boldsymbol{\mu}_j^{(t+1)} = \frac{\sum_{\ell=1}^{L} \sum_{i=1}^{n_\ell} \frac{R_{\ell ij}^{(t)}}{v_\ell^{(t+1)}} \Big(\boldsymbol{y}_{\ell,i} - \boldsymbol{F}_j^{(t)} \langle \boldsymbol{z}_{\ell ij} \rangle \Big)}{\sum_{\ell=1}^{L} \sum_{i=1}^{n_\ell} \frac{R_{\ell ij}^{(t)}}{v_\ell^{(t+1)}}},$$

$$\begin{aligned} \boldsymbol{F}_j^{(t+1)} = &\left(\sum_{\ell=1}^{L} \sum_{i=1}^{n_\ell} \frac{R_{\ell ij}^{(t)}}{v_\ell^{(t+1)}} (\boldsymbol{y}_{\ell,i} - \boldsymbol{\mu}_j^{(t+1)}) \langle \boldsymbol{z}_{\ell ij} \rangle^\top \right) \\ &\cdot \left(\sum_{\ell=1}^{L} \sum_{i=1}^{n_\ell} \frac{R_{\ell ij}^{(t)}}{v_\ell^{(t+1)}} \langle \boldsymbol{z}_{\ell ij}\boldsymbol{z}_{\ell ij}^\top \rangle \right)^{-1}. \end{aligned}$$

These expressions naturally generalize those in [1]. For the subsequent results, we initialized the parameter estimates by using final MPPCA estimates. MPPCA was initialized using 1000 iterations of K-Planes [13] [14].

5. EXPERIMENTS & RESULTS

5.1. Synthetic Datasets

We generated 25 separate synthetic datasets. Each dataset contained $n = 10^3$ data samples of dimension $d = 10^2$ according to the model (3), where there were $L = 2$ noise groups and $k = 3$ factors for each of the $J = 3$ affine subspaces. The factor matrices were generated as $\boldsymbol{F}_j = \boldsymbol{U}_j \mathrm{Diag}^{1/2}(\boldsymbol{\lambda})$ for $j = 1, \ldots, J$, where $\boldsymbol{U}_j \in \mathrm{St}_{d,k}$ was drawn uniformly at random, and $\boldsymbol{\lambda} = (16, 9, 4)$ in all datasets. The elements of the mean vectors $\boldsymbol{\mu}_j$ were drawn independently and uniformly at random in the interval $[0, 1]$.

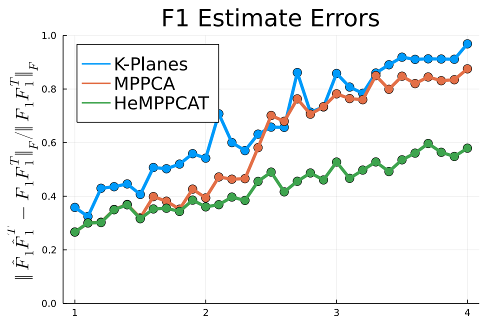
In all 25 datasets, the first $n_1 = 800$ samples had noise variance v_1 that we swept through from 1 to 4 in step sizes of 0.1. Mixtures 1 and 2 each generated 250 of these samples, while mixture 3 generated the remaining 300. The other $n_2 = 200$ samples had noise variance $v_2 = 1$, where mixtures 1 and 3 each generated 50 of these samples, and mixture 2 generated 100 of them.

We applied K-Planes, MPPCA, and HeMPPCAT to compute estimates $\hat{\boldsymbol{F}}_j$ of the underlying factors $\boldsymbol{F}_j$ across all 25 datasets. In every dataset, we recorded the normalized estimation errors $\|\hat{\boldsymbol{F}}_j \hat{\boldsymbol{F}}_j^\top - \boldsymbol{F}_j \boldsymbol{F}_j^\top\|_{\mathrm{F}} / \|\boldsymbol{F}_j \boldsymbol{F}_j^\top\|_{\mathrm{F}}$ of all methods at every value of v_1. For each v_1, we averaged the errors across all 25 datasets.

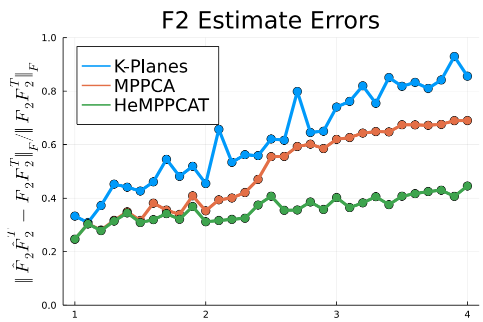
Figure 1 compares the average $\boldsymbol{F}_j$ estimation errors across the 25 datasets against v_1. When v_1 was close to v_2, the dataset noise was fairly homoscedastic and MPPCA and HeMPPCAT had similar estimation errors. As v_1 increased, the dataset noise became increasingly heteroscedastic, and HeMPPCAT had much lower errors than MPPCA. At almost all values of v_1, the non-statistical K-Planes method had higher errors than both MPPCA and HeMPPCAT.

5.2. Hopkins 155 Dataset

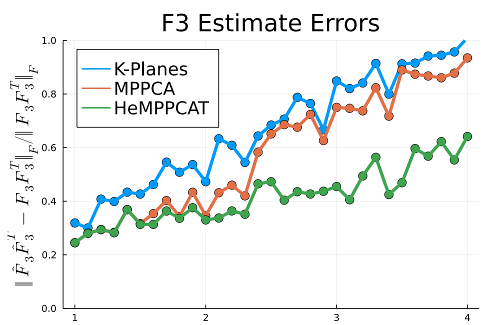
In computer vision, motion segmentation is the task of segmenting moving objects in a video sequence into several independent regions, each corresponding to a different object. The Hopkins 155 dataset is a series of 155 video sequences containing bodies in motion. The dataset has the coordinates of n feature points $\boldsymbol{x}_{f,i} \in \mathbb{R}^2$ tracked across F video frames. A feature point *trajectory* is formed by stacking the feature points across the frames: $\boldsymbol{t}_i = \begin{bmatrix} \boldsymbol{x}_{1,i}^\top & \ldots & \boldsymbol{x}_{F,i}^\top \end{bmatrix}^\top$. These trajectories are clustered to segment the motions using affine subspace clustering methods. A single body's trajectories lie in an affine subspace with dimension of at most 4 [15],



(a) $\boldsymbol{F}_1$ error vs. v_1



(b) $\boldsymbol{F}_2$ error vs. v_1



(c) $\boldsymbol{F}_3$ error vs. v_1

Fig. 1: Average $\boldsymbol{F}_j$ estimation error vs. v_1 (lower is better).

so the trajectories of J bodies lie in a union of J affine subspaces. The dataset includes the ground-truth cluster assignments for each trajectory.

Many motion segmentation methods assume the moving bodies are *rigid*. In practice, many common objects of interest do not satisfy rigid body motion. For instance, feature points on a walking human's legs move with different velocities than feature points on the torso. These differences can be modeled as heteroscedastic noise: trajectories of the leg feature points may lie in a different noise group than those of the torso.

Each video sequence contains either $J = 2$ or $J = 3$ moving bodies. To simulate nonrigid body motion, we added synthetic Gaussian noise to the trajectories. In each sequence, we created $L = 3$ synthetic noise groups with variances v_1, v_2, v_3 corresponding to signal to noise ratio (SNR) values of $-30, -25$, and -20 dB relative to that sequence's maximum trajectory ℓ_2 norm. Noise groups 1, 2, and 3 contained 50%, 35%, and 15% of all trajectories, respectively.

	K-Planes	MPPCA	HeMPPCAT
Noise group 1 (low noise)	24.1%	19.4%	**18.6%**
Noise group 2 (medium noise)	24.5%	27.3%	**19.3%**
Noise group 3 (high noise)	28.0%	34.8%	**20.1%**
Overall	24.8%	24.5%	**19.1%**

Table 1: Average misclassification rate on Hopkins 155 video dataset with synthetic heteroscedastic noise (lower is better).

For each sequence, we divided the dataset of trajectories into train and test sets using an 80/20 split. We applied K-Planes, MPPCA, and HeMPPCAT on the train set, and then used parameter estimates from all methods to classify trajectories in the test set. The test trajectories were classified based on nearest affine subspace using the K-Planes estimates, and by maximum likelihood using MPPCA and HeMPPCAT estimates, i.e., the predicted body for a test point $\boldsymbol{t}$ was $\operatorname{argmax}_j \hat{\pi}_j p(\boldsymbol{t} \mid j; \hat{\boldsymbol{\theta}})$, where $\hat{\pi}_j$ is the estimated mixing proportion of body j. We computed $p(\boldsymbol{t} \mid j; \hat{\boldsymbol{\theta}})$ according to the approaches' respective data models (1) and (3).

Table 1 shows the misclassification rate on the test set using the methods' parameter estimates. Using HeMPPCAT's estimates achieved lower classification error than using the other two approaches' estimates in each of the noise groups, and on the overall test set.

6. CONCLUSION

This paper generalized MPPCA to jointly estimate the underlying factors, means, and noise variances from data with heteroscedastic noise. The proposed EM algorithm sequentially updates the mixing proportions, noise variances, means, and factor estimates. Experimental results on synthetic and the Hopkins 155 datasets illustrate the benefit of accounting for heteroscedastic noise.

There are several possible interesting directions for future work. We could generalize this approach even further by accounting for other cases of heterogeneity, e.g., missing data or heteroscedastic noise across features. There may be faster convergence variants of EM such as a space-alternating generalized EM (SAGE) approach [16] that could be explored. Another direction could be jointly estimating the number of noise groups L and mixtures J along with the other parameters.

7. REFERENCES

[1] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol. 11, no. 2, pp. 443–82, Feb. 1999.

[2] Y. M. Lu and M. N. Do, "Sampling signals from a union of subspaces [A new perspective for the extension of this theory]," *IEEE Sig. Proc. Mag.*, vol. 25, no. 2, pp. 41–7, Mar. 2008.

[3] T. Blumensath, "Sampling and reconstructing signals from a union of linear subspaces," *IEEE Trans. Info. Theory*, vol. 57, no. 7, pp. 4660–71, July 2011.

[4] Y. C. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Trans. Info. Theory*, vol. 55, no. 11, pp. 5302–16, Nov. 2009.

[5] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE Trans. Patt. Anal. Mach. Int.*, vol. 27, no. 12, pp. 1945–59, Dec. 2005.

[6] R. Vidal, "Subspace clustering," *IEEE Sig. Proc. Mag.*, vol. 28, no. 2, pp. 52–68, Mar. 2011.

[7] Adnane Cabani, Karim Hammoudi, Halim Benhabiles, and Mahmoud Melkemi, "Maskedface-net–a dataset of correctly/incorrectly masked face images in the context of covid-19," *Smart Health*, vol. 19, pp. 100144, 2021.

[8] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE transactions on pattern analysis and machine intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[9] D. Hong, K. Gilman, L. Balzano, and J. A. Fessler, "HePPCAT: probabilistic PCA for data with heteroscedastic noise," *IEEE Trans. Sig. Proc.*, vol. 69, pp. 4819–34, Aug. 2021.

[10] A. Collas, F. Bouchard, A. Breloy, G. Ginolhac, C. Ren, and J-P. Ovarlez, "Probabilistic PCA from heteroscedastic signals: geometric framework and application to clustering," *IEEE Trans. Sig. Proc.*, vol. 69, pp. 6546–60, 2021.

[11] Alec S Xu, Laura Balzano, and Jeffrey A Fessler, "HeMPPCAT: mixtures of probabilistic principal component analysers for data with heteroscedastic noise," 2023, Preprint arXiv:2301.08852.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Stat. Soc. Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[13] Nandakishore Kambhatla and Todd K Leen, "Dimension reduction by local principal component analysis," *Neural computation*, vol. 9, no. 7, pp. 1493–1516, 1997.

[14] Paul S Bradley and Olvi L Mangasarian, "K-plane clustering," *Journal of Global optimization*, vol. 16, no. 1, pp. 23–32, 2000.

[15] Carlo Tomasi and Takeo Kanade, "Shape and motion from image streams under orthography: a factorization method," *International journal of computer vision*, vol. 9, no. 2, pp. 137–154, 1992.

[16] J. A. Fessler and A. O. Hero, "Space-alternating generalized expectation-maximization algorithm," *IEEE Trans. Sig. Proc.*, vol. 42, no. 10, pp. 2664–77, Oct. 1994.