Journal of Machine Learning Research 25 (2024) 1-42 Submitted 2/21; Published 3/24

Adaptive Latent Feature Sharing for Piecewise Linear Dimensionality Reduction

Adam Farooq A.FAROOQ6@ASTON.AC.UK
Department of Mathematics
Aston University
Birmingham, UK

Yordan P. Raykov ✉ YORDAN.RAYKOV@NOTTINGHAM.AC.UK
School of Mathematical Sciences
University of Nottingham
Nottingham, UK

Petar Raykov PETAR.RAYKOV@MRC-CBU.CAM.AC.UK
MRC Cognition and Brain Sciences Unit
University of Cambridge
Cambridge, UK

Max A. Little MAXL@MIT.EDU
Department of Computer Science
University of Birmingham
Birmingham, UK

Editor: Barbara Engelhardt

Abstract

Linear Gaussian exploratory tools such as principal component analysis (PCA) and factor analysis (FA) are widely used for exploratory analysis, pre-processing, data visualization, and related tasks. Because the linear-Gaussian assumption is restrictive, for very high dimensional problems, they have been replaced by robust, sparse extensions or more flexible discrete-continuous latent feature models. Discrete-continuous latent feature models specify a dictionary of features dependent on subsets of the data and then infer the likelihood that each data point shares any of these features. This is often achieved using *rich-get-richer* assumptions about the feature allocation process where the dictionary tries to couple the feature frequency with the portion of total variance that it explains. In this work, we propose an alternative approach that allows for better control over the feature to data point allocation. This new approach is based on two-parameter discrete distribution models which decouple feature sparsity and dictionary size, hence capturing both common and rare features in a parsimonious way. The new framework is used to derive a novel adaptive variant of factor analysis (aFA), as well as an adaptive probabilistic principal component analysis (aPPCA) capable of flexible structure discovery and dimensionality reduction in a wide variety of scenarios. We derive both standard Gibbs sampling, as well as efficient expectation-maximisation inference approximations converging orders of magnitude faster, to a reasonable point estimate solution. The utility of the proposed aPPCA and aFA models is demonstrated on standard tasks such as feature learning, data visualization, and data whitening. We show that aPPCA and aFA can extract interpretable, high-level features for raw MNIST or COLI-20 images, or when applied to the analysis of autoencoder

features. We also demonstrate that replacing common PCA pre-processing pipelines in the analysis of functional magnetic resonance imaging (fMRI) data with aPPCA, leads to more robust and better-localised blind source separation of neural activity.

Keywords: principal component analysis, factor analysis, dimensionality reduction, Bayesian nonparametrics

1. Introduction

Linear dimensionality reduction methods such as *factor analysis* (FA) and *principal component analysis* (PCA) are mainstays of high-dimensional data analysis, due to their simple geometric interpretation and attractive computational properties. Both FA and PCA can be seen as matrix decomposition techniques which aim to explain the dependence structure among high-dimensional observations through a decomposition of the positive-definite covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{D\times D}$, i.e., assuming $\mathbf{\Sigma} = \mathbf{\Lambda}\mathbf{\Lambda}^T + \boldsymbol{\epsilon}$ where $\mathbf{\Lambda} \in \mathbb{R}^{D\times K}$ is a factor loading matrix with $D > K$ and $\boldsymbol{\epsilon} \in \mathbb{R}^{D\times D}$ with entries depending on the assumed FA or PCA model. As data increases in size and complexity, the assumption that linear components are a linear combination of *all* of the original variables becomes increasingly restrictive. This has motivated a plethora of work on variants of *sparse* FA models (West, 2003; Knowles and Ghahramani, 2007; Paisley and Carin, 2009a; Bhattacharya and Dunson, 2011; Gao et al., 2013) which play a fundamental role in dimensionality reduction and latent structure discovery for high dimensional data. Sparse FA models tend to vary significantly depending on whether or not they are motivated by *overcomplete* setting with $N \ll D$, i.e., dimensionality larger than the number of observations (Aharon et al., 2006); a common example of which are gene expression studies (West, 2003; Carvalho et al., 2008). The reason for that is that, if $N \ll D$ for the data, and we assume the factor matrix is full rank, an infinite number of solutions are available for the representation problem, implying that explicit constraints on the solution must be set. Such constraints most often take the form of shrinkage assumptions about the diagonal and off-diagonal elements of the factor loading matrix. In the context of PCA models, Zou et al. (2006) place a least absolute shrinkage and selection operator (LASSO) regularisation on the factor loading vectors, which, compared to simple thresholding, leads to more interpretable factors or components. A fully Bayesian formulation can be derived using relevance determination priors (Engelhardt and Stephens, 2010; Campbell and Yau, 2017) such as a Student-t prior placed on the covariance of the factor loading matrix. West (2003) proposed placing a two-component mixture model over the loading vectors that enable factors to be switched on or off, imposing natural dimensionality reduction. In this scenario, the probability that factor loadings are non-zero is independent across all points. Bhattacharya and Dunson (2011) proposed a sparse Bayesian *infinite* factor model which assumes a *multiplicative gamma process* prior on the loading vectors. The sparse Bayesian infinite FA allows natural inference of the number of latent sparse factors but assumes coupling between the portion of explained variance and factor loading sparsity. This coupling is common to many sparse FA models and is a consequence of applying shrinkage through a single parameter such as the variance of the factor loadings on all loading parameters (Gao et al., 2013). These continuous sparsity-inducing priors all have the property that they impose strong shrinkage around zero but have sub-exponential tails, which allow signals to escape shrinkage. Gao et al. (2013) addressed this issue with

an alternative factor analysis setup where a flexible *three-parameter beta prior* is applied to the factor loading vectors to induce element-specific, factor-specific, and global shrinkage. Then, an additional two-component mixture prior is applied to cluster each factor as either sparse or dense. Durante (2017) provides an excellent introduction to more of the issues involved with multiplicative gamma process priors and recently, an additional flexible and sparse, nonparametric factor analysis prior was proposed in Legramanti et al. (2020).

Going beyond the overcomplete setting, FA models can be augmented with discrete latent variables which explicitly single out subsets of observations that share certain factor loadings. In this setup, the factor loadings might be a full rank matrix, but a set of latent variables augments the factors and only a subset of them take non-zero values. This formulation is achieved in *latent feature models* which augments the latent space and focuses on explicitly modelling the partitioning of the high dimensional data (Ghahramani et al., 2007). The same augmentation is applied in *latent class* FA models with the difference being the partition prior over the augmenting variables. Latent class FA models are designed for modelling non-overlapping clustering in a latent lower-dimensional space. Some well-known examples include variants of a mixture of FA models (Ghahramani et al., 1996; Ghahramani and Beal, 2000; Campbell and Yau, 2017) and variants of a mixture of PCA models (Parsons et al., 2004; Vidal, 2011). Latent feature models allow for richer clustering topology, but the challenge is designing a sufficiently flexible and intuitive model of the latent feature space. Several nonparametric FA models have addressed this using the *beta processes* (Paisley and Carin, 2009a), or their marginal *Indian buffet processes* (Knowles and Ghahramani, 2007; Rai and Daumé, 2009) (IBP) which have infinite capacity and can be used to infer random feature space dimensionality (i.e., number of features). However, with IBP FA models, feature allocation frequency is coupled with the portion of variance explained by the factors: as the portion of explained variance decreases, more loadings are switched off. This is a consequence of IBP regularizing signal and noise in the same way (Efron, 2008; Gao et al., 2013), and adds to the restrictive implications the IBP makes about the growth rate of the feature dictionary (Teh and Gorur, 2009). As a consequence, IBP FA models lead to non-interpretable, spurious features and overestimation of the underlying number of features (Elvira et al., 2017).

In this work, we propose a novel discrete-continuous latent factor model framework which can be used to capture both sparse and dense feature allocation distributions. The idea is to use separate parameters to control the total number of features in the feature dictionary and the number of features allocated to each data item: the ratio of these parameters then implicitly controls the sparsity of the feature allocation matrix. We show that truncated multinomial models or equivalent various forms of the *multivariate hypergeometric distribution* can be used as a feature allocation process. For simplicity, we mostly focus on the parametric version of the proposed models which fixes the number of unique instantiated features, but at the same time has a different parameter controlling the number of unique features used to represent each data point. This is critical since it allows us to naturally separate (1) features which explain a large proportion of total variance for a small subset of the data, and (2) spurious features which explain a small proportion of total variance for a potentially larger subset of the data. This formulation is natural in the context of data visualization and dimensionality reduction. In visualization, normally,

points are reduced to two or three dimensions; in dimensionality reduction, we model each point with $K \ll D$ dimensions.

2. Preliminaries

2.1 Linear Gaussian Latent Variable Models

Linear Gaussian latent variable models (LVMs) assume the following generative model for the data

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \boldsymbol{\mu} + \boldsymbol{\epsilon} \tag{1}$$

where the observed data, $\mathbf{y} \in \mathbb{R}^D$; $\mathbf{W} \in \mathbb{R}^{D \times K}$ is a transformation matrix, the columns of which are commonly referred to as *factor loadings* or *principal components* depending on the constraints placed on $\mathbf{W}$; $\mathbf{x} \in \mathbb{R}^K$ are unknown multivariate Gaussian latent variables, also referred to as factors or projections; $\boldsymbol{\mu} \in \mathbb{R}^D$ is a mean (offset) vector which is assumed to be zero, and $\boldsymbol{\epsilon}$ describes the model noise, typically Gaussian. Depending on the assumptions we impose on $\mathbf{x}$, $\mathbf{W}$, and $\boldsymbol{\epsilon}$, we can obtain various, widely-used techniques:

- The ubiquitous *principal component analysis* (PCA) (Pearson, 1901) can be derived from Equation (1), and further making the assumptions that $\boldsymbol{\mu} = 0$, the matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K]$ has orthonormal columns; i.e., $\mathbf{w}_i^T \mathbf{w}_j = 0$ if $i \neq j$, and the variance of the isotropic noise is 0, i.e., assume $\boldsymbol{\epsilon} \sim \mathcal{MVN}\left(\mathbf{0}, \sigma^2 \mathbf{I}_D\right)$ and $\sigma^2 \to 0$ (known as the small variance asymptotic (SVA) assumption).

- If we avoid the SVA of PCA, but still assume $\mathbf{W}$ has orthogonal vectors and Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{MVN}\left(\mathbf{0}, \sigma^2 \mathbf{I}_D\right)$, we recover *probabilistic PCA* (PPCA) (Tipping and Bishop, 1999).

- Omitting the orthogonality assumption on $\mathbf{W}$ and assuming more flexible elliptical noise $\boldsymbol{\epsilon}_n \sim \mathcal{MVN}\left(\mathbf{0}, \text{diag}\left(\boldsymbol{\sigma}^2\right)\right)$ with $\boldsymbol{\sigma}^2 = \left[\sigma_1^2, \ldots, \sigma_D^2\right]$, we obtain the classic *factor analysis* (FA) (Harman, 1960).

- Variants of *independent component analysis* (ICA) (Comon, 1994) can be obtained by assuming flexible elliptical noise $\epsilon \sim \mathcal{MVN}\left(\mathbf{0}, \text{diag}\left(\boldsymbol{\sigma}^2\right)\right)$ with $\boldsymbol{\sigma}^2 = \left[\sigma_1^2, \ldots, \sigma_D^2\right]$, but also, assuming a non-Gaussian distribution model for the latent variables $\mathbf{x} \in \mathbb{R}^K$; for example, the multivariate Laplace distribution (Knowles and Ghahramani, 2007).

2.2 Latent Feature Linear Gaussian Models

In latent feature linear Gaussian LVMs, we augment the model in Equation (1) with discrete indicators which aim to infer inherent groupings of points sharing columns of $\mathbf{W}$. We can write the following construction in matrix notation for N, D-dimensional observations

$$\mathbf{Y} = \mathbf{W}(\mathbf{X} \odot \mathbf{Z}) + \mathbf{E} \tag{2}$$

where $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$ is the observation matrix, $\mathbf{W}$ is a $(D \times K)$ transformation matrix, $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]$ is a binary indicator matrix selecting which of K hidden sources are active, $\odot$ denotes the *Hadamard product* (also known as the element-wise or *Schur product*),

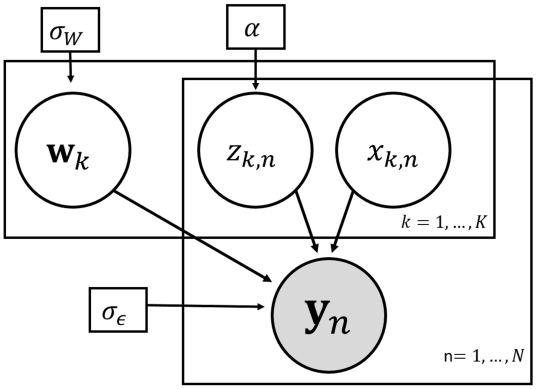


Figure 1: Graphical model for generic Bayesian latent feature models of which the proposed *adaptive factor analysis* (aFA) and the *adaptive probabilistic principal component analysis* (aPPCA) models are particular examples. If we assume $K \to \infty$ we recover Bayesian nonparametric models such as Bayesian nonparametric FA (isFA) and ICA models.

$\mathbf{E} = [\epsilon_1, \ldots, \epsilon_N]$ is a noise matrix consisting of N independent and identically distributed D-dimensional zero-mean vectors drawn from $\mathcal{MVN}\left(\mathbf{0}, \sigma^2 \mathbf{I}_D\right)$; finally $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ are the latent variables where each point $x_{k,n}$ is assumed Gaussian for FA and PCA models, and Laplace distributed for Bayesian ICA models. Linear Gaussian models (i.e., in the form of Equation 2) with orthogonal $\mathbf{W}$ are often referred to as *subspace models* motivated by the fact that columns of $\mathbf{W}$ span subspaces and $\mathbf{Z}$ indicates latent groupings of data which are efficiently represented by the same subspaces. In the special case where $\mathbf{Z}$ is made of one-hot-encoding vectors (i.e., latent class models), the model of Equation (2) can be referred to as a subspace clustering model where columns of $\mathbf{W}$ specify lower-dimensional subspaces locally preserving the variance of $\mathbf{Y}$. The graphical model for the generic latent feature model of Equation (2) is depicted in Figure 1. Figure 2 illustrates synthetic data generated from Equation (2) when making different assumptions about the feature allocation matrix $\mathbf{Z}$.

2.3 Beta-Bernoulli Latent Feature Models

Assume the model in Equation (2) with binary feature allocation vectors $\mathbf{z}_n$ which are independently drawn from a Bernoulli distribution with the K mixing parameters $\{p_k\}_{k=1,\ldots,K}$ (Knowles and Ghahramani, 2007). If the mixing parameters are placed are a conjugate beta distribution prior, as the number of latent features $K \to \infty$, one can show that the conjugate prior over the matrix $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]$ is the *beta process* (Hjort et al., 1990). The mixing parameters can be integrated out to work with the simpler IBP marginal process. Under the IBP prior, the indicator matrix $\mathbf{Z}$ has K rows and N columns; with K being the unknown number of represented features in the observed data which is assumed to increase with N. The expected number of features $\overline{K}$ follows a Poisson distribution with mean $\alpha \sum_{n=1}^{N} \frac{1}{N}$; for large N, $\overline{K} \approx \alpha \ln(N)$. The prior for the matrix $\mathbf{Z}$ under the IBP is

$$\mathrm{P}(\mathbf{Z}|\alpha) \propto \exp(-\alpha H_N)\, \alpha^K \left(\prod_{k=1}^{K} \frac{(m_k - 1)!(N - m_k)!}{N!} \right) \tag{3}$$

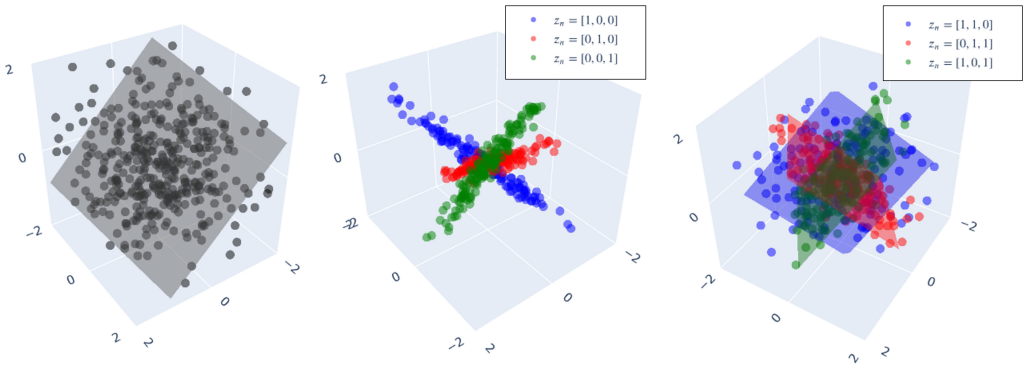


Figure 2: 3D synthetic data generated from latent feature linear Gaussian models; assuming (left) $\mathbf{Y} = \mathbf{WX} + \boldsymbol{E}$; (middle) $\mathbf{Y} = \mathbf{W}(\mathbf{X} \odot \mathbf{Z}) + \mathbf{E}$ with columns of $\mathbf{Z}$ being one-hot-encodings, equivalent to the *subspace clustering* assumption that data lies on one of three, linear 1D subspaces; and (right) $\mathbf{Y} = \mathbf{W}(\mathbf{X} \odot \mathbf{Z}) + \mathbf{E}$ with columns of $\mathbf{Z}$ being elements of the set: $\{[1, 1, 0], [0, 1, 1], [1, 0, 1]\}$. Shaded planes indicate the latent 2D subspace(s) spanned by different combinations of the PCs, i.e., columns of $\mathbf{W}$.

where $H_N = \sum_{n=1}^{N} \frac{1}{n}$ and $m_k = \sum_{n=1}^{N} z_{k,n}$. The IBP prior enforces sparsity on $\mathbf{Z}$ because it places diminishing probability on the event of having many common features k, i.e., features with large m_k. It has been observed that the number of data points being active in each feature follows *Zipf's law* (Teh and Gorur, 2009; Zipf, 1932); this implies that a small number of data points are active in all features; and a large number of data points are only active in a small number of features. This Zipf's law behaviour has been observed and proven as $N \to \infty$ (Teh and Gorur, 2009).

3. Latent Factor Analysis Models

In this section, we derive flexible latent feature FA models which leverage weakly constrained feature allocations and allow us to model a wide range of sparse FA models. We demonstrate that using uninformative discrete distributions or constrained distributions without replacement, one can derive simple modelling alternatives which have favourable properties such as the ability to represent both sparse and dense factor loadings.

In many machine learning applications such as data visualization, there are naturally occurring constraints for the intrinsic dimensionality of each point in the latent space. For example, we may wish to represent each point in exactly 2D, but not expect that the same two dimensions are shared across all data points; alternatively, we might wish to impose explicit constraints limiting the number of points sharing the same features.

3.1 Relevance Determination with Discrete Variables

Because latent class FA models make rigid partitioning assumptions, they fail to model richer clustering topologies, but latent feature FA models can infer uninterpretable features if not adequately constrained. For example, in beta-Bernoulli models, the first few fea-

tures are used to decompose most observations, and most of the remaining features only decompose a small portion of the data.

Let us revisit the common 'rich-get-richer' assumption underlying the feature allocation behaviour of many latent feature linear Gaussian models (see Section 2.2). Whereas latent feature FA models capture richer clustering topologies when compared to latent class FA models, beta-Bernoulli models impose strong assumptions about the distribution of the total number of factors and individual factor allocation frequency. Instead, a flexible relevance determination assumption can be that, each input data point is associated with a different subset of L features, selected from a total of K unique features. The parameter K then accounts for the global sharing of structure across overlapping groups of data points with common features; if K is large enough, each point can be associated with non-overlapping subsets of L features. This behaviour is equivalent to the mixture of the FAs model. As K decreases, more of these features are constrained to be shared across subsets of the data. L acts much like the number of latent dimensions in traditional linear LVMs, but here L is constrained by K. Thus, we can interpret L as the *local capacity* of the model, with K controlling the *global sharing capacity*. If $L = K$, we recover the classical linear Gaussian model of Equation (1), since all features are associated with all observed data points. As $K - L$ increases, more local structure in the data can be represented.

If we assume that, for each column of $\mathbf{Z}$ in a latent feature model, exactly L out of K features are non-zero, then there are $\binom{K}{L}$ possible configurations for each column of $\mathbf{Z}$. When a flat prior is placed on $\mathbf{Z}$, then each configuration has an equal likelihood, $\frac{1}{\binom{K}{L}}$. The joint probability over any allocation matrix is given by the product

$$\mathrm{P}\left(\mathbf{Z}|K, L\right) = \prod_{n=1}^{N} \frac{1}{\binom{K}{L}} \tag{4}$$

The restriction that L out of K features are non-zero, means that the columns of $\mathbf{Z}$ can no longer be distributed across the K feature rows. Instead, each of the $\frac{1}{\binom{K}{L}}$ configurations have categorical likelihood which depends on a different combination of L non-zero factor loadings

$$\mathrm{P}\left(\mathbf{z}_n = \mathbf{z}^* | \dots\right) \propto \prod_{j \in \mathbf{z}*} \mathrm{P}\left(\mathbf{y}_n | \mathbf{w}_j, \dots\right) \tag{5}$$

where $\mathbf{z}^*$ denotes some K-dimensional configuration $[1, 0, 0, 1, \dots, 0, 1]^T$ with L 1's. For large K and small L, the number of unique configurations $\mathbf{z}^*$ grows factorially and we may wish to approximate the posterior over $\mathbf{z}_n$ assuming conditional independence of the feature assignments. In that case, we sequentially sample without replacement from the categorical posterior

$$l|\mathbf{y}_n, \mathbf{W}, \mathbf{X} \sim \text{Categorical}\left(\frac{(1 - z_{1,n})\,\mathrm{P}\left(\mathbf{y}_n|\mathbf{w}_1, \dots\right)}{\sum_{k=1}^{K}(1 - z_{k,n})\,\mathrm{P}\left(\mathbf{y}_n|\mathbf{w}_k, \dots\right)}, \dots, \frac{(1 - z_{K,n})\,\mathrm{P}\left(\mathbf{y}_n|\mathbf{w}_K, \dots\right)}{\sum_{k=1}^{K}(1 - z_{k,n})\,\mathrm{P}\left(\mathbf{y}_n|\mathbf{w}_k, \dots\right)}\right). \tag{6}$$

This information is encoded in $\mathbf{z}_n$ by setting the l-th element to 1, this is repeated L times to ensure each $\mathbf{z}_n$ satisfies the constraint that L features are allocated per point.

The posterior in Equation (6) is identical to the posterior in a mixture model, but draws are made without replacement from the categorical posterior. If we are to further assume that the probability of each feature being active depends on how often it is selected in the rest of the data (i.e., the 'rich-get-richer' effect applies), we augment Equation (6) with independent counts

$$l|\mathbf{z}_n, \mathbf{y}_n, \mathbf{W} \sim \text{Categorical}\Big(\frac{(1-z_{1,n})\, m_1^{(-n)} \text{P}\left(\mathbf{y}_n|\mathbf{w}_1, \dots\right)}{\sum_{k=1}^{K}(1-z_{k,n})\, m_k^{(-n)} \text{P}\left(\mathbf{y}_n|\mathbf{w}_k, \dots\right)}, \dots \\ \dots, \frac{(1-z_{K,n})\, m_K^{(-n)} \text{P}\left(\mathbf{y}_n|\mathbf{w}_K, \dots\right)}{\sum_{k=1}^{K}(1-z_{k,n})\, m_k^{(-n)} \text{P}\left(\mathbf{y}_n|\mathbf{w}_k, \dots\right)}\Big) \tag{7}$$

where $m_k^{(-n)} = \sum_{i \neq n} z_{k,i}$. This results in a *multivariate hypergeometric* model for the numbers of active allocations in $\mathbf{Z}$.

3.2 Constrained Factor Allocation

Above, we assumed that the main feature allocation constraint is the number of non-zero factor loadings per points (i.e., a column-wise sparsity constraint on $\mathbf{Z}$ depending on L). However, we can also control the row-wise sparsity using constraints on the total number of times that a factor can be allocated to a point. Let us assume that each data point is associated with L out of K factors, and explicitly model the number of non-zero factor loadings across rows. We can place the *truncated multinomial distribution* as a prior on $\mathbf{Z}$ with K different categories and probability of success π_k for $k = 1, \dots, K$ and $\sum_{k=1}^{K} \pi_k = 1$. The truncated multinomial can be used to restrict the number of trials L, as well as the total number of times a category can be selected, i.e., c_k for each category k with $\sum_{k=1}^{K} c_k \geq L$. A sample from the truncated multinomial distribution is then a K-dimensional vector $\mathbf{m}$ of counts

$$\text{P}\left(\mathbf{m}|\boldsymbol{\pi}\right) = \frac{L!}{V\left(\boldsymbol{\pi}, L, \mathbf{c}\right)} \prod_{k=1}^{K} \frac{\pi_k^{m_k}}{m_k!} \tag{8}$$

where $\mathbf{m} = [m_1, \dots, m_K]^T$ and $V\left(\boldsymbol{\pi}, L, \mathbf{c}\right)$ is a normalising constant

$$V\left(\boldsymbol{\pi}, L, \mathbf{c}\right) = \sum_{l_1=0}^{c_1} \cdots \sum_{l_K=0}^{c_K} \left(\mathbb{I}\left(\sum_i l_i = L\right) L! \prod_{k=1}^{K} \frac{\pi_k^{l_k}}{l_k!} \right) \tag{9}$$

where $\mathbb{I}\left(\cdot\right)$ is an indicator function which is one if the statement inside is true, otherwise zero. Under this constrained model, we can write the conditional probability over the sparse matrix $\mathbf{Z}$ given $\pi = [\pi_1, \dots, \pi_K]$

$$\text{P}\left(\mathbf{Z}|\boldsymbol{\pi}\right) = \prod_{n=1}^{N} \prod_{k=1}^{K} \text{P}\left(z_{k,n}|\pi_k, \mathbf{c}^*\right) \propto \prod_{n=1}^{N} \frac{L!}{V\left(\boldsymbol{\pi}, L, \mathbf{c}\right)} \prod_{k=1}^{K} \frac{\pi_k^{z_{k,n}}}{z_{k,n}!} \tag{10}$$

where $\text{P}\left(z_{k,n}|\pi_k, \mathbf{c}^*\right)$ cannot be easily distributed since we need to keep track of $\mathbf{c}^*$ the total number of available draws from each factor, such that $\sum_{k=1}^{K} z_{k,n} = L$. In the fully Bayesian setting, one can model the allocation marginal probabilities $\pi_1, \dots, \pi_K$ with a

Dirichlet distribution parametrized by the counts $m_1, \ldots, m_K$. A nonparametric extension of this constrained model can be derived by taking the limit $K \to \infty$ in Equation (10) and integrating out π.

3.3 Adaptive FA Model and Inference

The complete Bayesian specification of the proposed adaptive FA framework takes the form

$$\begin{aligned} \mathbf{w}_k &\sim \mathcal{MVN}\left(\mathbf{0}, \sigma_w^2 \mathbf{I}_D\right), \ k \in \{1, \ldots, K\} \\ \boldsymbol{\epsilon}_n &\sim \mathcal{MVN}\left(\mathbf{0}, \sigma^2 \mathbf{I}_D\right) \\ \mathbf{x}_n &\sim \mathcal{MVN}\left(\mathbf{0}, \sigma_x^2 \mathbf{I}_K\right) \\ \mathbf{z}_n &\sim \mathcal{F}\left(\cdot\right) \\ \mathbf{y}_n &= \mathbf{W}\left(\mathbf{x}_n \odot \mathbf{z}_n\right) + \boldsymbol{\epsilon}_n \end{aligned} \tag{11}$$

where $\mathcal{F}(\cdot)$ denotes the chosen prior for $\mathbf{Z}$. For simplicity, we will use a single variance parameter σ^2 and assume the factor loading variables have fixed variance $\sigma_w^2 = 1$ (i.e., $\mathbf{w}_k \sim \mathcal{MVN}(\mathbf{0}, \mathbf{I}_D)$). Because of the prohibitive computational costs involved with training most existing latent feature FA models, we next derive a scalable expectation-maximisation (EM) algorithm for approximate inference in the aFA model. The data log-likelihood for the proposed model is

$$\begin{aligned} \mathcal{L}_N = -\sum_{n=1}^{N} \Bigg(& \frac{K}{2} \ln\left(\sigma_x^2\right) + \frac{D}{2} \ln\left(\sigma^2\right) \\ & + \frac{1}{2\sigma_x^2} \mathbf{x}_n^{\mathrm{T}} \mathbf{x}_n + \frac{1}{2\sigma^2} \mathbf{y}_n^{\mathrm{T}} \mathbf{y}_n - \frac{1}{\sigma^2} \mathbf{x}_n^{\mathrm{T}} \mathbf{A}_n^{\mathrm{T}} \mathbf{W}^{\mathrm{T}} \mathbf{y}_n + \frac{1}{2\sigma^2} \mathbf{x}_n^{\mathrm{T}} \mathbf{A}_n^{\mathrm{T}} \mathbf{W}^{\mathrm{T}} \mathbf{W} \mathbf{A}_n \mathbf{x}_n \Bigg) \end{aligned} \tag{12}$$

where $\mathbf{A}_n = \mathrm{diag}\left(\mathbf{z}_n\right)$ is a $(K \times K)$ matrix with diagonal elements set to $\mathbf{z}_n$. In the parametric aFA setting, we do not need to integrate out the factor parameters and mixing terms, therefore an efficient EM algorithm for training the aFA model can be derived. It can be used both for initialisation of a full Gibbs sampler or for rapidly obtaining a (local) maximum-a-posteriori solution for the model. For EM inference, we will treat the factor loading matrix $\mathbf{W}$ as a parameter instead of a random variable. In the E-step, the expectation of the latent variables $\mathbf{x}_{1,\ldots,N}$ is taken with respect to the posterior distribution; this results in the complete data log-likelihood

$$\begin{aligned} \mathcal{L}_N^{\text{complete}} = -\sum_{n=1}^{N} \Bigg(& \frac{K}{2} \ln\left(\sigma_x^2\right) + \frac{D}{2} \ln\left(\sigma^2\right) \\ & + \frac{1}{2\sigma_x^2} \mathrm{tr}\left(\mathbb{E}\left[\mathbf{x}_n \mathbf{x}_n^{\mathrm{T}}\right]\right) + \frac{1}{2\sigma^2} \mathbf{y}_n^{\mathrm{T}} \mathbf{y}_n - \frac{1}{\sigma^2} \mathbb{E}\left[\mathbf{x}_n\right]^{\mathrm{T}} \mathbf{A}_n^{\mathrm{T}} \mathbf{W}^{\mathrm{T}} \mathbf{y}_n \\ & + \frac{1}{2\sigma^2} \mathrm{tr}\left(\mathbf{A}_n^{\mathrm{T}} \mathbf{W}^{\mathrm{T}} \mathbf{W} \mathbf{A}_n \mathbb{E}\left[\mathbf{x}_n \mathbf{x}_n^{\mathrm{T}}\right]\right) \Bigg) \end{aligned} \tag{13}$$

Algorithm 1 EM algorithm for parametric adaptive factor (aFA) analysis.

Input: $\mathbf{Y}, \boldsymbol{\Theta}, \text{MaxIter}$
Initialise: Sample a random $(K \times N)$ binary matrix $\mathbf{Z}$ and initialise $\{\mathbf{W}, \mathbf{X}\}$ using PCA
for iter $\leftarrow 1$ to MaxIter
  for $n \leftarrow 1$ to N
    Set $\mathbf{x}_n = \left(\sigma_x^{-2}\mathbf{I}_K + \sigma^{-2}\mathbf{A}_n\mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{A}_n\right)^{-1}\left(\sigma^{-2}\mathbf{A}_n\mathbf{W}^{\mathrm{T}}\mathbf{y}_n\right)$
    Set $\boldsymbol{\Psi}_n = \left(\sigma_x^{-2}\mathbf{I}_K + \sigma^{-2}\mathbf{A}_n\mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{A}_n\right)^{-1} + \mathbf{x}_n\mathbf{x}_n^T$
  Update $\mathbf{z}_{1,\dots,N}$ using (6)
  Set $\mathbf{W} = \left(\sum_{n=1}^N \mathbf{y}_n\left(\mathbf{A}_n\mathbf{x}_n\right)^{\mathrm{T}}\right)\left(\sum_{n=1}^N \mathbf{A}_n\boldsymbol{\Psi}_n\mathbf{A}_n\right)^{-1}$
  Set $\sigma^2 = \frac{1}{ND}\sum_{n=1}^N\left(\mathbf{y}_n^{\mathrm{T}}\mathbf{y}_n - 2\mathbf{x}_n^T\mathbf{A}_n\mathbf{W}^{\mathrm{T}}\mathbf{y}_n + \text{trace}\left(\mathbf{A}_n\mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{A}_n\boldsymbol{\Psi}_n\right)\right)$
  Set $\sigma_x^2 = \frac{1}{NK}\sum_{n=1}^N \text{trace}\left(\boldsymbol{\Psi}_n\right)$

where

$$\mathbb{E}\left[\mathbf{x}_n\right] = \left(\sigma_x^{-2}\mathbf{I}_K + \sigma^{-2}\mathbf{A}_n^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{A}_n\right)^{-1}\left(\sigma^{-2}\mathbf{A}_n^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{y}_n\right)$$
$$\mathbb{E}\left[\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\right] = \left(\sigma_x^{-2}\mathbf{I}_K + \sigma^{-2}\mathbf{A}_n\mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{A}_n\right)^{-1} + \mathbb{E}\left[\mathbf{x}_n\right]\mathbb{E}\left[\mathbf{x}_n\right]^T$$

In the M-step, the complete data log-likelihood in Equation (13) is maximised with respect to the other parameters; this is done by solving $\frac{\partial\mathcal{L}_N^{\text{complete}}}{\partial\mathbf{W}} = 0$, $\frac{\partial\mathcal{L}_N^{\text{complete}}}{\partial\sigma_x} = 0$ and $\frac{\partial\mathcal{L}_N^{\text{complete}}}{\partial\sigma} = 0$, which results in the following parameter updates

$$\mathbf{W} = \left(\sum_{n=1}^N \mathbf{y}_n\left(\mathbf{A}_n\mathbf{x}_n\right)^{\mathrm{T}}\right)\left(\sum_{n=1}^N \mathbf{A}_n\mathbb{E}\left[\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\right]\mathbf{A}_n\right)^{-1}$$
$$\sigma^2 = \frac{1}{ND}\sum_{n=1}^N\left(\mathbf{y}_n^{\mathrm{T}}\mathbf{y}_n - 2\mathbf{x}_n^T\mathbf{A}_n\mathbf{W}^{\mathrm{T}}\mathbf{y}_n + \text{trace}\left(\mathbf{A}_n\mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{A}_n\mathbb{E}\left[\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\right]\right)\right)$$
$$\sigma_x^2 = \frac{1}{NK}\sum_{n=1}^N \text{trace}\left(\mathbb{E}\left[\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\right]\right)$$

Since we are often interested only in a point estimate for the indicator variables $\mathbf{Z}$, iterative optimisation via a coordinate descent procedure can lead to a robust, local maximum-a-posteriori (MAP) estimate i.e., $\mathbf{Z}^{\text{MAP}}$ (Wang and Dunson, 2011; Broderick et al., 2013; Raykov et al., 2016). Then, the M-step update for the point estimates of the latent variables $\mathbf{z}_n$ involves choosing the top L active factors from Equation (6) or most likely configuration from Equation (4). The probabilities $\mathrm{P}\left(\mathbf{y}_n|\mathbf{w}_k, \dots\right)$ inside Equation (6) are

$$\mathrm{P}\left(\mathbf{y}_n|\mathbf{w}_k, \dots\right) = \gamma\exp\left(\frac{\mu^2}{2\gamma^2}\right) \tag{14}$$

where $\mu = \frac{\mathbf{w}_k^T\boldsymbol{\epsilon}_n^{-k}}{\sigma^2 + \mathbf{w}_k^T\mathbf{w}_k}$, $\gamma = \frac{\sigma^2}{\sigma^2 + \mathbf{w}_k^T\mathbf{w}_k}$, and $\boldsymbol{\epsilon}_n^{-k}$ is the same as $\boldsymbol{\epsilon}_n$ from Equation (11) but with $z_{k,n} = 0$. The complete EM algorithm for the proposed aFA is summarised in Algorithm 1.

4. Latent Feature Subspace Models

Latent feature visualization methods have received little attention, despite the popularity of sparse principal component analysis techniques (Zou et al., 2006; Jolliffe et al., 2003).

This is most likely due to the complexity of specifying distributions over, and performing parameter inference with, orthogonal matrices. In this section, we extend the Bayesian nonparametric FA model of Knowles and Ghahramani (2007) to the PPCA setting in which the columns of the transformation matrix $\mathbf{W}$ are orthogonal. We argue that nonparametric PPCA is likely to suffer from the same limitations, as isFA, in the presence of dense PCs, and in order to address these limitations, introduce an efficient *adaptive probabilistic principal component analysis* (aPPCA) framework which also uses hypergeometric feature allocations. The proposed aPPCA method allows for explicit control over both the number of unique columns K in $\mathbf{W}$, as well as, the observation-specific number of active vectors, L.

Latent feature subspace models are a special case of latent feature linear Gaussian models (see Section 2.2) where the latent traits are assumed to share orthogonal one-dimensional subspaces, characterised via the projection vectors $\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_K$ forming $\mathbf{W}$. If two points $\mathbf{y}_i$ and $\mathbf{y}_j$ both have high probabilities $\mathrm{P}(z_{k,i})$ and $\mathrm{P}(z_{k,j})$, $\mathbf{y}_i$ and $\mathbf{y}_j$ share certain covariance structure in the direction of $\mathbf{w}_k$. Both nonparametric and adaptive PPCA models share the following construction

$$
\begin{aligned}
&\mathbf{W} \sim \mathcal{B}(\cdot) \\
&\boldsymbol{\epsilon}_n \sim \mathcal{MVN}\left(\mathbf{0}, \sigma^2 \mathbf{I}_D\right) \\
&\mathbf{x}_n \sim \mathcal{MVN}\left(\mathbf{0}, \mathbf{I}_K\right) \\
&\mathbf{z}_n \sim \mathcal{F}(\cdot) \\
&\mathbf{y}_n = \mathbf{W}\left(\mathbf{x}_n \odot \mathbf{z}_n\right) + \boldsymbol{\epsilon}
\end{aligned}
\tag{15}
$$

where $\mathcal{B}(\cdot)$ is the distribution over matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K]$ with orthonormal columns (i.e., *Bingham* or *matrix von Misses-Fisher* distributions); i.e., $\mathbf{w}_i^T \mathbf{w}_j = 0$ if $i \neq j$ and $\mathbf{w}_i^T \mathbf{w}_i = 1$, $\mathcal{F}(\cdot)$ is the same as for aFA models in Section 3.

4.1 Inference

Computing the posterior distribution of the latent variables $\{\mathbf{X}, \mathbf{Z}\}$ and the projection matrix $\mathbf{W}$ is analytically intractable and we have to resort to approximate inference. Unlike with aFA above, the posterior updates of the orthonormal matrix $\mathbf{W}$ are not available in closed form. Numerically optimising over $\mathbf{W}$ and simultaneously marginalising $\mathbf{X}$ leads to slow mixing and an EM scheme leads to poor local solutions for this model. As a solution, an efficient Markov Chain Monte Carlo (MCMC) scheme (Gelman et al., 2013) can be derived which iterates between explicit updates for $\mathbf{W}$, $\mathbf{z}_n$, $\mathbf{x}_n$ and the hyperparameters we wish to infer, i.e., σ^2 and α (updates for σ^2 and α are given in Appendix B). Sampling from directional posteriors is prohibitively slow, so we propose a MAP scheme for the updates on $\mathbf{W}$. We could use automated MCMC platforms such as STAN (Carpenter et al., 2017) for inference, adopting continuous relaxation of the discrete variables such as Concrete distributions (Maddison et al., 2016) or numerical solver extensions such as Nishimura et al. (2017). However, since the proposed model is tractable, closed-form inference in latent feature PPCA is more efficient.

The joint data likelihood of both of the latent feature subspace models we propose takes the form

$$\mathrm{P}\left(\mathbf{Y}, \mathbf{W}, \mathbf{X}, \mathbf{Z} | \sigma^2, \alpha\right) = \prod_{n=1}^{N} \left(\mathrm{P}\left(\mathbf{y}_n | \mathbf{W}, \mathbf{x}_n, \mathbf{z}_n, \sigma^2\right) \prod_{k=1}^{K} \mathrm{P}\left(x_{k,n}\right) \mathrm{P}\left(z_{k,n} | \alpha\right) \right) \times \mathrm{P}\left(\mathbf{W}\right) \tag{16}$$

We can check whether the MCMC sampler has converged using standard tests such as Raftery and Lewis (1992) directly on Equation (16). Comparing the Bayesian nonparametric sparse PPCA model and the aPPCA model, the only difference is in $\mathrm{P}\left(\mathbf{Z}\right)$. We will see that this affects the posterior updates of $\mathbf{Z}$, but the rest of the inference algorithm is otherwise identical for both models.

Posterior of $\mathbf{W}$: In order to comply with the orthogonality constraint on $\mathbf{W}$, i.e., $\mathbf{w}_i \perp \mathbf{w}_j \ \forall i \neq j$, we must ensure that the distribution $\mathcal{B}\left(\cdot\right)$ from Equation (15) has support on the Stiefel manifold (see Tagare, 2011 for a good introduction). Elvira et al. (2017) explored exactly this problem in the context of latent feature subspace modelling and proposed using a conjugate *Bingham* prior (Bingham, 1974) independently on the columns of $\mathbf{W}$ leading to an independent *von Mises-Fisher* posterior over each column, where re-scaling is required after each sample to maintain orthogonality. However, empirical trials suggest that this results in very poor mixing; see Appendix D. To overcome this issue, we propose joint sampling of the columns of $\mathbf{W}$. We place a uniform prior over the Stiefel manifold on the matrix $\mathbf{W}$ which allows us to work with a *matrix von Mises-Fisher* (Khatri and Mardia, 1977) posterior

$$\mathrm{P}\left(\mathbf{W} | \mathbf{Y}, \mathbf{X}, \mathbf{Z}, \sigma^2\right) = {}_0\mathrm{F}_1^{-1}\left(\emptyset, \frac{D}{2}, \mathbf{B}\mathbf{B}^T\right) \exp\left(\mathrm{tr}\left(\mathbf{B}\mathbf{W}\right)\right) \tag{17}$$

where $\mathbf{B} = \frac{1}{2\sigma^2}\left(\mathbf{X} \odot \mathbf{Z}\right)\mathbf{Y}^T$ and ${}_0\mathrm{F}_1^{-1}\left(\cdot\right)$ is a hypergeometric function (Herz, 1955). The normalisation term of the matrix von Mises-Fisher posterior is not available in closed form, hence it is common to sample from it using rejection sampling. Fallaize and Kypraios (2016) proposed a Metropolis-Hastings scheme to generate samples from the matrix posterior in Equation (17). The resulting posterior of $\mathbf{W}$ converges faster than independent column-wise von-Mises-Fisher posteriors, but can be further sped up by numerical optimisation methods. Here, we propose updating the matrix $\mathbf{W}$ by maximising the posterior in Equation (17) over the Stiefel manifold, i.e., enforcing orthogonality $\mathbf{w}_i \perp \mathbf{w}_j \ \forall i \neq j$. An efficient implementation can be achieved using the Pymanopt toolbox (Townsend et al., 2016), for optimisation over manifolds with different geometries; this step is outlined in Appendix C.

Posterior of $\mathbf{X}$: The posterior distribution over the latent variables $x_{k,n}$ for which $z_{k,n} = 1$, is sampled from a Gaussian

$$x_{k,n} | \cdots \sim \mathcal{N}\left(\frac{\mathbf{y}_n^T \mathbf{w}_k}{\sigma^2 + 1}, \frac{\sigma^2}{\sigma^2 + 1}\right) \tag{18}$$

where all variables upon which $x_{k,n}$ depends, have been omitted for notational convenience, and $\mathbf{w}_k$ is the k-th column of the matrix $\mathbf{W}$.

4.1.1 Bayesian nonparametric sparse PPCA model

In Bayesian nonparametric PPCA, we place an IBP prior over the indicator matrix $\mathbf{Z}$; this assumes that after a finite number N of observations, only a finite number K of 1-D subspaces are active. This results in the first K rows of $\mathbf{Z}$ having non-zero entries, the remaining rows being all zeros. By construction, K cannot exceed the dimension of the data D and this leads to truncation of the IBP such that K has a upper limit of $K^{\max}$ where $K \leq K^{\max}$, therefore in Bayesian nonparametric PPCA, $\mathbf{Z}$ is a $(K^{\max} \times N)$ binary matrix, with the sum of the first K rows being non-zero and the sum of the remaining $K^{\max} - K$ rows being zero. We sample the matrix $\mathbf{Z}$ in two stages: sampling "existing features" and "new features"; in both of these stages the latent variables $x_{k,n}$ are marginalised out. The posterior distribution over the existing features $z_{k,n}$ is Bernoulli distributed

$$
\begin{aligned}
z_{k,n} \ldots \sim & \\
& \text{Bernoulli}\left(\frac{\mathrm{P}\left(\mathbf{y}_n | z_{k,n}=1\right)\mathrm{P}\left(z_{k,n}=1|\mathbf{z}_{k,-n}\right)}{\mathrm{P}\left(\mathbf{y}_n | z_{k,n}=1\right)\mathrm{P}\left(z_{k,n}=1|\mathbf{z}_{k,-n}\right)+\mathrm{P}\left(\mathbf{y}_n | z_{k,n}=0\right)\mathrm{P}\left(z_{k,n}=0|\mathbf{z}_{k,-n}\right)}\right) \\
= & \text{Bernoulli}\left(\frac{\frac{m_{k,-n}}{N}\exp\left(\frac{1}{2\sigma^2(\sigma^2+1)}\left(\mathbf{y}_n^T\mathbf{w}_k\right)\right)\left(\frac{\sigma^2}{\sigma^2+1}\right)^{\frac{1}{2}}}{\frac{m_{k,-n}}{N}\exp\left(\frac{1}{2\sigma^2(\sigma^2+1)}\left(\mathbf{y}_n^T\mathbf{w}_k\right)\right)\left(\frac{\sigma^2}{\sigma^2+1}\right)^{\frac{1}{2}}+1}\right)
\end{aligned}
\tag{19}
$$

where all variables upon which $z_{k,n}$ depends have been omitted for notational convenience, and $m_{k,-n} = \sum_{i \neq n} z_{k,i}$.

Then, we sample κ number of new features using $\kappa \sim \text{Poisson}\left(\frac{\alpha}{N}\right)$, where we maintain $\kappa > 0$ or $\kappa + K \leq K^{\max}$. For observed data point n, the posterior distribution over the new features is

$$
z_{K+j,n} | \cdots \sim \text{Bernoulli}\left(\frac{\exp\left(\frac{1}{2\sigma^2(\sigma^2+1)}\sum_{k=K+1}^{K+\kappa}\left(\mathbf{y}_n^T\mathbf{w}_k\right)^2\right)\left(\frac{\sigma^2}{\sigma^2+1}\right)^{\frac{\kappa}{2}}}{\exp\left(\frac{1}{2\sigma^2(\sigma^2+1)}\sum_{k=K+1}^{K+\kappa}\left(\mathbf{y}_n^T\mathbf{w}_k\right)^2\right)\left(\frac{\sigma^2}{\sigma^2+1}\right)^{\frac{\kappa}{2}}+1}\right) \tag{20}
$$

for $j = 1, \ldots, \kappa$ new features; all variables upon which $z_{K+j,n}$ depends have been omitted for notational convenience.

4.1.2 Adaptive PPCA model

In many applications of PPCA, constraints on the latent feature dimensionality occur naturally. In particular, for data visualization, we are mostly interested in reducing high dimensional data down to two or three dimensions; in regression problems when PCA is used to remove *multicollinearity* from the input features, the output dimensionality is usually fixed to D (the dimensionality of the input). In these scenarios, the constraints on $\mathbf{Z}$ in Section 3 allow for explicit control over the number of latent subspaces L used to decompose each single observed data point. In the aPPCA context, K denotes the number of unique orthogonal linear subspaces which we will use to reduce the original data into the lower dimensional space; each input data point can be associated with a different subset

Algorithm 2 Inference in Bayesian nonparametric PPCA using Gibbs sampling.

Input: $\mathbf{Y}, \mathbf{\Theta}, \text{MaxIter}, K$
Initialise: Sample a random $(K^{max} \times N)$ binary matrix $\mathbf{Z}$ and initialise $\mathbf{W}$ using PCA
for iter $\leftarrow 1$ to MaxIter
  for $n \leftarrow 1$ to N
    for $k \leftarrow 1$ to K
      Sample $z_{k,n}$ using Equation (19)
    Sample $\kappa \sim \text{Poisson}\left(\frac{\alpha}{N}\right)$
    Accept κ new features from Equation (20) and update K accordingly
  for $n \leftarrow 1$ to N
    for $k \leftarrow 1$ to K
      if $z_{k,n} = 1$
        Sample $x_{k,n}$ using Equation (18)
  Sample $\mathbf{W}$ using Equation (17)
  Sample $\left\{\sigma^2, \alpha\right\}$ from Appendix B

Algorithm 3 Inference in parametric aPPCA using Gibbs sampling.

Input: $\mathbf{Y}, \mathbf{\Theta}, \text{MaxIter}$
Initialise: Sample a random $(K \times N)$ binary matrix $\mathbf{Z}$ and initialise $\mathbf{W}$ using PCA
for iter $\leftarrow 1$ to MaxIter
  for $n \leftarrow 1$ to N
    for $k \leftarrow 1$ to K
      if $z_{k,n} = 1$
        Sample $x_{k,n}$ using (18)
  Update $\mathbf{z}_{1,\ldots,N}$ using (6)
  Sample $\mathbf{W}$ using (17)
  Sample $\left\{\sigma^2, \alpha\right\}$ using Appendix B

of L subspaces, selected from a total of K subspaces. Therefore, any single observed data point is represented by lower dimensional spaces subsets of $\mathbb{R}^L$. Note that the orthogonality assumption $\mathbf{w}_i \perp \mathbf{w}_j \; \forall i \neq j$ for the columns of $\mathbf{W}$ implies that $K \leq D$.

If we assume an uninformative allocation prior over $\mathbf{Z}$, the updates are parallel across N, since the number of observed data points assigned to each latent subspace no longer affects its assignment probability. In dimensionality reduction applications we often assume L being two or three, so that l_1 might indicate the x-axis, l_2 the y-axis and l_3 the z-axis of the lower dimensional subspace. A Gibbs sampler for aPPCA is suggested in Algorithm 3.

4.2 Relationship to PCA

If we marginalize the likelihood (from Equation 16) with respect to the discrete and continuous latent variables $\{\mathbf{x}_n, \mathbf{z}_n\}$) and take the SVA limit $\sigma^2 \to 0$, the maximum likelihood solution with respect to the transformation matrix $\mathbf{W}$ is a scaled version of the K largest eigenvectors of the covariance matrix of the data (up to an orthonormal rotation); proof of this is given in Appendix A. Furthermore, different priors over the matrix $\mathbf{Z}$ result in different variants of the model, giving explicit control over the scale of the different projection axis.

		Sparse				Balance				Dense				Class				Full			
$\frac{K}{2}$	aFA	0.15	0.31	**0.93**	4.20	0.35	0.42	**1.10**	**4.10**	0.77	0.93	1.36	**4.20**	0.08	**0.21**	**1.01**	4.20	0.86	1.15	1.67	4.10
	fsFA	0.12	0.34	1.11	4.21	0.25	0.41	1.20	4.20	0.37	0.63	1.26	4.30	0.02	0.25	1.12	**4.17**	0.43	0.72	1.42	4.45
	FA	0.96	1.25	1.63	**4.06**	1.04	1.12	1.79	4.49	2.33	2.64	3.44	5.88	1.08	1.79	2.69	4.75	3.24	2.77	3.73	6.51
	sdFA	0.07	0.35	1.42	4.20	0.10	0.39	1.52	4.40	0.12	0.40	1.12	5.02	0.09	0.25	1.02	4.50	0.28	0.36	1.21	4.60
	isFA	**0.03**	**0.26**	1.36	4.21	**0.04**	**0.35**	1.42	4.30	**0.04**	**0.29**	**1.01**	5.10	**0.02**	0.25	0.97	4.48	**0.06**	**0.29**	**1.07**	**4.03**
K	aFA	**0.01**	0.19	0.79	4.10	0.05	**0.18**	0.85	4.10	0.06	**0.18**	0.92	4.00	**0.01**	**0.15**	**0.78**	4.10	0.09	0.18	0.75	4.20
	fsFA	0.03	0.20	0.80	4.18	0.05	0.38	0.87	4.10	0.07	0.25	0.80	4.00	0.01	0.20	0.82	4.00	0.08	0.26	0.74	4.00
	FA	0.01	**0.18**	**0.77**	**3.42**	**0.01**	0.19	**0.75**	**3.28**	**0.01**	0.19	**0.74**	**3.12**	0.01	0.19	0.79	**3.06**	**0.01**	**0.18**	**0.72**	**3.09**
	sdFA	0.07	0.35	1.42	4.20	0.10	0.39	1.52	4.40	0.12	0.40	1.12	5.02	0.09	0.25	1.02	4.50	0.28	0.36	1.21	4.60
	isFA	0.03	0.26	1.36	4.21	0.04	0.35	1.42	4.30	0.04	0.29	1.01	5.10	0.02	0.25	0.97	4.48	0.06	0.29	1.07	4.03
$K+4$	aFA	**0.01**	**0.18**	0.72	**2.97**	0.01	0.18	**0.71**	**2.80**	0.01	1.60	0.70	2.85	**0.01**	**0.17**	**0.68**	2.86	0.01	0.18	0.71	2.87
	fsFA	0.03	0.25	1.00	4.08	0.03	0.27	0.99	3.99	0.04	0.29	0.96	4.00	0.02	0.24	0.96	3.99	0.05	0.29	1.06	4.01
	FA	0.01	0.18	**0.71**	3.00	**0.01**	**0.17**	0.72	3.02	**0.01**	**0.17**	**0.69**	**2.89**	0.01	0.17	0.70	**2.85**	**0.01**	**0.17**	**0.68**	**2.75**
	sdFA	0.07	0.35	1.42	4.20	0.10	0.39	1.52	4.40	0.12	0.40	1.12	5.02	0.09	0.25	1.02	4.50	0.28	0.36	1.21	4.60
	isFA	0.03	0.26	1.36	4.21	0.04	0.35	1.42	4.30	0.04	0.29	1.01	5.10	0.02	0.25	0.97	4.48	0.06	0.29	1.07	4.03
$2K$	aFA	**0.01**	**0.13**	**0.51**	**2.07**	**0.00**	**0.13**	**0.49**	**2.06**	**0.00**	**0.12**	**0.48**	**2.05**	**0.00**	**0.14**	**0.48**	**1.99**	**0.00**	**0.13**	**0.50**	**2.00**
	fsFA	0.03	0.25	0.98	4.06	0.03	0.26	0.96	3.90	0.04	0.28	0.96	3.88	0.02	0.24	0.95	3.94	0.05	0.28	1.03	3.96
	FA	0.01	0.16	0.66	2.40	0.01	0.15	0.61	2.40	0.01	0.15	0.60	2.38	0.01	0.17	0.63	2.25	0.01	0.16	0.61	2.30
	sdFA	0.07	0.35	1.42	4.2	0.10	0.39	1.52	4.40	0.12	0.40	1.12	5.02	0.09	0.25	1.02	4.50	0.28	0.36	1.21	4.60
	isFA	0.03	0.26	1.36	4.21	0.04	0.35	1.42	4.30	0.04	0.29	1.01	5.10	0.02	0.25	0.97	4.48	0.06	0.29	1.07	4.03
$3K$	aFA	**0.00**	**0.01**	**0.06**	**0.25**	**0.00**	**0.01**	**0.06**	**0.25**	**0.00**	**0.01**	**0.06**	**0.24**	**0.00**	**0.02**	**0.06**	**0.23**	**0.00**	**0.02**	**0.08**	**0.24**
	fsFA	0.03	0.25	0.97	4.01	0.00	0.25	0.94	4.00	0.04	0.28	0.94	3.95	0.02	0.24	0.92	3.89	0.05	0.28	1.00	3.89
	FA	0.01	0.12	0.47	1.07	0.01	0.12	0.47	1.15	0.01	0.13	0.47	1.11	0.01	0.13	0.51	1.11	0.01	0.12	0.47	1.09
	sdFA	0.07	0.35	1.42	4.2	0.10	0.39	1.52	4.40	0.12	0.40	1.12	5.02	0.09	0.25	1.02	4.50	0.28	0.36	1.21	4.60
	isFA	0.03	0.26	1.36	4.21	0.04	0.35	1.42	4.30	0.04	0.29	1.01	5.10	0.02	0.25	0.97	4.48	0.06	0.29	1.07	4.03

Table 1: Mean squared reconstruction error of different variants of factor analysis (FA) methods on five different data sets of dimension $D = 35$ and $K = 10$ latent features; the data generating process is described in Section 5.1. We have evaluated all methods and configurations using 4 different values of the noise parameter σ^2 controlling the signal-to-noise ratio in the generated data: $\sigma^2 = \{0.01, 0.25, 1, 4\}$ displayed left-to-right in each of the 5 main columns. Parameters for each model were estimated using 80% of the data set, and the resulting model was tested on the remaining 20%; average of the mean square out-of-sample reconstruction error over 20 different experiments, is reported. Standard deviation estimates have been omitted due to all estimates being < 0.005.

5. Experiments

This section provides some empirical results on the performance of the proposed variants of PCA and FA applied to data visualization, data whitening and blind source separation. The methods are evaluated on different kinds of synthetic data, images of handwritten digits from MNIST, images of objects from the Coil-20 data set, and functional magnetic resonance imaging (fMRI) data.

		Sparse				Balance				Dense				Class				Full			
$\frac{K}{2}$	aFA	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
	fsFA	**0.99**	**0.65**	0.64	0.70	**0.99**	0.85	0.79	0.81	**0.92**	0.87	0.87	**0.93**	0.67	**0.88**	**0.84**	**0.87**	0.85	0.89	0.95	**0.99**
	FA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	sdFA	0.63	0.63	**0.84**	0.65	0.77	0.53	0.78	0.62	0.70	0.65	0.61	0.52	0.55	0.44	0.50	0.49	0.41	0.32	0.45	0.47
	isFA	0.91	0.61	0.59	**0.75**	0.94	**0.87**	**0.92**	**0.88**	0.81	**0.98**	**0.90**	0.92	**0.88**	0.78	0.73	0.73	**0.98**	**0.89**	**0.98**	0.94
K	aFA	0.80	**0.80**	0.80	**0.80**	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	**0.80**	0.80	**0.80**	0.80	0.80	0.80	0.80
	fsFA	**0.99**	0.34	0.48	0.73	**0.96**	**0.96**	**0.99**	0.76	0.95	0.97	**0.98**	0.86	0.89	0.79	**0.85**	0.67	**0.99**	**0.95**	0.95	0.90
	FA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	sdFA	0.63	0.63	**0.84**	0.65	0.77	0.53	0.78	0.62	0.70	0.65	0.61	0.52	0.55	0.44	0.50	0.49	0.41	0.32	0.45	0.47
	isFA	0.91	0.61	0.59	0.75	0.94	0.87	0.92	**0.88**	**0.99**	**0.98**	0.90	**0.92**	**0.88**	0.78	0.73	0.73	0.98	0.89	**0.98**	**0.94**
$K+4$	aFA	0.71	**0.71**	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71
	fsFA	0.84	0.24	0.33	0.65	0.91	0.77	0.78	0.70	**0.98**	0.85	0.81	0.64	0.84	0.78	0.67	0.52	0.84	0.79	0.86	0.88
	FA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	sdFA	0.63	0.63	**0.84**	0.65	0.77	0.53	0.78	0.62	0.70	0.65	0.61	0.52	0.55	0.44	0.50	0.49	0.41	0.32	0.45	0.47
	isFA	**0.91**	0.61	0.59	**0.75**	**0.94**	**0.87**	**0.92**	**0.88**	0.81	**0.98**	**0.90**	**0.92**	**0.88**	**0.78**	**0.73**	**0.73**	**0.98**	**0.89**	**0.98**	**0.94**
$2K$	aFA	0.80	**0.80**	0.80	**0.80**	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	**0.80**	0.80	**0.80**	0.80	0.80	0.80	0.80
	fsFA	0.76	0.45	0.26	0.66	0.94	0.81	0.58	0.53	0.92	0.69	0.60	0.66	0.67	0.70	0.60	0.46	0.70	**0.91**	0.78	0.64
	FA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	sdFA	0.63	0.63	**0.84**	0.65	0.77	0.53	0.78	0.62	0.70	0.65	0.61	0.52	0.55	0.44	0.50	0.49	0.41	0.32	0.45	0.47
	isFA	**0.91**	0.61	0.59	0.75	**0.94**	**0.87**	**0.92**	**0.88**	**0.99**	**0.98**	**0.90**	**0.92**	**0.88**	0.78	**0.73**	0.73	**0.98**	0.89	**0.98**	**0.94**
$3K$	aFA	0.87	**0.87**	**0.87**	**0.87**	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	**0.87**	**0.87**	**0.87**	0.87	0.87	0.87	0.87
	fsFA	0.81	0.36	0.37	0.48	0.85	0.52	0.58	0.47	0.98	0.59	0.49	0.63	0.73	0.55	0.67	0.54	0.57	0.76	0.63	0.65
	FA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	sdFA	0.63	0.63	0.84	0.65	0.77	0.53	0.78	0.62	0.70	0.65	0.61	0.52	0.55	0.44	0.50	0.49	0.41	0.32	0.45	0.47
	isFA	**0.91**	0.61	0.59	0.75	**0.94**	**0.87**	**0.92**	**0.88**	**0.99**	**0.98**	**0.90**	**0.92**	**0.88**	0.78	0.73	0.73	**0.98**	**0.89**	**0.98**	**0.94**

Table 2: Sparsity of the variants of factor analysis (FA) methods evaluated in Table 1; the data generating process is described in Section 5.1. We have evaluated all methods and configurations using 4 different values of the noise parameter σ^2 controlling the signal-to-noise ratio in the generated data: $\sigma^2 = \{0.01, 0.25, 1, 4\}$ displayed left-to-right in each of the 5 main columns. For aFA sparsity is computed as the portion of zeros out of all values of the Z matrix, whereas for all other methods the reported sparsity is the portion of zeros out of the W matrix; conventional FA does not induce sparsity (i.e. the reported sparsity value would be 0.00 in all reported scenarios).

5.1 Synthetic Data from Latent Feature FA Models

First, we generate a wide variety of latent feature linear Gaussian data sets, assuming that the data matrix $\mathbf{Y} \in \mathbb{R}^{D \times N}$ takes the form $\mathbf{Y} = \mathbf{W}(\mathbf{X} \odot \mathbf{Z}) + \mathbf{E}$ with $\mathbf{X} \in \mathbb{R}^{K^* \times N}$ a latent feature matrix with standard Gaussian distribution; $\mathbf{W} \in \mathbb{R}^{D \times K^*}$ is a factor loading matrix with columns drawn from a multivariate Gaussian with mean zero and covariance matrix $\sigma_W^2 \mathbf{I}_{K^*}$ with $\sigma_W^2 = 1$; $\mathbf{E} \in \mathbb{R}^{D \times N}$ is a noise matrix with multivariate Gaussian columns each with mean zero and covariance matrix $\sigma^2 \mathbf{I}_D$ with varying signal-to-noise ratio across experiments: $\sigma^2 = \{0.01, 0.25, 1, 4\}$. The core of the generative model remains the same across the different data sets we generate ($N = 1200$, $D = 35$) with only the number of assumed latent features K the indicator matrix $\mathbf{Z}$ changing. We have considered five separate synthetic data sets and the distribution of $\mathbf{Z}$ for each setup is displayed in Figure 3. We evaluate the reconstruction error of the estimated MAP solutions of the different FA methods (i.e., with changing treatment of $\mathbf{Z}$) across each scenario; parameters for each model were estimated using 80% of the data set and the model was tested on the remaining 20%. The five FA methods tested are:

- Factor analysis (FA): the $\mathbf{Z}$ matrix is full of ones and all factors are shared across all points;
- Infinite sparse FA (isFA): the $\mathbf{Z}$ matrix is modelled with an IBP prior (from Equation 3) and most factors are shared only across small, overlapping subsets of points;
- Finite sparse FA (fsFA): the $\mathbf{Z}$ matrix is modelled with a finite Beta-Bernoulli distribution across all points and features;
- Adaptive FA (aFA): columns of $\mathbf{Z}$ are modelled with the prior in Equation (4), and aFA is trained with the EM scheme in Algorithm 1;
- Sparse and dense FA (sdFA): the factor loading matrix $\mathbf{W}$ is split into two components: one component is assumed to be dense and the other sparse Gao et al. (2013).

In this comparison, conventional FA implicitly aims to minimize the reconstruction error (at the SVA limit) whereas all of the other FA models introduce regularization constraints which aim to induce more parsimonious factors $\mathbf{W}$ which preserve the variance of only portions of the input. The regularization implies that the sparse FA models aim to preserve only a portion of the sample covariance, systematically leading to higher reconstruction error (in $N \gg D$ setting). Therefore, our emphasis is to evaluate the ability of aFA and different FA techniques to obtain comparable reconstruction error to the conventional FA (Table 1), but inducing sparser factor loadings $\mathbf{W}$ (Table 2). The different FA models are included as a reference point for their relative performance across different sparsity scenarios and signal-to-noise ratio, but the direct comparison across methods is difficult due to their different hyperparameters and intended use. The number of factors are fixed for the FA and fsFA to the generating number of factors K^*. The concentration parameter for the isFA is set such that the most common estimated number of factors is equal to K^*, and sdFA parameters are set following standard Bayesian model selection as described in Gao et al. (2013).

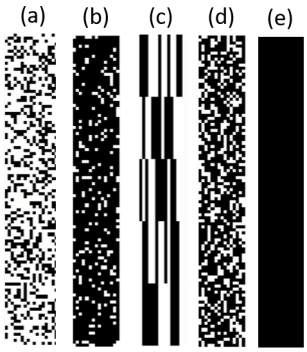


Figure 3: A plot of the different distributions used to model the latent space in Table 1 and Table 2. The subplots display different samples of the zero-one indicator matrix $\mathbf{Z}$: black cells indicate 1's and white cells indicate 0's. Five different latent models are considered: (a) Sparse latent feature model, (b) Dense latent feature model, (c) Latent class model in which sharing of some feature between subsets of points implies sharing of all features of those points, (d) Balanced latent feature model sampled from a specific hypergeometric distribution, and (e) Collapsed latent space consisting of a single state.

The results in Table 1 and Table 2 summarize the key results of the comparison when the generating $K^* = 10$. Further results are available in Appendix E (Table 4 and Table 5) which list results in the case of $K^* = 15$. The reported reconstruction error for sdFA and isFA does not change in the different rows (i.e. for values of the set K) since they are inferred with a fixed hyperparameters across rows and varying hyperparameters across noise levels; the fixed value of K used for inference affects the reported results for the parametric FA, fsFA and aFA. The second parameter L controlling the sparsity of $\mathbf{Z}$ is fixed deterministically to $L = K - 2$ (whereas K is fixed to values indicated in the left column of Tables 1 and 2) across experiments to reduce the degrees of freedom for aFA and ease the interpretability of Tables 1, 2, 4, and 5[1].

The results in Table 1 suggest that given large enough values for K and L, aFA achieves lowest mean reconstruction error across different noise levels. For fixed values of K and L, sdFA and aFA perform robustly across different sparsity data sets. As seen in Table 5, the sparsity of $\mathbf{Z}$ in aFA is deterministically determined by the choices of K and L. For high enough values of K and L with respect to the generating parameters, we get consistently low reconstruction error when the signal-to-noise ratio is high (i.e. σ^2 is low). As expected, increasing the noise levels leads to lower performance for all methods with sparser configurations of the different FA behaving more robustly at the higher noise levels. The aFA uses more more parameters (i.e. inferrable columns of $\mathbf{W}$) as K and L increase so intuitively we also see robust performance of aFA with respect to signal-to-noise ratio in those settings (for larger assumed values of K). In practice, alternating the ratio of K and L (i.e. either by using different fixed values or in a Bayesian fashion) would enable even better consistency across noise and sparsity levels. The fsFA performs similarly to the

1. Choosing K and L differently across the different synthetic data sets naturally renders lower reconstruction errors

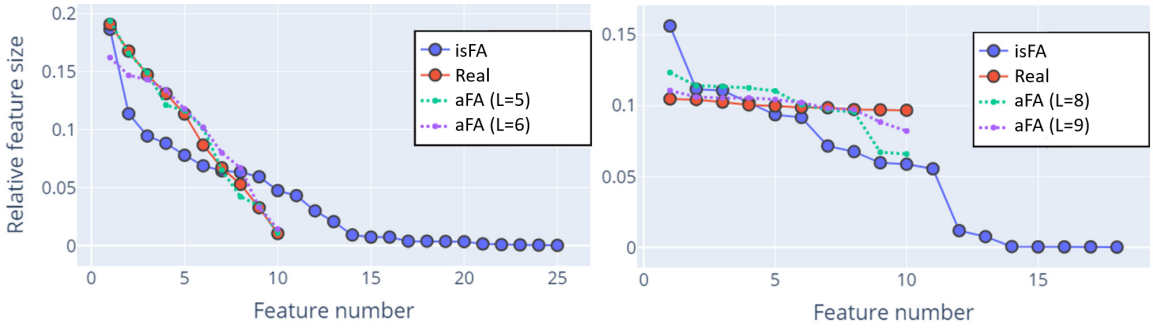


Figure 4: Estimated proportion of data associated with the different factors for sparse (left) and dense (right) synthetically generated linear Gaussian data (as in Tables 1). The y-axis denotes the proportion of points associated with a factor (i.e., factor allocation frequency); the x-axis denotes the factor numbers where factors are ordered by size (i.e., number of data points assigned to them). The true feature allocation frequency is displayed in red; the remaining lines show the feature allocation frequency associated with the estimated factors using the nonparametric factor analysis (isFA) and the proposed adaptive factor analysis (aFA) model.

conventional FA with slightly higher reconstruction error due to the sparsity regularization. We do observe that sdFA infers denser $\mathbf{W}$ when compared to the other sparse FA methods (namely fsDA and isFA) across data sets which are generated from more balance or dense models. The difference in the sparsity assumption in isFA and aFA is in the assumed priors on the matrix $\mathbf{Z}$. To provide some intuition of the induced power law behaviour under the IBP prior, in Figure 4 we display the distribution of the raw-totals $m_k = \sum_{n=1}^{N} z_{k,n}$. Those are shown in the case when the observed generating linear Gaussian model is sparse (left panel) with linearly decreasing raw-totals m_k (left-ordered) and dense (right panel) when the raw-totals m_k are uniformly distributed.

5.2 Factor Sharing between MNIST Digits

In this section we demonstrate estimated the parameters of the proposed aFA model on $N = 2500$ odd-labelled digits (500 of each type) from the MNIST handwritten digit data set. The raw pixel data were first reduced to $D = 350$ using standard PCA since this still preserves 99.5% of the total variance within the data. The total number of unique factors is set to $K = 100$ and the number L of observation-specific factors is set to maximize the factor profiles of the different digits.

In Figure 5, we show the factor sharing across the digits which are calculated based on the proportion of factors shared between different digit pairs. To estimate factor sharing, we have assumed greedily that indicator variables take the value 1 for the first L highest probability factors and 0 for the remaining factors. We count the number of factors shared between samples of 1's and 1's, 1's and 3's, 1's and 5's, 1's and 7's, 1's and 9's, then we normalise by the largest number of features shared; the procedure is repeated for the full grid. Larger and darker circles indicate sharing of more factors. As expected, observations depicting the same digits have the most shared factors; 1's and 7's also share significant

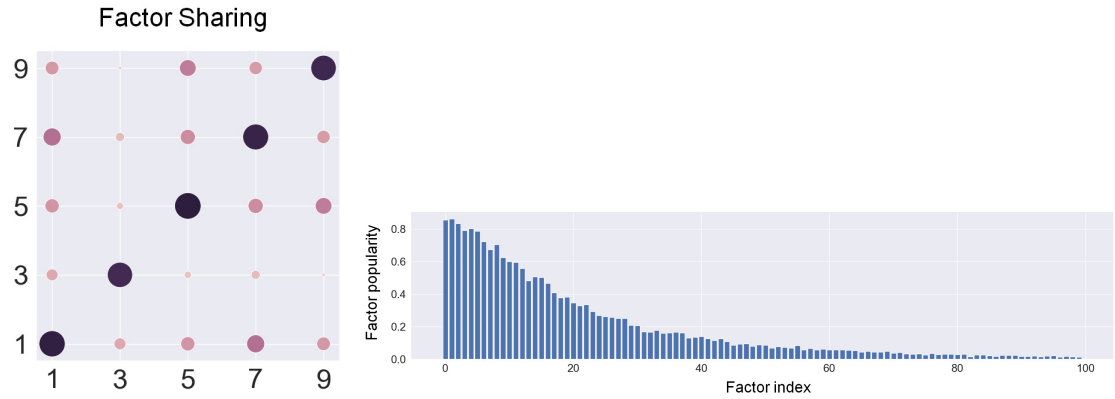


Figure 5: aFA model parameters estimated on 2500 odd-labelled MNIST digits, 500 of each label. Factor sharing grid between digits (left): circles are sized according to the number of features shared between digit pairs denoted on the x-axis and y-axis; colour enforces this effect where darker circles indicate more sharing and brighter circles, less. Distribution of feature allocation processes (right): y-axis denotes the proportion of data sharing the current factor; x-axis indicates the factor number where the factors are ordered from the most (left) to the least (right) frequently allocated, with a small number of data points allocated (right).

structure as well as 5's and 9's which broadly coincides with the geometry of the digits. The results can be directly compared with a similar experiment in Paisley and Carin (2009b). In Figure 5 we display the estimated feature weights obtained by summing over the Z matrix and normalising. Varying L and K one can study how well sparse and dense aFA models infer features specific to the different digits.

5.3 Visualization with aPPCA

Despite the increased popularity of nonlinear manifold embedding algorithms for data visualization, linear dimensionality reduction methods remain of fundamental importance to exploratory data visualization, arguably due their scalability, stability, and transparent data representation. In this section, we provide a simple illustrations of how latent feature PCA complements conventional PCA visualizations. Typically we use PCA to project all of the data down to the first two PCs, in aPPCA each point is also reduced to say $L = 2$ components, but these components are computed using only a fraction of the data, having some larger K unique sparse PCs in total. This implies we visualise the data using multiple scatter plots, each plot showing different subsets of the projected data instead of a single, crowded plot obtained using PCA.

5.3.1 MNIST Data set

First, we look at subspace sharing of MNIST digits. For more intuitive visualization, we first use a two-layer multilayer perceptron variational autoencoder (VAE) Kingma and Welling (2013) to reduce the dimension of the 10,000 MNIST digits. The 784-dimensional data is reduced using the VAE to 10 dimensions, and then we estimate the aPPCA model param-

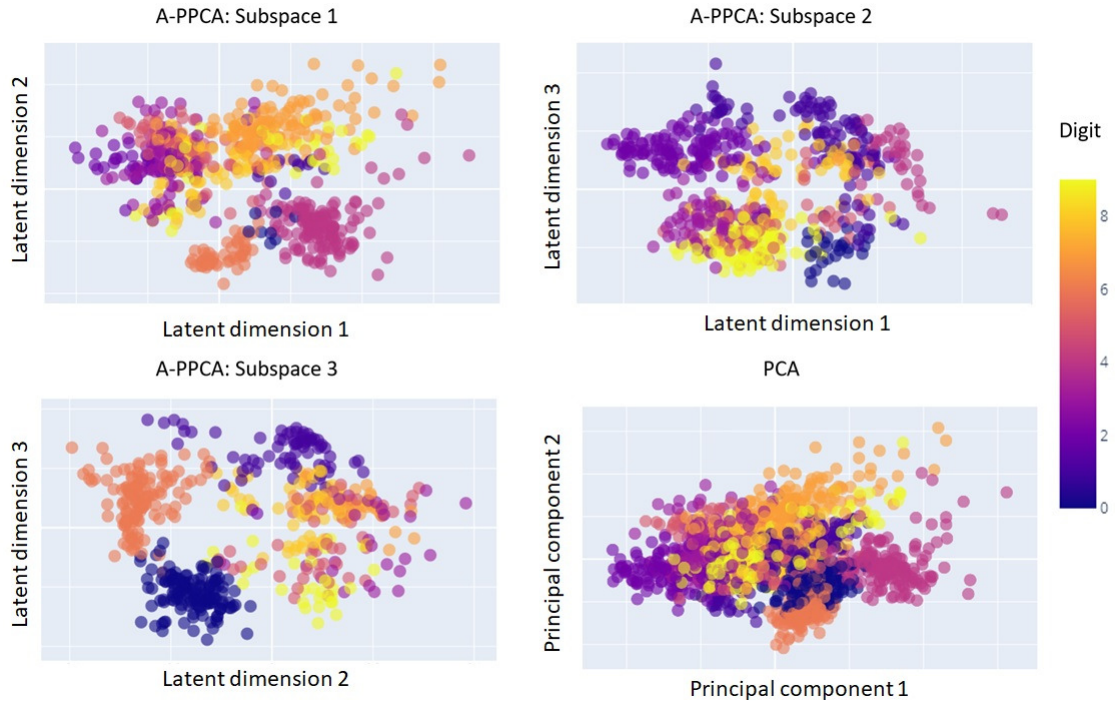


Figure 6: Scatter plot of 2D projections of 10,000 MNIST digits, obtained using aPPCA and PCA. The first three subplots contain only proportions of the data which have been estimated by aPPCA to lie in the corresponding subspace (i.e., subspace 1 is spanned by features 1 and 2; subspace 2 by features 2 and 3; and subspace 3 by 1 and 3). The fourth subplot shows the 2D projection of all digits obtained using PCA.

eters with $K = 3$ and $L = 2$ to visualise the digits in the latent space. We will assume that subspace 1 is spanned by the inferred features 1 and 2; subspace 2 by features 2 and 3; subspace 3 by features 1 and 3. Note that all pairs of subspaces share one of their principal axes. Figure 6 displays the reduced data in each of these subspaces, where we can see increased separation between many of the distinct clusters of different digits. From Figure 7 we can see that distinct geometric properties of digits are encoded in the different latent subspaces. Figure 7 shows randomly selected digits from each subspace; we can see that most digits in subspace 1 are written in thicker font; most digits in subspace 3 are slanted. This visualization reduces the crowding effect of PCA (Maaten and Hinton, 2008) and produces multiple 2D plots which jointly decompose and organize the observed data.

5.3.2 COIL-20 Data set

We consider another data visualization example, this time using data from the Columbia University Image Library (COIL-20) (Nene et al., 1996). The data set contains low resolution images (32×32 pixels) of 20 different objects. The objects are placed on a motorised turntable against a blank background and the turntable is rotated through 360 degrees to vary object pose with respect to a fixed camera. 72 images of each object are taken, at pose intervals of 5-degree rotation, and the images are size normalised. This means that

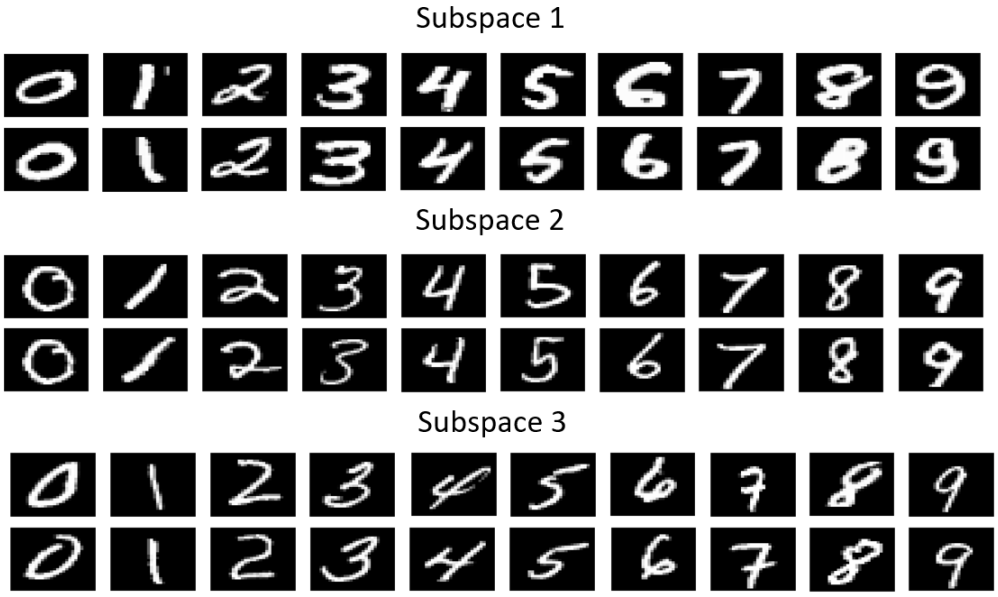


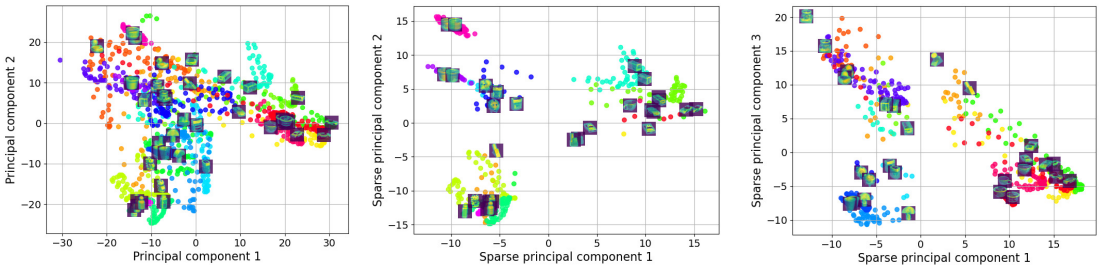
Figure 7: Randomly selected MNIST digits from each of the identified subspaces. The top panel consist of mostly thicker digits; the bottom panel is dominated by slanted digits.

objects which are very similar at different view angles, will result in very similar 72-image observations. First, we reduce all 1,440 images onto the two PCs which are computed to preserve the variance globally across all data points. Images from the different objects are displayed in different colours in Figure 8, whereas a fraction of the actual images is overlaid on the scatter plot. We see that some of the objects, such as two of the toy cars framed from front view angle (i.e., green, and yellow class on the far right of the plot), are well separated into a group with other rectangular objects having similar geometry. However, most of the objects are bundled in the centre of the plot and not distinguishable in the reduced, global 2D latent space.

Next, we fit an aPPCA model with $K = 4$ and $L = 2$ which effectively learns four sparse PCs with each data point associated to a subset of exactly two of these components. In (b)-(c) of Figure 8, we display the points sharing combinations of the estimated subspaces spanned by the sparse principal components (i.e., four unique components leads to $(4 \times 3)/2 = 6$ subspaces with some shared axes). We see that projections onto the sparse PCs reduce the crowding effect of PCA. In addition, the different sparse PCs encode interpretable geometric properties of the objects observed. For example, objects with smaller values along the sparse PC number 2, tend to be narrower, whereas objects with large values along the sparse PC number 1 tend to be less cylindrical. Within each 2D subspace, the different object projections are easier to separate and different objects with similar projections also have visually clear image similarity under a rotation angle.

5.3.3 Interpreting global structure in manifold embedding

Toy problems such as COIL-20 have been used to showcase manifold embedding methods such as t-SNE (Maaten and Hinton, 2008) and more recently UMAP (McInnes et al.,

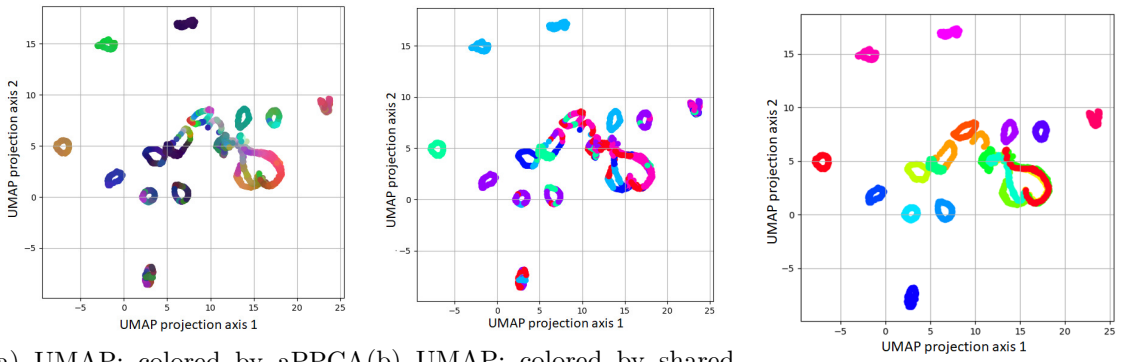


(a) PCA (b) aPPCA: subspace $\{1, 2\}$ (c) aPPCA: subspace $\{1, 3\}$

Figure 8: 2D projections of the COIL-20 data set images using PCA and aPPCA methods. (a) 2D scatter plot obtained by reducing the 1,024D images to 2D with (global) PCA, a sample of the original images is placed over their projection. (b)-(c) 2D projections of data points onto the sparse principal components with which they they are associated, inferred using aPPCA. Where (a) includes all data points in a single projection, aPPCA in (b)-(c) identifies subsets of the data sharing principal components, hence principal components are estimated using only a subset of the observations (i.e., sparse principal components).

2018). Empirically, both t-SNE and UMAP often lead to very good class separability in the lower dimensional projections particularly in scenarios where class separability in the original high-dimensional data is good (i.e., such as for COIL-20). At the same time, it is well known that many manifold embedding algorithms such as UMAP and t-SNE do not preserve the global structure of the data manifold, unlike linear methods such as PCA and multidimensional scaling, or kernel space models such as Gaussian process latent variable models. This often leads to lower dimensional projections which reflect class separability well when captured in localised regions of the manifold (such as in MNIST and COIL-20), but do not capture similarities across different classes adequately. To illustrate, Figure 9 shows 2D projections of COIL-20, obtained using UMAP. On the right (Figure 9c), objects from the same class are associated with the same colour. Certain objects have been separated into 2 or 3 clusters (i.e., the duck and the bowl), depending on the angle of view, but if the aim is object classification based on 2D embedding of the data, the task is nearly trivial. The challenge is less clear if we are looking to uncover latent structure between the objects.

McInnes et al. (2018) has suggested using PCA to reduce data onto its first three PCs and colour UMAP embeddings using RGB values defined by the 3D PCA projections of each point. This approach suggests that points close in the PCA projection of the data, would also have a similar colour. By contrast, as colours transition, this means that data points are projected far apart on some of the PCs. The problem with using PCA as diagnostics for UMAP projections in this manner, is that we are likely to crowd observations, overestimating proximity between most points due to the simplistic, global assumptions of PCA. If we are interested in using manifold embedding methods such as UMAP, which preserve the local structure of the original manifold, we can instead use piecewise linear methods such as aPPCA, which capture the global structure of the manifold, and use these to annotate the 2D UMAP projections (Figure 9b). In this figure, we use different symbols to denote points associated with different subspaces; the colours depend on the



(a) UMAP: colored by aPPCA embeddings

(b) UMAP: colored by shared subspaces

(c) UMAP: colored by classes

Figure 9: 2D projections of the COIL-20 data set images using UMAP. The x-axis and y-axis are determined based on the UMAP projection. In (a) the colours encode the non-zero 3D projection of the points performed using aPPCA. Each point is associated with exactly three sparse PCs, but the total number of components is larger. In (b) the colours encode subspace sharing with same colours (i.e., points in the same colour share at least two sparse PCs). In (c) the colours encode indicate the object classes.

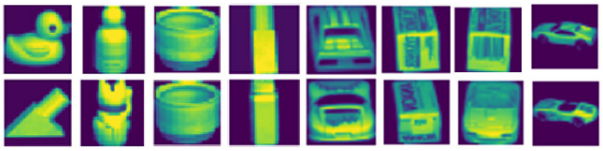


Figure 10: Example images from different object classes in the COIL-20 data set, with shared subspaces and close proximity in the 3D orthogonal aPPCA projection of input images. Proximity was defined with basic K-means clustering of the lower dimensional projections, where Figure 8 shows how clustered specific subspaces are. Note that objects sharing subspaces are merely estimated to have a shared covariance structure.

3D projection obtain with a single run of aPPCA with $K = 4$ and $L = 3$ (i.e., leading to four subspaces spanned by sparse PCs $\{1, 2, 3\}$; $\{2, 3, 4\}$; $\{1, 2, 4\}$ and $\{1, 3, 4\}$). Note that under this diagnostic, similar colours (in RGB values) indicate similarity in the reduced form. We can see that aPPCA much of the omitted cross-object similarities is specific to certain rotations: the rotated Maneki-neko (i.e., lucky cat figurine) and cylindrical bottle; the duck toy and the similar shape wooden part; the different clusters of bowl images and others. To further aid intuition, we have also included images of rotated object similarities identified using subspace decomposition diagnostics with aPPCA, see Figure 10.

5.4 Data pre-processing

Another ubiquitous use of PCA is *data whitening*. This is an often-used pre-processing step that aims to *decorrelate* the observed data to simplify subsequent processing and analysis, for example, image data tends to have highly correlated adjacent pixels. In this capacity,

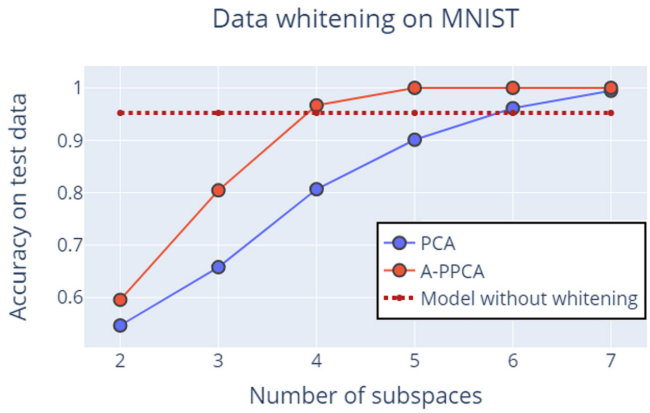


Figure 11: Classification accuracy of a multilayer perceptron was evaluated using 10,000 MNIST digits in three different setups: no whitening; data pre-processed with PCA; data pre-processed with adaptive probabilistic PCA (aPPCA). On the x-axis we show the number of reduced dimensions for different instances of the same classifier. The y-axis indicates the out-of-sample digit classification accuracy, evaluated using 10-fold cross-validation.

PCA works by "rotating" the data in observation space, retaining dimensionality, unlike visualization applications.

Here we show a simple example demonstrating how aPPCA can be used to do more effective *local* whitening which can lead to more accurate and interpretable supervised classification in decorrelated latent feature space. To demonstrate this, we compare a classifier trained on raw data with the same classifier trained on the first few PC projections of the data where the PCs are estimated (1) globally using PCA and (2) locally, within subsets of the data using aPPCA. For simplicity, we show an example of pre-processing the MNIST handwritten digit classification data set, before training a multilayer perceptron. We train a simple multilayer perceptron with one hidden layer with a softmax activation function evaluated using 10-fold cross-validation on $10,000$-image subset of the $784-D$ MNIST data set. We compare the performance of the same classifier network when (1) trained on the original 784-D pre-processed data, (2) trained on lower-dimensional projection of the data using PCA (3) trained on data locally whitened by aPPCA (K-dimensional). The classifier is a multilayer perceptron in all three scenarios. Figure 11 shows the classification accuracy of these three different pre-processing approaches as we vary K, i.e., the number of PCs onto which we can project the data. For aPPCA, we have kept $L = K - 1$ for simplicity. Intuitively, we also see increases in performance if multiple, separate classifiers are trained on each L-dimensional subspace, but usually, after whitening with PCA, a single classifier is used.

A key feature of aPPCA for localised data whitening is that it estimates more robust subspaces than global PCA, which can be seen in the smaller number of subspaces (i.e., PCs or columns of $\mathbf{W}$) required for training of the same classifier to achieve the same level of out-of-sample prediction performance. The multilayer perceptron trained on PCA whitened data requires more subspaces in training to achieve comparable out-of-sample performance.

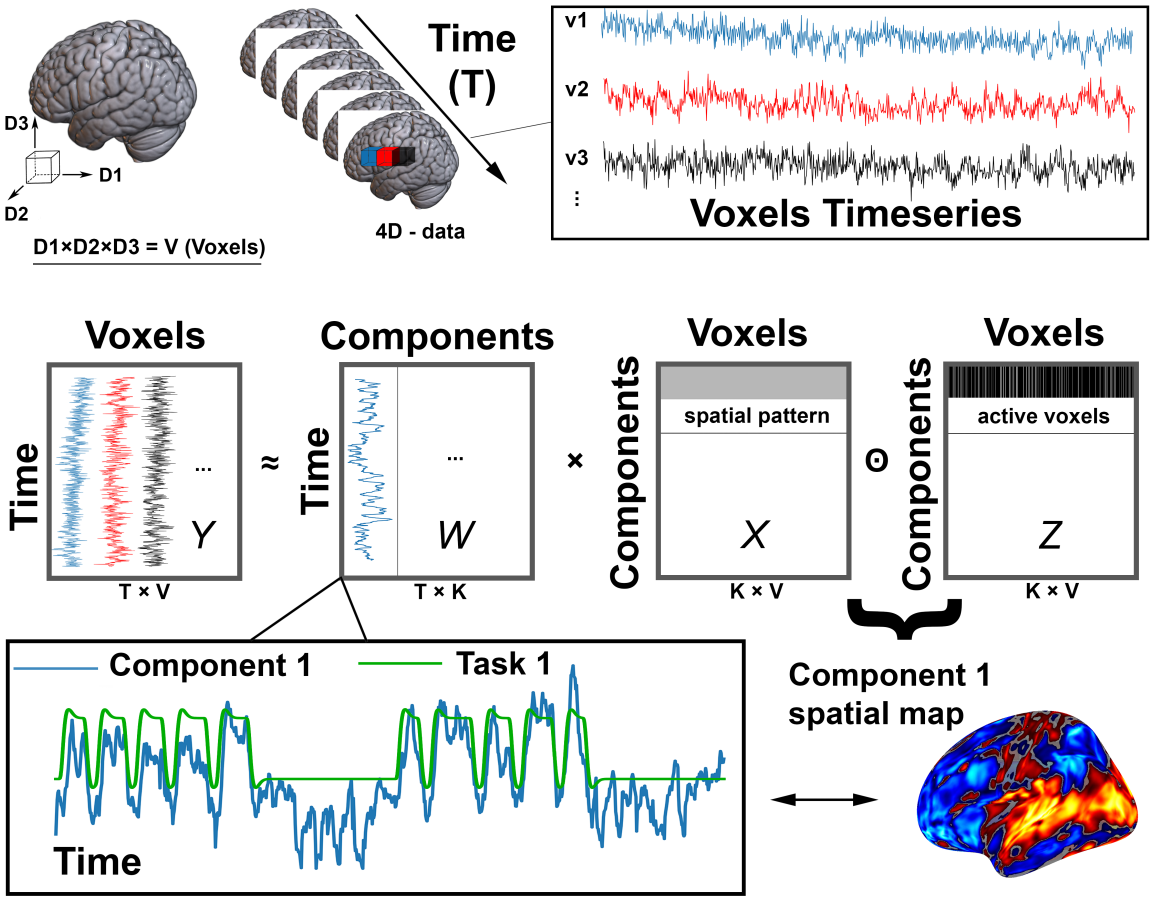


Figure 12: fMRI data of 3D brain volumes was collected over time (e.g. every 0.8 seconds). Typically, images are vectorised and represented as 2D $\mathbf{T} \times \mathbf{V}$ matrices (top panel), with $\mathbf{V}$ being the number of all voxels in all dimensions and $\mathbf{T}$ the number of time instances. This matrix can then be reduced to a $\mathbf{K} \times \mathbf{V}$ matrix (i.e., $\mathbf{X}$) which represents spatial maps of regions with intrinsically similar time courses (middle panel). $\mathbf{W}$ denotes the modelled transformation matrix and $\mathbf{Z}$ indicates whether components (i.e., rows of $\mathbf{X}$) should be included in the representation of the data matrix or not. Columns of $\mathbf{W}$, also referred to as components, are easier to interpret in terms of their correlation to experimental stimuli.

5.5 Blind source separation in fMRI

Functional magnetic resonance imaging (fMRI) is a technique for the non-invasive study of brain function. fMRI can act as an indirect measure of neuronal activation in the brain, by detecting blood oxygenation level-dependent (BOLD) contrast (Zarahn et al., 1997). BOLD relies on the fact that oxygenated (diamagnetic) and deoxygenated (paramagnetic) blood have different magnetic properties. When neurons fire there is an increase in localised flow of more oxygenated blood, which can be detected using BOLD fMRI.

fMRI time-series data is often represented as a series of 3D images (see Figure 12). However, this data can be also represented as a 2D matrix using vectorised voxel matrices

over time (time by voxels). In this representation, each matrix row contains all voxels from the brain image (or the subset selected for analysis) from a single time instant. Although useful, fMRI data often suffers from a low image contrast-to-noise ratio, it is biased by subject head motions, scanner drift (i.e., due to equipment overheating), and signals from irrelevant physiological sources (cardiac or pulmonary). Therefore, direct analysis of raw fMRI measurements is rare (Pruim et al., 2015) and domain experts tend to work with pre-processed, reduced statistics of the data. In clinical studies, due to the typical scarcity of fMRI series per subject and the low signal-to-noise ratio, flexible black-box algorithms are rarely used. The preferred methods for pre-processing of fMRI series and localisation of active spatial regions of the brain are variants of linear dimensionality reduction methods such as PCA and FA (Calhoun et al., 2003; Taghia et al., 2017; Beckmann and Smith, 2005; Pruim et al., 2015; Højen-Sørensen et al., 2002). Typically, of primary interest is then analysis of a representative subset of the inferred PCs or factors respectively, instead of the use of raw data.

A key problem with this approach is that these simple methods assume that the components/factors are a linear combination of all of the data, i.e., in other words, PCA and FA assume that all components are *active* for the full duration of the recording. Common implementations for fMRI series (McKeown et al., 2003; Calhoun et al., 2009) might threshold the inferred components, or use sparse versions of the decomposition techniques. These can still lead to biased component decomposition, and we are likely to overestimate the size of the activation area of the brain for some components and completely overlook functional areas of the brain which are active for short periods of time. Here, we show that our proposed *adaptive* linear methods, are better-motivated models for alleviating this problem and can infer better localised spatial regions of activation from fMRI. Furthermore, we can potentially discover novel short-term components in a principled, probabilistic, data-driven fashion.

As a proof of concept, here we apply aPPCA to fMRI data collected from a single participant while exposed to continuous visual stimuli (Raykov et al., 2021). The fMRI data were initially realigned to correct for subject motion and registered to a group template (Montreal Neurological Institute Template). Using a 3T Siemens scanner, a whole brain image with a voxel resolution of $2 \times 2 \times 2$ mm was acquired every 0.8 seconds. The data had $215,302$ voxels and 989 time instances. The aPPCA decomposition was performed by treating time instances as features, which is standard procedure in the neuroimaging field. For aPPCA we used $K = 500$ unique components and constraint of $L = 200$ components, which were selected to achieve component similarity with the benchmark and enable visually intuitive comparisons. We also performed PPCA with $K = 200$ components for comparison, see Figure 13.

The figure shows the component most associated with the task estimated both with aPPCA and PPCA. aPPCA results in sparser maps across space, which enhance localisation. This sparsity increases with higher numbers of components that explain less variance in the data. This can be useful for identifying noisy components and brain areas that are only transiently active during task performance. We also show the corrected t-statistic map (Figure 13) which shows the voxels that have a significant correlation with the visual stimuli. The map is family-wise error (FWE) rate corrected at $p < 0.05$ at voxel threshold

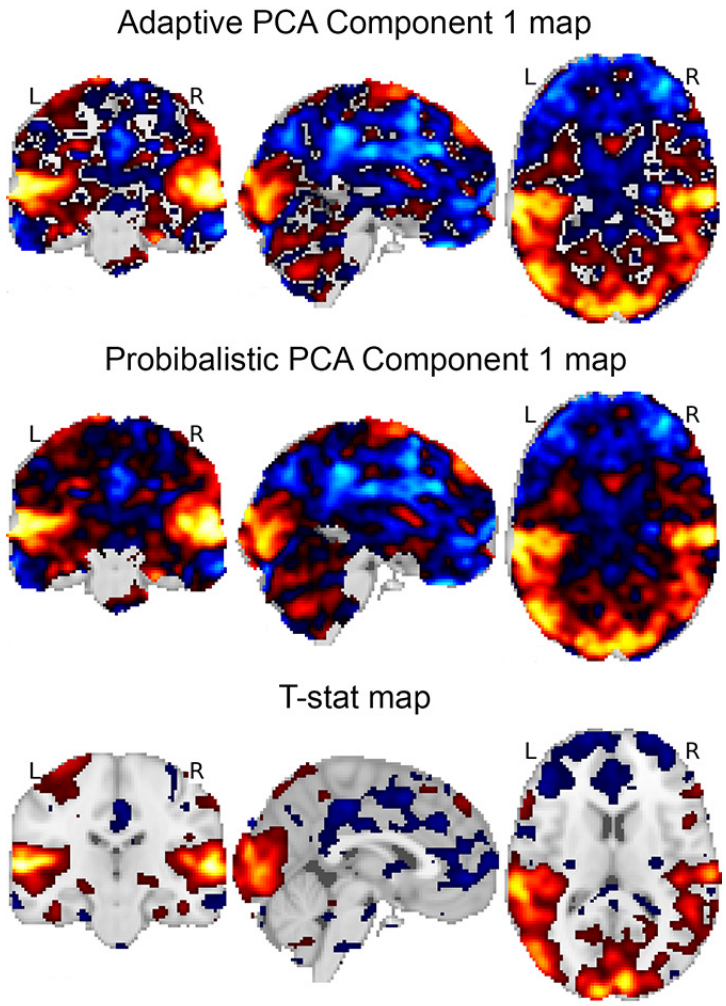


Figure 13: Lower dimensional fMRI recording reduced across time, plotted against the subject brain. The fMRI time series of length T is reduced to K components and here we display the single component most associated with the stimuli during the experiment. Reduced projection estimated using aPPCA (top) and projection estimated using PPCA (middle). The larger numbers of grey regions indicate that aPPCA projection better localises regions of the brain fluctuation through time, as a response to the visual stimuli. Reference regions of activation can be seen from the *t-map* (bottom) displaying the correlation of the component with the ground-truth visual stimuli.

$p < 0.001$. One benefit of decomposition methods versus standard correlation methods is that they do not need a predefined model of assumed task activation.

Direct quantitative evaluation of pre-processing tools for fMRI data is an open problem, due to the lack of a clear ground-truth definition of brain-activity-related components. We have measured the mean reconstruction error across all $215,302$ voxels as well as the standard deviation across voxels. We find that the highest error with the highest standard

deviation, average root mean square error (RMSE) of **16.5**, and the standard deviation of RMSE of **4.8**), was obtained using PPCA. aPPCA reconstruction gradually reduces these errors depending upon the ratio of K to L, with the best scoring reconstruction having an average RMSE of **14.1** and standard deviation RMSE (across voxels) of **3.0**. The lower standard deviation of error across voxels supports our hypothesis of better-preserved local region information using aPPCA. Due to the simplicity of the imaging setup, both methods were able to identify components highly correlated to the stimuli (Figure 13). The typical goal for experts would be to examine functions of the specific brain regions or networks, as well as potentially affected areas of the brain after head trauma or stroke.

The typical practice would be to threshold the observation-specific loadings (i.e., reduced form data) and only consider voxels that *significantly* contribute to selected subsets of components. The adaptive nature of aPPCA allows us to infer the voxel association with specific components (i.e., $\mathbf{Z}$ switches off voxels not part of a component) in a principled fashion as a part of a fully probabilistic model. In addition, the experimental user has explicit control over the contrast voxels used in different components (ratio of K to L) and this can be useful for achieving better spatial localisation without thresholding, which is an inherently subjective and overly simplistic, procedure.

6. Conclusion

In this work, we have studied generic discrete latent variable augmentation for ubiquitous linear Gaussian methods applied for feature learning, whitening, and dimensionality reduction applications. The manuscript details shortcomings of existing Bayesian nonparametric linear Gaussian methods and demonstrates that flexible alternatives can be derived by modelling discrete latent features without replacement, such as latent hypergeometric models and truncated multinomial latent feature models. We propose two such novel frameworks, the aFA, and aPPCA models, which overcome the inherent over-partitioning in Beta processes, allowing for more flexible regularisation of the model capacity when compared to Beta-Bernoulli models. The proposed models can be extended to many other related methods such as generalised linear Gaussian models, Gaussian process latent variable models (GPLVMs), kernel PCA methods, and others. Dai et al. (2015) has already introduced the problem of handling discontinuity in GPLVMs and proposed a simple *spike and slab prior* to augment the continuous latent variables in GPLVMs. Augmenting GPLVMs with discrete hypergeometric feature allocation indicators, would in principle allow for a richer and more compact model of the manifold using a smaller number of underlying, feature-specific, Gaussian processes.

In our study of aPPCA models, we have also proposed efficient practical inference methods for distributions on Stiefel manifolds. The utility of the proposed tools is demonstrated on synthetic latent feature Gaussian data sets, MNIST handwritten digit images, COIL-20 object images, and brain imaging fMRI data. The synthetic data study shows that a wide range of feature allocation distributions can be captured with our novel multivariate hypergeometric model. We have applied aPPCA to MNIST variational autoencoder projections, showing that it can be used to identify images sharing clear geometric features. aFA was applied to nearly raw MNIST digits to show that images of visually similar digits share more factors than visually distinct digits. We conclude with an application of aPPCA to

a widely encountered problem in brain imaging with fMRI and demonstrate an accurate decomposition of active spatial regions in the brain during different stimuli (or at rest). We also demonstrate that this discrete-continuous decomposition leads to more accurate localisation of active brain regions. This finding has the potential to lead to significant improvements to analysis pipelines for fMRI data for neurological screening and cognitive neuroscience applications.

Acknowledgments

We would like to thank the associated editor and the reviewers for the useful suggestions, which improved significantly this work from its earlier versions. We also would like to express gratitude to Prof. David Saad and Prof. David Lowe for the useful feedback on this work in its earlier forms. Y. P. Raykov was supported by the Engineering and Physical Sciences Research Council [Trusted Data Driven Products EP/T022493/1]. P. Raykov was supported by the Economic and Social Research Council [studentship grant ES/J500173/1] and [fellowship grant ES/V012444/1].

Appendix A. Adaptive PCA

In this section, we demonstrate that the proposed aPPCA model from Section 4 is indeed a generalisation of the ubiquitous PCA using *small variance asymptotics* Broderick et al. (2013). Let us first start by marginalising the discrete and continuous latent variables $\{\mathbf{x}_n, \mathbf{z}_n\}$ which are not of explicit interest in conventional PCA. To compute the marginal likelihood of $\mathbf{y}_n$ we compute the expectations

$$\mathbb{E}_{\mathrm{P}(\mathbf{x}_n,\mathbf{z}_n)}\left[\mathbf{y}_n\right] \quad \text{and} \quad \mathbb{E}_{\mathrm{P}(\mathbf{x}_n,\mathbf{z}_n)}\left[\left(\mathbf{y}_n - \mathbb{E}\left[\mathbf{y}_n\right]\right)\left(\mathbf{y}_n - \mathbb{E}\left[\mathbf{y}_n\right]\right)^T\right] \tag{21}$$

where we use $\mathbb{E}\left[\,\right] = \mathbb{E}_{\mathrm{P}(\mathbf{x}_n,\mathbf{z}_n)}\left[\,\right]$ for notational convenience. We express the moments of the marginal likelihood starting with the posterior mean of the marginal, $\mathbb{E}\left[\mathbf{y}_n\right]$

$$\begin{aligned}
\mathbb{E}\left[\mathbf{y}_n\right] &= \mathbb{E}\left[\mathbf{W}\left(\mathbf{x}_n \odot \mathbf{z}_n\right) + \boldsymbol{\mu} + \boldsymbol{\epsilon}_n\right] \\
&= \mathbf{W}\left(\mathbb{E}\left[\mathbf{x}_n\right] \odot \mathbb{E}\left[\mathbf{z}_n\right]\right) + \boldsymbol{\mu} + \mathbb{E}\left[\boldsymbol{\epsilon}_n\right] \\
&= \mathbf{W}\left(0 \odot \boldsymbol{\rho}\right) + \boldsymbol{\mu} + 0 \\
&= \boldsymbol{\mu}
\end{aligned}$$

where we have used a diagonal $(K \times K)$ matrix $\boldsymbol{\rho}$ to denote the expectation of each feature, which is determined by the prior on the matrix $\mathbf{Z}$

$$\rho_{k,k} = \begin{cases} \frac{L}{K} & \text{if multivariate hypergeometric prior} \\ \frac{1}{N}\sum_n z_{k,n} & \text{if IBP prior} \end{cases} \tag{22}$$

For the variance of the marginal, we can write

$$\begin{aligned}
\mathbb{E}\left[\left(\mathbf{y}_n - \mathbb{E}\left[\mathbf{y}_n\right]\right)\left(\mathbf{y}_n - \mathbb{E}\left[\mathbf{y}_n\right]\right)^T\right] &= \mathbb{E}\left[\left(\mathbf{W}\left(\mathbf{x}_n \odot \mathbf{z}_n\right) + \boldsymbol{\epsilon}_n\right)\left(\mathbf{W}\left(\mathbf{x}_n \odot \mathbf{z}_n\right) + \boldsymbol{\epsilon}_n\right)^T\right] \\
&= \mathbf{W}\boldsymbol{\rho}\mathbf{W}^T + \sigma^2\mathbf{I}_D.
\end{aligned}$$

Finally, using the obtained expression for $\mathbb{E}\left[\mathbf{y}_n\right]$ and $\mathbb{E}\left[\left(\mathbf{y}_n - \mathbb{E}\left[\mathbf{y}_n\right]\right)\left(\mathbf{y}_n - \mathbb{E}\left[\mathbf{y}_n\right]\right)^T\right]$, combined with the Gaussian likelihood of $\mathbf{y}_n$ resulting in a linear Gaussian model, we can write the marginal likelihood as

$$\mathrm{P}\left(\mathbf{y}_n \mid \mathbf{W}, \boldsymbol{\rho}, \sigma^2\right) = \frac{1}{\left(2\pi\right)^{\frac{D}{2}}}\left|\mathbf{C}\right|^{-1/2}\exp\left(-\frac{1}{2}\mathbf{y}_n^T\mathbf{C}^{-1}\mathbf{y}_n\right) \tag{23}$$

where we used $\mathbf{C} = \mathbf{W}\boldsymbol{\rho}\mathbf{W}^T + \sigma^2\mathbf{I}_D$ to denote the model covariance.

Now, the marginal likelihood in this collapsed aPPCA model is almost identical to the PPCA model (Tipping and Bishop, 1999b) with the key difference being the weights $\boldsymbol{\rho}$ which can be a single scalar shared across all dimensions, or direction specific. In fact, we can say that the PPCA model is a special case of the collapsed aPPCA model when the diagonal of $\boldsymbol{\rho}$ are full of ones, which occurs when the matrix $\mathbf{Z}$ is full of ones, in turn implying all observations are active in all K numbers of 1D subspaces.

The complete data log-likelihood of the collapsed model is

$$\begin{aligned}\mathcal{L} &= \sum_{n=1}^{N} \ln\left(\mathrm{P}\left(\mathbf{y}_n \mid \mathbf{W}, \boldsymbol{\rho}, \sigma^2\right)\right) \\ &= -\frac{N}{2}\left(D\ln(2\pi) + \ln|\mathbf{C}| + tr\left(\mathbf{C}^{-1}\mathbf{S}\right)\right)\end{aligned}$$

where $\mathbf{S} = \frac{1}{N}\mathbf{Y}\mathbf{Y}^T$. To find the maximum likelihood estimate for $\mathbf{W}$, we differentiate the likelihood and solve

$$\frac{d\mathcal{L}}{d\mathbf{W}} = -\frac{N}{2}\left(2\mathbf{C}^{-1}\mathbf{W}\boldsymbol{\rho} - 2\mathbf{C}^{-1}\mathbf{S}\mathbf{C}^{-1}\mathbf{W}\boldsymbol{\rho}\right) = 0. \tag{24}$$

The maximum likelihood estimate for $\mathbf{W}$ should then satisfy:

$$\begin{aligned}\mathbf{C}^{-1}\mathbf{W}\boldsymbol{\rho} &= \mathbf{C}^{-1}\mathbf{S}\mathbf{C}^{-1}\mathbf{W}\boldsymbol{\rho} \\ \mathbf{W}^{\mathrm{ML}}\boldsymbol{\rho} &= \mathbf{S}\mathbf{C}^{-1}\mathbf{W}^{\mathrm{ML}}\boldsymbol{\rho}\end{aligned}$$

To find the solution for the above we first express the $\mathbf{W}\boldsymbol{\rho}^{1/2}$ term using its singular value decomposition:

$$\mathbf{W}\boldsymbol{\rho}^{1/2} = \mathbf{U}\mathbf{L}\mathbf{V}^T \tag{25}$$

which leads to

$$\mathbf{C}^{-1}\mathbf{W}\boldsymbol{\rho}^{1/2} = \mathbf{U}\mathbf{L}\left(\mathbf{L}^2 + \sigma^2\mathbf{I}_K\right)^{-1}\mathbf{V}^T.$$

Then, we compute

$$\begin{aligned}\mathbf{S}\mathbf{C}^{-1}\mathbf{W}\boldsymbol{\rho}^{1/2} &= \mathbf{W}\boldsymbol{\rho}^{1/2} \\ \mathbf{S}\mathbf{U}\mathbf{L}\left(\mathbf{L}^2 + \sigma^2\mathbf{I}_K\right)^{-1}\mathbf{V}^T &= \mathbf{U}\mathbf{L}\mathbf{V}^T \\ \mathbf{S}\mathbf{U}\mathbf{L} &= \mathbf{U}\left(\mathbf{L}^2 + \sigma^2\mathbf{I}_K\right)\mathbf{L}\end{aligned}$$

which implies that $\mathbf{u}_j$ is the eigenvector of $\mathbf{S}$ with eigenvalue of $\lambda_j = \sigma^2 + l_j^2$. Therefore, all potential solutions for $\mathbf{W}^{\mathrm{ML}}$ may be written as

$$\mathbf{W}^{\mathrm{ML}} = \mathbf{U}_K\left(\mathbf{K}_K - \sigma^2\mathbf{I}_K\right)^{1/2}\mathbf{R}\boldsymbol{\rho}^{-1/2} \tag{26}$$

where

$$k_{jj} = \begin{cases}\lambda_j & \text{eigenvalue of } \mathbf{u}_j \\ \sigma^2 & \text{otherwise}\end{cases} \tag{27}$$

where $\mathbf{R}$ is $(D \times K)$ orthonormal matrix. The weighting term $\boldsymbol{\rho}$ allows explicit control over the scale of the different projection axes; it determines if we should place importance on the role of the input to the projection axis, which is meant to reflect our posterior belief of re-scaling, again because not all data points share all subspaces. Appropriate scaling with $\boldsymbol{\rho}$ can address the well known pitfalls of PCA such as the disproportionate crowding of the projections due to outliers or multi-modalities; the sphericalisation of the projection.

Appendix B. Updating hyperparameters

In this section we list a Bayesian procedure for inferring the hyperparameters of aPPCA.

B.1 Updating σ^2

We place an inverse-Gamma prior on σ^2 with parameters $\{\gamma, \vartheta\}$

$$\mathrm{P}\left(\sigma^2|\gamma,\vartheta\right) = \frac{\vartheta^{\gamma}}{\Gamma\left(\gamma\right)}\left(\sigma^2\right)^{-\gamma-1}\exp\left[-\frac{\vartheta}{(\sigma^2)}\right]. \tag{28}$$

This leads to posterior distribution over σ^2 of the form

$$\begin{aligned}\mathrm{P}\left(\sigma^2|\gamma,\vartheta,\mathbf{Y},\mathbf{W},\mathbf{X},\mathbf{Z}\right) &= \frac{\vartheta^{\gamma}}{\Gamma\left(\gamma\right)}\left(\sigma^2\right)^{-\gamma-1}\exp\left[-\frac{\vartheta}{\sigma^2}\right] \\ &\times \frac{1}{\left(2\pi\sigma^2\right)^{\frac{ND}{2}}}\exp\left(-\frac{1}{2\sigma^2}\sum_{n=1}^{N}\left[\left(\mathbf{y}_n-\mathbf{W}\left(\mathbf{x}_n\odot\mathbf{z}_n\right)\right)^T\left(\mathbf{y}_n-\mathbf{W}\left(\mathbf{x}_n\odot\mathbf{z}_n\right)\right)\right]\right) \\ &\propto \left(\sigma^2\right)^{-(\gamma+ND/2)-1} \\ &\times \exp\left(-\frac{1}{\sigma^2}\left(\frac{1}{2}\mathrm{tr}\left[\left(\mathbf{Y}-\mathbf{W}\left(\mathbf{X}\odot\mathbf{Z}\right)\right)^T\left(\mathbf{Y}-\mathbf{W}\left(\mathbf{X}\odot\mathbf{Z}\right)\right)\right]+\vartheta\right)\right).\end{aligned}$$

which is still an inverse-Gamma distribution with parameters $\gamma^{post} = \gamma + \frac{ND}{2}$ and $\vartheta^{post} = \frac{1}{2}\mathrm{tr}\left[\left(\mathbf{Y}-\mathbf{W}\left(\mathbf{X}\odot\mathbf{Z}\right)\right)^T\left(\mathbf{Y}-\mathbf{W}\left(\mathbf{X}\odot\mathbf{Z}\right)\right)\right] + \vartheta$.

B.2 Updating α

We place a Gamma prior on the IBP concentration parameter α with parameters $\{\lambda, \mu\}$

$$\mathrm{P}\left(\alpha|\lambda,\mu\right) = \frac{\mu^{\lambda}}{\Gamma\left(\lambda\right)}\left(\alpha\right)^{\lambda-1}\exp\left[-\mu\alpha\right]. \tag{29}$$

This leads to posterior distribution over α of the form

$$\begin{aligned}\mathrm{P}\left(\alpha|\lambda,\mu,\mathbf{Y},\mathbf{W},\mathbf{X},\mathbf{Z}\right) &= \frac{\mu^{\lambda}}{\Gamma\left(\lambda\right)}\left(\alpha\right)^{\lambda-1}\exp\left[-\mu\alpha\right] \\ &\times \exp\left(-\alpha H_N\right)\alpha^K \times \left(\prod_{k=1}^{K}\frac{(m_k-1)!\,(N-m_k)!}{(N)!}\right) \\ &\propto \left(\alpha\right)^{\lambda+K-1}\exp\left(-\alpha\left(H_N+\mu\right)\right)\end{aligned}$$

which is still a gamma distribution with parameters $\lambda^{post} = \lambda + K$, $\mu^{post} = H_N + \mu$ and $H_N = \sum_{n=1}^{N}\frac{1}{n}$.

Appendix C. Projection matrix update using *Pymanopt*

For both variants of aPPCA, the matrix $\mathbf{W}$ is updated numerically by minimising the negative log of Equation (17) over the Stiefel manifold with respect to the matrix $\mathbf{W}$. Figure 14 shows the implementation of this using the Pymanopt toolbox Townsend et al. (2016).

```python
#Import libraries
import autograd.numpy as auto_np
from pymanopt.manifolds import Stiefel
from pymanopt import Problem
from pymanopt.solvers import SteepestDescent

#Define the Stiefel manifold
manifold = Stiefel(D, K)
#Define the cost function
def cost(W): return -auto_np.trace((X*Z).T@W.T@Y)/(2*sigma_Y**2)
#Define the problem
problem = Problem(manifold=manifold, cost=cost)
#Choose a solver
solver = SteepestDescent()
#Find solution for W
W = solver.solve(problem)
```

Figure 14: Python code for aPPCA updates on the rotation matrix $\mathbf{W}$ using Pymanopt toolbox.

Appendix D. Learning W

In this section we compare two different schemes of learning the projection matrix $\mathbf{W}$ while maintaining orthogonality. We generate synthetic data $\mathbf{Y} \in \mathbb{R}^{D\times N}$ which takes the form $\mathbf{Y} = \mathbf{W}(\mathbf{X} \odot \mathbf{Z}) + \mathbf{E}$ with $\mathbf{X} \in \mathbb{R}^{K\times N}$ a latent feature matrix with standard Gaussian distribution; $\mathbf{W} \in \mathbb{R}^{D\times K}$ is a projection matrix with orthogonal columns; $\mathbf{E} \in \mathbb{R}^{D\times N}$ is a noise matrix with multivariate Gaussian columns each with mean zero and covariance matrix $\sigma^2\mathbf{I}_D$ with $\sigma^2 = 0.01$. The core of the generative model remains the same across the different data sets we generate ($N = 1000$, $D = 35$), and only the number of latent features K and the indicator matrix $\mathbf{Z}$, changes. We have considered five separate synthetic data sets and the distribution of $\mathbf{Z}$ for each setup is displayed in Figure 3. We evaluate how two different schemes are able to learn the projection matrix $\mathbf{W}$; the first approach will sample each column of $\mathbf{W}$ using the von Misses-Fisher distribution (and then re-scale to maintain orthogonality), and the second method jointly optimises all columns of $\mathbf{W}$ on the Stiefel manifold using the Pymanopt toolbox Townsend et al. (2016). For each method we solely focus on learning $\mathbf{W}$, therefore other parameters are fixed to the true value. For each experiment $\mathbf{W}$ is estimated using 80% of the data and then tested on the remaining 20%; results are reported in Table 3.

Results from Table 3 show that the projection matrix $\mathbf{W}$ estimated using the Pymanopt toolbox has a lower mean squared prediction error when compared to estimating it with the von Misses-Fisher distribution.

Appendix E. Additional results from the synthetic study

In this section, we include some additional figures illustrating the performance of the different FA techniques from Section 5.1. We reproduce the results in Table 4 and Table 5 using different values for K^* of the data generation process. Since we assume unchange dimensionality $D = 35$, the evaluation simply probes the effect of having different lower dimensional data structure (i.e. inducing denser loadings and higher intrinsic dimensionality

Data		Pymanopt	von Misses-Fisher
Sparse matrix	$K = 10$	0.095 $\pm$0.002	0.323 $\pm$0.016
	$K = 20$	0.096 $\pm$0.002	0.471 $\pm$0.013
Balanced matrix	$K = 10$	0.095 $\pm$0.001	0.563 $\pm$0.029
	$K = 20$	0.096 $\pm$0.001	0.760 $\pm$0.017
Dense matrix	$K = 10$	0.095 $\pm$0.001	0.548 $\pm$0.026
	$K = 20$	0.096 $\pm$0.001	0.793 $\pm$0.017
Subsapce clustering	$K = 10$	0.093 $\pm$0.002	0.566 $\pm$0.020
	$K = 20$	0.096 $\pm$0.001	0.769 $\pm$0.010
Single State	$K = 10$	0.093 $\pm$0.001	0.614 $\pm$0.011
	$K = 20$	0.096 $\pm$0.003	0.920 $\pm$0.015

Table 3: Performance of two different schemes learning the projection matrix $\mathbf{W}$ on five different data sets of dimension $D = 35$ and latent features $K = 10$ & $K = 20$. For each experiment, $\mathbf{W}$ was estimated using 80% of the data and tested on the remaining 20%; average and 1-standard deviation of mean square prediction error (i.e. of predicting generating $\mathbf{W}$) over 20 different experiments is reported.

of the latent space). Table 4 reports the mean square reconstruction error and Table 5 the portion of inferred sparsity when different $K^* = 15$ is used.

The aFA model performs similarly across the different settings tested here, since the latent space structures in the synthetic data are all special cases for the multivariate hypergeometric model. For smaller generating number of factors K^* compare to D (Table 1 vs Table 4), we see systematically that isFA is associated with larger uncertainty for dense and balanced feature allocation matrices, whereas the sdFA for the sparse settings. The fsFA performs similarly to the conventional FA with slightly higher reconstruction error due to the sparsity regularization. The consistently higher reconstruction error for sdFA is most likely explained with the different indented use of sparse FA models where one would have to use significantly higher K to achieve comparable reconstruction error. However, an important benefit for this approach is that the relative performance remains consistent across different sparsity settings.

		Sparse				Balance				Dense				Class				Full			
$\frac{K}{2}$	aFA	1.22	1.60	2.01	4.78	2.42	2.70	3.24	5.46	3.78	3.77	5.12	7.35	1.43	2.22	2.14	5.59	5.07	5.44	5.78	7.90
	fsFA	1.34	1.75	2.27	4.12	2.62	2.99	3.55	4.13	4.11	4.12	5.59	4.06	1.56	2.46	2.33	4.04	5.48	5.90	6.26	4.01
	FA	1.37	1.57	2.05	3.18	2.58	3.06	3.32	3.09	4.38	4.15	5.35	3.01	1.64	2.43	2.25	3.01	5.51	5.83	5.78	2.92
	sdFA	0.04	0.35	1.26	4.92	0.03	**0.36**	1.32	6.32	**0.09**	0.41	1.58	6.42	0.72	0.45	1.62	5.22	0.06	0.45	1.25	5.88
	isFA	**0.03**	**0.26**	**1.10**	**4.61**	**0.02**	0.85	**1.20**	**6.20**	0.09	**0.30**	**1.42**	**6.11**	**0.06**	**0.30**	**0.27**	**4.53**	**0.06**	**0.28**	**1.06**	**5.76**
K	aFA	0.04	0.19	0.73	3.15	0.06	0.21	0.75	3.10	0.12	0.21	0.77	3.02	0.01	0.19	0.19	3.02	0.10	0.22	0.76	2.96
	fsFA	0.04	0.26	0.26	1.01	0.02	0.30	0.30	1.05	0.09	0.30	0.30	1.04	0.06	0.30	0.30	0.27	0.06	0.28	0.28	1.06
	FA	**0.01**	**0.18**	**0.18**	**0.73**	**0.01**	**0.18**	**0.18**	**0.73**	**0.01**	**0.18**	**0.18**	**0.74**	**0.01**	**0.18**	**0.18**	**0.18**	**0.01**	**0.18**	**0.18**	**0.73**
	sdFA	0.04	0.35	0.35	1.26	0.03	0.36	0.36	1.32	0.09	0.41	0.41	1.58	0.72	0.45	0.45	1.62	0.06	0.45	0.45	1.25
	isFA	0.29	0.27	0.27	1.10	0.02	0.85	0.85	1.20	0.09	0.30	0.30	1.43	0.06	0.30	0.30	0.27	0.06	0.28	0.28	1.06
$K+4$	aFA	**0.01**	**0.17**	0.70	**2.91**	**0.01**	**0.18**	**0**.72	2.91	**0.01**	0.18	0.73	2.88	0.01	0.18	0.18	2.82	0.01	0.18	0.72	2.83
	fsFA	0.04	0.26	1.0	4.13	0.02	0.29	1.02	4.06	0.09	0.30	1.05	4.03	0.06	0.30	0.27	4.06	0.06	0.28	1.04	3.91
	FA	0.01	0.17	**0.69**	2.99	0.01	0.18	**0.70**	**2.90**	0.01	**0.18**	**0.70**	**2.83**	**0.01**	**0.17**	**0.17**	**2.81**	**0.01**	**0.17**	**0.69**	**2.76**
	sdFA	0.04	0.35	1.26	4.92	0.03	0.36	1.32	6.32	0.09	0.41	1.58	6.42	0.72	0.45	1.62	5.22	0.06	0.45	1.25	5.88
	isFA	0.03	0.27	1.10	4.61	0.03	0.85	1.20	6.20	0.09	0.30	1.43	6.11	0.06	0.30	0.27	4.53	0.06	0.28	1.06	5.76
2K	aFA	**0.01**	**0.10**	**0.56**	**2.31**	**0.01**	**0.14**	**0.57**	2.30	**0.01**	**0.14**	**0.57**	2.25	**0.01**	**0.14**	**0.14**	**2.20**	**0.01**	**0.14**	**0.56**	2.23
	fsFA	0.04	0.25	1.00	4.02	0.02	0.28	1.01	4.01	0.09	0.29	1.01	3.94	0.06	0.29	0.26	4.00	0.06	0.27	1.00	3.90
	FA	0.01	0.13	0.60	2.35	0.01	0.15	0.60	**2.29**	0.01	0.16	0.60	**2.21**	0.01	0.15	0.15	2.21	0.01	0.15	0.59	**2.18**
	sdFA	0.04	0.35	1.26	4.92	0.03	0.36	1.32	6.32	0.09	0.41	1.58	6.42	0.72	0.45	1.62	5.22	0.06	0.45	1.25	5.88
	isFA	0.03	0.27	1.10	4.61	0.02	0.85	1.20	6.20	0.09	0.30	1.43	6.11	0.06	0.30	0.27	4.53	0.06	0.28	1.06	5.76
$3K$	aFA	**0.00**	**0.06**	**0.25**	0.94	**0.00**	**0.06**	**0.25**	0.98	**0.00**	**0.06**	**0.27**	1.01	**0.00**	**0.06**	**0.07**	0.94	**0.00**	**0.07**	**0.27**	1.01
	fsFA	0.04	0.25	0.97	4.01	0.02	0.28	0.98	3.88	0.09	0.29	0.98	3.91	0.06	0.29	0.26	3.97	0.06	0.26	0.99	3.80
	FA	0.01	0.10	0.44	**0.89**	0.01	0.11	0.42	**0.88**	0.01	0.11	0.44	**0.85**	0.01	0.11	0.11	**0.84**	0.01	0.11	0.42	**0.88**
	sdFA	0.04	0.35	1.26	4.92	0.03	0.36	1.32	6.32	0.09	0.41	1.58	6.42	0.72	0.45	1.62	5.22	0.06	0.45	1.25	5.88
	isFA	0.03	0.27	1.10	4.61	0.02	0.85	1.20	6.20	0.09	0.30	1.43	6.11	0.06	0.30	0.27	4.53	0.06	0.28	1.06	5.76

Table 4: Mean squared reconstruction error of different variants of factor analysis (FA) methods on five different data sets of dimension $D = 50$ and $K = 15$ latent features; the data generating process is described in Section 5.1. We have evaluated all methods and configurations using 4 different values of the noise parameter σ^2 controlling the signal-to-noise ratio in the generated data: $\sigma^2 = \{0.01, 0.25, 1, 4\}$ displayed left-to-right in each of the 5 main columns. Parameters for each model were estimated using 80% of the data set, and the resulting model was tested on the remaining 20%; average of the mean square out-of-sample reconstruction error over 20 different experiments, is reported. Standard deviation estimates have been omitted due to all estimates being < 0.005.

		Sparse				Balance				Dense				Class				Full			
$\frac{K}{2}$	aFA	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71
	fsFA	0.73	**0.63**	0.68	0.70	0.97	0.87	0.90	0.82	0.94	0.95	**0.91**	0.91	0.80	**0.86**	**0.87**	**0.89**	0.94	0.94	0.95	0.93
	FA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	sdFA	**0.81**	0.61	**0.81**	0.62	0.75	0.59	0.75	0.58	0.68	0.56	0.75	0.56	0.53	0.39	0.50	0.52	0.49	0.32	0.45	0.48
	isFA	0.76	**0.63**	0.60	**0.72**	**0.98**	**0.90**	**0.91**	**0.87**	**0.99**	**0.96**	0.89	**0.93**	**0.81**	0.74	0.73	0.80	**0.99**	**0.98**	**0.97**	**0.94**
K	aFA	0.87	**0.87**	**0.87**	**0.87**	0.87	0.87	0.87	**0.87**	0.87	0.87	0.87	0.87	0.87	**0.87**	**0.87**	**0.87**	0.87	0.87	0.87	0.87
	fsFA	**0.97**	0.33	0.43	0.77	**0.99**	**0.96**	**0.93**	0.85	**0.99**	**0.97**	**0.96**	0.91	**0.88**	0.79	0.82	0.70	**0.99**	**0.98**	**0.97**	**0.96**
	FA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	sdFA	0.81	0.61	0.81	0.62	0.75	0.59	0.75	0.58	0.68	0.56	0.75	0.56	0.53	0.39	0.50	0.52	0.49	0.32	0.45	0.48
	isFA	0.76	0.63	0.60	0.72	0.98	0.90	0.91	**0.87**	0.99	0.96	0.89	**0.93**	0.81	0.74	0.73	0.80	**0.99**	**0.98**	**0.97**	0.94
$K+4$	aFA	0.79	**0.90**	0.79	**0.79**	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	**0.79**	**0.79**	0.79	0.79	0.79	0.79	0.79
	fsFA	**0.91**	0.30	0.38	0.69	0.96	0.81	0.78	0.75	0.93	0.88	0.85	0.74	0.76	0.75	0.66	0.59	0.88	0.77	0.83	0.82
	FA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	sdFA	0.81	0.61	**0.81**	0.62	0.75	0.59	0.75	0.58	0.68	0.56	0.75	0.56	0.53	0.39	0.50	0.52	0.49	0.32	0.45	0.48
	isFA	0.76	0.63	0.60	0.72	**0.98**	**0.90**	**0.91**	**0.87**	**0.99**	**0.96**	**0.89**	**0.93**	**0.81**	0.74	0.73	**0.80**	**0.99**	**0.98**	**0.97**	**0.94**
2K	aFA	0.87	**0.90**	**0.87**	**0.87**	0.87	0.87	0.87	**0.87**	0.87	0.87	0.87	0.87	**0.87**	**0.87**	**0.87**	**0.87**	0.87	0.87	0.87	0.87
	fsFA	**0.89**	0.42	0.31	0.62	0.94	0.72	0.60	0.55	0.89	0.64	0.67	0.66	0.63	0.67	0.61	0.46	0.67	0.85	0.77	0.69
	FA	-	-	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-
	sdFA	0.81	0.61	0.81	0.62	0.75	0.59	0.75	0.58	0.68	0.56	0.75	0.56	0.53	0.39	0.50	0.52	0.49	0.32	0.45	0.48
	isFA	0.76	0.63	0.60	0.72	**0.98**	**0.90**	**0.91**	**0.87**	**0.99**	**0.96**	**0.89**	**0.93**	0.81	0.74	0.73	0.80	**0.99**	**0.98**	**0.97**	**0.94**
$3K$	aFA	**0.91**	**0.91**	**0.91**	**0.91**	0.91	**0.91**	**0.91**	**0.91**	0.91	0.91	**0.91**	0.91	**0.91**	**0.91**	**0.91**	**0.91**	0.91	0.91	0.91	0.91
	fsFA	0.80	0.43	0.39	0.46	0.84	0.54	0.52	0.54	0.94	0.69	0.61	0.54	0.77	0.57	0.62	0.47	0.52	0.75	0.57	0.58
	FA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	sdFA	0.81	0.61	0.81	0.62	0.75	0.59	0.75	0.58	0.68	0.56	0.75	0.56	0.53	0.39	0.50	0.52	0.49	0.32	0.45	0.48
	isFA	0.76	0.63	0.60	0.72	**0.98**	0.90	**0.91**	0.87	**0.99**	**0.96**	0.89	**0.93**	0.81	0.74	0.73	0.80	**0.99**	**0.98**	**0.97**	**0.94**

Table 5: Sparsity of the variants of factor analysis (FA) methods evaluated in Table 4; the data generating process is described in Section 5.1. We have evaluated all methods and configurations using 4 different values of the noise parameter σ^2 controlling the signal-to-noise ratio in the generated data: $\sigma^2 = \{0.01, 0.25, 1, 4\}$ displayed left-to-right in each of the 5 main columns. For aFA sparsity is computed as the portion of zeros out of all values of the $\mathbf{Z}$ matrix, whereas for all other methods the reported sparsity is the portion of zeros out of the $\mathbf{W}$ matrix; conventional FA does not induce sparsity (i.e. the reported sparsity value would be 0.00 in all reported scenarios).

References

Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

Christian F Beckmann and Stephen M Smith. Tensorial extensions of independent component analysis for multisubject fmri analysis. *Neuroimage*, 25(1):294–311, 2005.

Anirban Bhattacharya and David B Dunson. Sparse bayesian infinite factor models. *Biometrika*, pages 291–306, 2011.

Christopher Bingham. An antipodally symmetric distribution on the sphere. *The Annals of Statistics*, pages 1201–1225, 1974.

Tamara Broderick, Brian Kulis, and Michael Jordan. MAD-Bayes: MAP-based asymptotic derivations from Bayes. In *International Conference on Machine Learning*, pages 226–234, 2013.

Vince D Calhoun, Tulay Adali, Lars Kai Hansen, Jan Larsen, and James J Pekar. ICA of functional MRI data: an overview. In *in Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation*. Citeseer, 2003.

Vince D Calhoun, Jingyu Liu, and Tülay Adalı. A review of group ICA for fMRI data and ICA for joint inference of imaging, genetic, and ERP data. *Neuroimage*, 45(1):S163–S172, 2009.

Kieran R Campbell and Christopher Yau. Probabilistic modeling of bifurcations in single-cell gene expression data using a Bayesian mixture of factor analyzers. *Wellcome open research*, 2, 2017.

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

Carlos M Carvalho, Jeffrey Chang, Joseph E Lucas, Joseph R Nevins, Quanli Wang, and Mike West. High-dimensional sparse factor modeling: applications in gene expression genomics. *Journal of the American Statistical Association*, 103(484):1438–1456, 2008.

Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3): 287–314, 1994.

Zhenwen Dai, James Hensman, and Neil Lawrence. Spike and slab Gaussian process latent variable models. *arXiv preprint arXiv:1505.02434*, 2015.

Daniele Durante. A note on the multiplicative gamma process. *Statistics & Probability Letters*, 122:198–204, 2017.

Bradley Efron. Microarrays, empirical Bayes and the two-groups model. *Statistical Science*, 23(1):1–22, 2008.

Clément Elvira, Pierre Chainais, and Nicolas Dobigeon. Bayesian nonparametric principal component analysis. *arXiv preprint arXiv:1709.05667*, 2017.

Barbara E Engelhardt and Matthew Stephens. Analysis of population structure: a unifying framework and novel methods based on sparse factor analysis. *PLoS Genetics*, 6(9): e1001117, 2010.

Christopher J Fallaize and Theodore Kypraios. Exact Bayesian inference for the Bingham distribution. *Statistics and Computing*, 26(1-2):349–360, 2016.

Chuan Gao, Christopher D Brown, and Barbara E Engelhardt. A latent factor model with a mixture of sparse and dense factors to model gene expression data with confounding effects. *arXiv preprint arXiv:1310.4792*, 2013.

Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis*. CRC press, 2013.

Zoubin Ghahramani and Matthew J Beal. Variational inference for Bayesian mixtures of factor analysers. In *Advances in Neural Information Processing Systems*, pages 449–455, 2000.

Zoubin Ghahramani, Geoffrey E Hinton, et al. The EM algorithm for mixtures of factor analyzers. Technical report, CRG-TR-96-1, University of Toronto, 1996.

Zoubin Ghahramani, Thomas L Griffiths, and Peter Sollich. Bayesian nonparametric latent feature models. In *ISBA: 8th World Meeting on Bayesian Statistics*, 2007.

Harry H Harman. *Modern factor analysis.* University of Chicago Press, 1960.

Carl S Herz. Bessel functions of matrix argument. *Annals of Mathematics*, pages 474–523, 1955.

Nils Lid Hjort et al. Nonparametric Bayes estimators based on beta processes in models for life history data. *The Annals of Statistics*, 18(3):1259–1294, 1990.

Pedro A.d.F.R. Højen-Sørensen, Ole Winther, and Lars Kai Hansen. Analysis of functional neuroimages using ICA with adaptive binary sources. *Neurocomputing*, 49(1-4):213–225, 2002.

Ian T Jolliffe, Nickolay T Trendafilov, and Mudassir Uddin. A modified principal component technique based on the LASSO. *Journal of Computational and Graphical Statistics*, 12 (3):531–547, 2003.

CG Khatri and Kanti V Mardia. The von Mises–Fisher matrix distribution in orientation statistics. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1): 95–106, 1977.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

David Knowles and Zoubin Ghahramani. Infinite sparse factor analysis and infinite independent components analysis. In *International Conference on Independent Component Analysis and Signal Separation*, pages 381–388. Springer, 2007.

Sirio Legramanti, Daniele Durante, and David B Dunson. Bayesian cumulative shrinkage for infinite factorizations. *Biometrika*, 107(3):745–752, 2020.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: a continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Leland McInnes, John Healy, and James Melville. Umap: uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Martin J McKeown, Lars Kai Hansen, and Terrence J Sejnowsk. Independent component analysis of functional MRI: what is signal and what is noise? *Current Opinion in Neurobiology*, 13(5):620–629, 2003.

Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-100). 1996.

Akihiko Nishimura, David Dunson, and Jianfeng Lu. Discontinuous Hamiltonian Monte Carlo for discrete parameters and discontinuous likelihoods. *arXiv preprint arXiv:1705.08510*, 2017.

John Paisley and Lawrence Carin. Nonparametric factor analysis with beta process priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 777–784. ACM, 2009a.

John Paisley and Lawrence Carin. Nonparametric factor analysis with beta process priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 777–784, 2009b.

Lance Parsons, Ehtesham Haque, and Huan Liu. Subspace clustering for high dimensional data: a review. *ACM SIGKDD Explorations Newsletter*, 6(1):90–105, 2004.

Karl Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

Raimon HR Pruim, Maarten Mennes, Daan van Rooij, Alberto Llera, Jan K Buitelaar, and Christian F Beckmann. ICA-AROMA: A robust ICA-based strategy for removing motion artifacts from fMRI data. *Neuroimage*, 112:267–277, 2015.

Adrian E Raftery and Steven M Lewis. [Practical Markov chain Monte Carlo]: comment: one long run with diagnostics: implementation strategies for Markov chain Monte Carlo. *Statistical Science*, 7(4):493–497, 1992.

Piyush Rai and Hal Daumé. The infinite hierarchical factor regression model. In *Advances in Neural Information Processing Systems*, pages 1321–1328, 2009.

Petar P Raykov, James L Keidel, Jane Oakhill, and Chris M Bird. Activation of person knowledge in medial prefrontal cortex during the encoding of new lifelike events. *Cerebral Cortex*, 31(7):3494–3505, 04 2021. ISSN 1047-3211.

Yordan P Raykov, Alexis Boukouvalas, Max A Little, et al. Simple approximate MAP inference for Dirichlet processes mixtures. *Electronic Journal of Statistics*, 10(2):3548–3578, 2016.

Hemant D Tagare. Notes on optimization on Stiefel manifolds. In *Technical report, Technical report*. Yale University, 2011.

Jalil Taghia, Srikanth Ryali, Tianwen Chen, Kaustubh Supekar, Weidong Cai, and Vinod Menon. Bayesian switching factor analysis for estimating time-varying functional connectivity in fMRI. *Neuroimage*, 155:271–290, 2017.

Yee W Teh and Dilan Gorur. Indian buffet processes with power-law behavior. In *Advances in Neural Information Processing Systems*, pages 1838–1846, 2009.

Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

James Townsend, Niklas Koep, and Sebastian Weichwald. Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *The Journal of Machine Learning Research*, 17(1):4755–4759, 2016.

René Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2011.

Lianming Wang and David B Dunson. Fast Bayesian inference in Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 20(1):196–216, 2011.

Mike West. Bayesian factor regression models in the "large p, small n" paradigm. *Bayesian statistics*, 7:733–742, 2003.

Eric Zarahn, Geoffrey K Aguirre, and Mark D'Esposito. Empirical analyses of BOLD fMRI statistics. *NeuroImage*, 5(3):179–197, 1997.

George Kingsley Zipf. Selected studies of the principle of relative frequency in language. 1932.

Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.