**Latent Feature Sharing: An Adaptive Approach to Linear Decomposition Models**

Adam Farooq\(^1\), Yordan P. Raykov\(^1\)*, Petar Raykov\(^2\), Max A. Little\(^3,4\)

June 23, 2020

**Abstract**

Latent feature models are canonical tools for exploratory analysis in classical and modern multivariate statistics. Many high-dimensional data can be approximated using a union of low-dimensional subspaces or factors. The allocation of data points to these latent factors itself typically uncovers key relationships in the input and helps us represent hidden causes explaining the data. A widely adopted view is to model feature allocation with discrete latent variables, where each data point is associated with a binary vector indicating latent features possessed by this data point. In this work we revise some of the issues with existing parametric and Bayesian nonparametric processes for feature allocation modelling and propose a novel framework that can capture wider set of feature allocation distributions. This new framework allows for explicit control over the number of features used to express each point and enables a more flexible set of allocation distributions including feature allocations with different sparsity levels. We use this approach to derive a novel adaptive Factor analysis (aFA), as well as, an adaptive probabilistic principle component analysis (aPPCA) capable of flexible structure discovery and dimensionality reduction in a wide case of scenarios. Motivated by the often prohibitive slowness of feature allocation models, we derive both standard a Gibbs sampler, as well as, an expectation-maximization inference algorithms for aPPCA and aFA that converge orders of magnitude faster to a reasonable point estimate solution. We demonstrate that aFA can handle richer feature distributions, when compared to widely used sparse FA models and Bayesian nonparametric FA models. The utility of the proposed aPPCA model is demonstrated for standard PCA tasks such as feature learning, data visualization and data whitening. We show that aPPCA and aFA can infer interpretable high level features both when applied on raw MNIST and when applied for interpreting autoencoder features. We also demonstrate an application of the aPPCA to more robust blind source separation for functional magnetic resonance imaging (fMRI).

---

1 - Aston University; School of Engineering and Applied Sciences, Birmingham, United Kingdom
2 - University of Sussex; School of Psychology, Brighton, UK
3 - University of Birmingham; School of Computer Science, Birmingham, United Kingdom
4 - Media Lab, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America

* - Corresponding author

---

1 Introduction

Latent feature models provide principled and interpretable means for structure decomposition through leveraging specified relationships in the observed data. They are complementary to flexible continuous latent variable models (LVMs) or black-box autoencoder approaches which do not explicitly handle discreteness in the latent space and in fact can often be used in conjunction. A widely used application of latent feature models has a building block for Bayesian *sparse factor analysis* models \([1,2,3]\) which are backbone tools for dimensionality reduction and latent structure discovery in high dimensional data. In contrast, latent feature visualization counterparts have received a lot less attention despite the large popularity of empirical sparse principle component analysis techniques \([4,5]\). In this work, we propose a more flexible set of latent feature factor analysis models, based on adopting the multivariate hypergeometric distribution as feature allocation process. We fill the gap for latent feature visualization counterparts by...
proposing a fully Bayesian paradigm for specifying the probabilistic principal component analysis (PPCA) \[6\]. Markov Chain Monte Carlo (MCMC) inference scheme is proposed for these models, as well as, a novel scalable approximate expectation-maximization inference.

Linear dimensionality reduction methods are a mainstay of high-dimensional data analysis, due to their simple geometric interpretation and attractive computational properties. In linear Gaussian LVMs we assume the following generative model for the data:

\[
y = WX + \mu + \epsilon
\]

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

- The ubiquitous principal component analysis (PCA) \[7\] can be derived from Equation (1) and further making the assumptions that \( \mu = 0 \), vectors of \( W \) are orthogonal and the variance of the isotropic noise is \( 0 \), i.e. assume \( \epsilon \sim N(0, \sigma^2 I_D) \) and \( \sigma \rightarrow 0 \).
- If we avoid the small variance asymptotic step from above, but still assume \( W \) has orthogonal vectors and Gaussian noise \( \epsilon \sim N(0, \sigma^2 I_D) \), we recover probabilistic PCA (PPCA) \[8\].
- In the case where we omit the orthogonality assumption on \( W \) and assume more flexible elliptical noise \( \epsilon \sim N(0, \text{diag}(\sigma)) \) with \( \sigma = (\sigma_1, \ldots, \sigma_D) \), we obtain the classic factor analysis (FA) \[9\].
- Variants of independent component analysis \[10\] can be obtained by assuming flexible elliptical noise \( \epsilon \sim N(0, \text{diag}(\sigma)) \) with \( \sigma = (\sigma_1, \ldots, \sigma_D) \), but also assuming a non-Gaussian distribution model for the latent variables \( x \in \mathbb{R}^K \); for example Laplace distribution model \[2\].

A widely accepted challenge shared in all of these linear Gaussian techniques is that the columns of \( W \) (i.e. the principal components or factors) are a linear combination of all the original variables. This problem also persists for more flexible continuous latent variable models \[11\] and often makes it difficult to interpret the results. To handle these issues, there has been a plethora of prior work on developing sparse PCA \[4\] and sparse FA models \[1\]. Zou et al. \[4\] places a least absolute shrinkage and selection operator (LASSO) regularization on columns of \( x \), which leads to more interpretable components, compared to simple thresholding. Similar models have been achieved with a fully Bayesian approach of placing relevance determination priors \[12\]. West \[1\] have further suggested placing a two-component mixture model over the loadings \( x \) that allow to switch on and off factors from \( W \) imposing natural dimensionality reduction. In this scenario, the probabilities of factors having non-zero loadings are independent across all points. The sparse PCA \[4\] can be used to produce modified principal components with sparse loadings, however, falls short in scenarios where we wish to also model the factor sharing among specific subsets of the input data.

In such cases explicit modelling of partitions in the high dimensional input space can be achieved via augmenting the latent space with additional of discrete latent variables \[1\]. In latent feature models, we denote these latent variables with binary vectors \( z \in \mathbb{R}^K \) which indicate all the features associated with that point. This approach allows to capture flexible cluster topology and also can account for overlapping factors and mixed group membership. This is in contrast to latent class LVMs which are designed for subspace clustering \[13,14\] or mixture of factor analyzers \[15,16,12\]. The challenge is in designing a sufficiently flexible and intuitive model of the latent feature space. Several nonparametric FA models have addressed this using the Beta processes \[3\], or their marginal Indian buffet processes \[2,17\] (IBP) which have infinite capacity and can be used to infer the feature space dimensionality (i.e. number of features). However, the IBP imposes some explicit sparsity constraints on the feature allocation distribution which can lead to producing non-interpretable spurious features and overestimation of the underlying number of features \[18\].

The multivariate hypergeometric model we propose here allows for intuitive control over the sparsity of the feature allocation matrix. We show that the parameters of the hypergeometric prior allow for control over the expected sharing, while the IBP assumes log-linear growth of the number of factors \[19\] and decaying factor representation \[20\]. The proposed model is parametric since it fixes the number of unique features instantiated, but at the same time has a different parameter controlling the number of unique features used to represent each data point. This is a vital point since it allows us to naturally separate (1) features which explain large variance percentage for a small subset of the data from (2) spurious features which explain small variance percentage for a potentially larger subset of the data. This formulation is natural in the context of data visualization and dimensionality reduction, where natural constraints on the feature representation for each data point occur – in visualization normally points are reduced to two or three dimensions; in dimensionality reduction, we model each point with \( K << D \) dimensions.
Figure 1: Graphical model for the proposed adaptive factor analysis (aFA) and the adaptive probabilistic principal component analysis (aPPCA) models as well as isFA and iICA.

2 Preliminaries

2.1 Latent feature factor analysis models

In latent feature linear Gaussian LVMs, we augment the model from Equation (1) can assume and write the following constriction in matrix notation for $N$ $D$-dimensional observations:

$$Y = WX \odot Z + E \tag{2}$$

where $Y = [y_1, \ldots, y_N]$ is the observation matrix, $W$ is a $(D \times K)$ factor (or mixing) matrix, $Z = [z_1, \ldots, z_N]^T$ 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, $E = [\epsilon_1, \ldots, \epsilon_N]$ is a noise matrix consisting of $N$ independent and identically distributed $D$-dimensional zero-mean vectors drawn from $\mathcal{N}(0, \sigma I_D)$; finally $X = [x_1, \ldots, x_N]^T$ are the latent variables where each point $x_{k,n}$ is assumed Gaussian for FA and PCA models, and Laplace distributed for Bayesian independent component analysis models. The graphical model is depicted in Figure 1.

2.2 Inference

The joint likelihood of the model (see Figure 1), generally can be written as:

$$P(Y, W, X, Z|\theta) = \prod_{n=1}^{N} \left( \prod_{k=1}^{K} P(y_n|W, x_n, z_{k,n}, \sigma) \prod_{k=1}^{K} P(x_{k,n}) P(z_{k,n}|\alpha) \right) \times \prod_{k=1}^{K} P(w_k|\sigma_W) \tag{3}$$

where we use $\theta$ to denote jointly the hyperparameters. For the infinite sparse FA (isFA) model Knowles and Ghahramani [2], we assume a Gaussian prior on the factor matrix $W$ and IBP prior on $Z$ which results in $\theta = \{\alpha, \sigma, \sigma_W\}$ where $\alpha$ is the concentration parameter for the IBP, $\sigma^2$ is the variance of the observed data and $\sigma^2_W$ is the variance of the factors. We will only briefly summarize a straightforward Gibbs sampler for this isFA model. Paisley and Carin [3] proposed a scalable variational inference algorithm for estimating this model.

The posterior distribution over the latent variables $x_{k,n}$ for which its respective $z_{k,n} = 1$ is sampled from a Gaussian:

$$P(x_{k,n} | \ldots) = \mathcal{N} \left( x_{k,n} \mid \frac{w_k^T \epsilon_{-k,n}}{\sigma^2 + w_k^T w}, \frac{\sigma^2}{\sigma^2 + w_k^T w} \right) \tag{4}$$

where we have omitted all the variables upon which $x_{k,n}$ depends on, $w_k$ is the $k$-th column of the matrix $W$ and $\epsilon_{-k,n}$ is $(y_n - W(x_n \odot z_n))$ with $z_{k,n} = 0$, or the noise associated with $n$-th point and $k$-th feature.
The posterior distribution over the $k$-th factor loading $w_k$ is a $D$-dimensional multivariate Gaussian:

$$
P(w_k | \ldots) = \mathcal{N}
\left(w_k \left| \frac{\sigma_w^2}{\sum_w \sigma_w^2 + \sigma_k^2} X_k X_k^T \left( \frac{1}{\sigma_k^2} + \frac{1}{\sigma_w^2} \right), I_D \right) \right)
$$

(5)

where $X_k$ is the $k$-th column of the matrix $X$ and $E_{-k}$ is $(Y - XW)$ with $w_k = 0$.

The matrix $Z$ is sampled in two steps: the first involves sampling existing features and the second, sampling new features. The latent variables $z_{k,n}$ are marginalized out since the collapsed Gibbs sampler can lead to faster convergence [21]; the marginal distribution is available in closed form as the Gaussian prior over the hidden sources is conjugate to the Gaussian likelihood over the observed data. The existing features $z_{k,n}$ can be sampled directly using the Bernoulli posterior:

$$
P(z_{k,n} | \ldots) = \text{Bern}
\left(\frac{P(y_n | z_{k,n} = 1) P(z_{k,n} = 1 | z_{k,-n})}{P(y_n | z_{k,n} = 1) P(z_{k,n} = 1 | z_{k,-n}) + P(y_n | z_{k,n} = 0) P(z_{k,n} = 0 | z_{k,-n})}\right)
$$

(6)

In the described setup, the posterior for new features is not available in closed form, but it can be approximated using a Metropolis-Hastings step. For each observation, adding $\kappa$ number of new features and their corresponding parameters (columns of matrix $W$) are jointly proposed and accepted with probability proportional to likelihood improvement brought about by these new features.

### 3 Adaptive factor analysis

The latent feature FA models described above assume a product of independent Bernoulli distributions to represent the feature allocation process, with a shared Beta distribution or Beta process prior. This modelling choice leads to intrinsic sparsity assumption about the allocation indicators $Z$: log-linear growth of the number of factors [19] and decaying factor representation [20]. The result is that the underlying IBP can lead to producing non-interpretable spurious features and inconsistent overestimation of the number of inferred factors [18].

To address some of these issues here we propose a constrained allocation model for $Z$ which relies on the multivariate hypergeometric distribution. The hypergeometric distribution describes the probability of $L$ successes in $K$ draws, **without replacement**. Under the Beta-Bernoulli feature allocation model, features are used with independent probabilities, modelled with replacement and popular features become more common because of the reinforcement effect. In contrast, we will see that FA model with hypergeometric feature allocations, can capture a wider set of allocation distributions, while also very efficient to train.

Motivated by the huge computational costs involved with training latent feature FA models, we derive a scalable expectation-maximization (EM) algorithm for approximate inference in adaptive FA.

#### 3.1 Feature allocation models with replacement

To implement the proposed aFA, we place a **multivariate hypergeometric distribution** [22] as a prior over all the latent feature indicators $Z$:

$$
P(Z | K, L, m_1, \ldots, K_1 = 1) = \prod_{n=1}^{N} \frac{\binom{1}{z_{1,n}} \binom{1}{z_{2,n}} \times \cdots \times \binom{1}{z_{K,n}}}{\binom{K}{L}}
$$

(7)

for given hyperparameter values of $L$ and $K$, such that $L < K$. For Equation (7), $z_{k,n} \in \{0, 1\}$ and we have $\sum_{k=1}^{K} z_{k,n} = L$. The parameter $L$ allows for explicit control over the number of latent factors used to decompose each observation. $K$ denotes the number of unique factors used to represent the data, i.e. the number of columns in $W$. This implies that each input data point is associated with different subset of $L$ factors, selected from a total of total of $K$ unique factors. $K$ accounts for the global sharing of structure across overlapping groups of data points with common factors; if $K$ is large enough, each point can, in principle, be associated with non-overlapping subsets of $L$ factors, equivalent to mixture of FA model. But, as $K$ reduces, more of these factors 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 $K$. This allows us to interpret $L$ as the local capacity of the model and $K$ controls global capacity or sharing. If $L = K$ we recover classical FA models, since all features are associated with all observed data points. As $K - L$ increases, more local structure of the data can be captured.
3.2 Scalable inference for aFA

The joint likelihood for the proposed model takes the same form as the FA from Equation (3) with changed distribution on Z. The parametric nature of the hypergeometric model allows us to write an efficient EM algorithm for training the aFA which can be used both for initialization of a full Gibbs sampler or for obtaining quick maximum-a-posteriori solution for the model. We marginalize over the continuous latent variables X and at each iteration we compute the expectation of the likelihood with respect to X: \( \mathbb{E}_X \{ \log P(\mathbf{Y}, \mathbf{Z}, \mathbf{W}, \mathbf{X}, \mathbf{Z} | \mathbf{Y}, \mathbf{X}, \mathbf{W}, \sigma, \sigma_x) \} \). The log-likelihood can be expressed as:

\[
L_N = -\sum_{n=1}^{N} \left( \frac{K}{2} \ln (\sigma_x^2) + \frac{D}{2} \ln (\sigma) \right) + \frac{1}{\sigma^2} x_n^T x_n + \frac{1}{2\sigma^2} y_n^T y_n - \frac{1}{\sigma^2} x_n^T A_n^T W^T y_n + \frac{1}{2\sigma^2} x_n^T A_n^T W^T W A_n x_n
\]

where \( A_n \) is a \((K \times K)\) matrix with the diagonal elements being \( z_n \). The expectation of \( x_n \) from above can then be written as:

\[
\mathbb{E} [x_n] = (\sigma_x^{-2} I_K + \sigma^{-2} A_n^T W^T W A_n)^{-1} (\sigma^{-2} A_n^T W^T y_n)
\]

Using \( \mathbb{E} [x_n] \), we can rewrite the marginal log-likelihood after integrating \( x_n \):

\[
L_N = -\sum_{n=1}^{N} \left( \frac{K}{2} \ln (\sigma_x^2) + \frac{D}{2} \ln (\sigma) \right) + \frac{1}{\sigma^2} \text{tr} \left( \mathbb{E} [x_n x_n^T] \right) + \frac{1}{2\sigma^2} y_n^T y_n - \frac{1}{\sigma^2} \mathbb{E} [x_n^T A_n^T W^T y_n]
+ \frac{1}{2\sigma^2} \text{tr} \left( A_n^T W^T W A_n \mathbb{E} [x_n x_n^T] \right)
\]

In the EM maximization step, we update the rest of the parameters and the indicator variables by solving \( \partial L_N / \partial \mathbf{W} \), \( \partial L_N / \partial \sigma \), \( \partial L_N / \partial \sigma_x \) and \( \partial L_N / \partial z_n = 0 \).

Since we are often interested only in a point estimate for the indicator variables \( Z \), iterative optimization via coordinate descend can lead to robust MAP estimatems i.e. \( Z^{MAP} \). The complete EM algorithm for the proposed aFA is summarized in Algorithm 1. Typically, it converges in only a few iterations and later we show its MAP decomposition leads to comparable reconstruction error to a Gibbs trained aFA. The EM algorithm for aFA will also lead to lower reconstruction error compared to other well known parametric and nonparametric FA algorithms.

4 The adaptive probabilistic principal component analysis (aPPCA) model

In this section we propose novel adaptive probabilistic principal component analysis (aPPCA) framework which models linear subspace sharing across overlapping subsets of observations. The aPPCA can be described as a latent feature approach, in which the latent features are assumed to share orthogonal one-dimensional subspaces, characterized via the projection vectors \( w_1, w_2, ..., w_K \) forming \( W \). If two points \( y_i \) and \( y_j \) are associated with a projection vector \( w_k \), it means that sufficient information about these points can be preserved by projecting them in the direction specified by \( w_k \).

We study two variants of aPPCA, which differ in the way they model the latent subspace allocation. First, we derive a direct extension of the nonparametric FA model from [2] to the PPCA setup in which the columns of the transformation matrix \( W \) are orthogonal. Second, also propose a parametric hypergeometric variant of the aPPCA which allows for explicit control over both the number of unique columns \( K \) in \( W \), as well as the observation specific number of active vectors \( L \).

Both these proposed aPPCA models share the following construction:

\[
\begin{align*}
\mathbf{y}_n & = \mathbf{W} (\mathbf{x}_n \odot \mathbf{z}_n) + \mu + \epsilon_n \\
\mathbf{x}_n & \sim \mathcal{N}(0, I_K) \\
\epsilon_n & \sim \mathcal{N}(0, \sigma^2 I_D)
\end{align*}
\]


We can check whether the MCMC sampler has converged using standard tests such as Raftery and Lewis [30] directly.

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

Input: $Y, \Theta, \text{MaxIter}$

Initialise: Sample a random $(K \times N)$ binary matrix $Z$ and initialize $\{W, X\}$ using PCA

for iter ← 1 to MaxIter

for $n ← 1$ to $N$

Set $I = \{k : z_{k,n} = 1\}$

for $l ← 1$ to $L$

Set $z_{l,l,n} = 0$

Sample $I$ using [19]

for $n ← 1$ to $N$

Set $x_n = (\sigma_1^{-2}I_K + \sigma_2^{-2}A(n)W^TA(n))^{-1} (\sigma_2^{-2}A(n)W^T y_n)$

Set $\Psi_n = (\sigma_1^{-2}I_K + \sigma_2^{-2}A(n)W^TA(n))^{-1} + x_n x_n^T$

Set $W = \left(\sum_{n=1}^N Y_n (A(n)x_n)^T \right) \left(\sum_{n=1}^N A(n)\Psi_n A(n)\right)^{-1}$

Set $\sigma^2 = \frac{1}{N} \sum_{n=1}^N (y_n y_n - 2x_n^A(n)W^T y_n + \text{trace} \left(A(n)W^TWA(n)\Psi_n\right))$

Set $\sigma_z^2 = \frac{1}{N} \sum_{n=1}^N \text{trace} (\Psi_n)\right)$

for $n = 1, \ldots, N$, where $y_n \in \mathbb{R}^D$ is the $D$-dimensional observed data; $x_n \in \mathbb{R}^K$ is the lower dimensional latent variable; $W = [w_1, w_2, \ldots, w_K]$ is an unobserved $(D \times K)$ projection matrix with $w_i \perp w_j$ for all $i \neq j$; $z_n \in \mathbb{R}^R$ is a binary vector indicating the active subspaces for point $n$, $\epsilon_n$ is zero-mean Gaussian noise; and without loss of generality we assume the $D$-dimensional mean vector $\mu = \frac{1}{N} \sum_{n=1}^N y_n$ is zero. Under both of the proposed aPPCA models, we can write the likelihood of point $n$ as:

$$P(y_n | W, x_n, z_n, \sigma) = \frac{1}{(2\pi\sigma^2)^{\frac{D}{2}}} \exp \left(-\frac{1}{2\sigma^2} (y_n - W(x_n \odot z_n))^T (y_n - W(x_n \odot z_n)) \right) \quad (12)$$

4.1 Inference

Computing the posterior distribution of the latent variables $\{X, Z\}$ and the projection matrix $W$ is analytically intractable and we have to resort to approximate inference. The main difference for the aPPCA model here, when compared to the aFA model above, is that the new distribution of the orthonormal matrix $W$ does not allow for closed form updates. Numerically optimizing over $W$ and marginalizing $X$ leads to slow mixing and EM scheme leads to poor local solutions for this model. An efficient Markov Chain Monte Carlo (MCMC) scheme [26] can be derived which iterates between explicit updates for $W, z_n, x_n$, and the hyperparameters we wish to infer, i.e. $\sigma^2$ and $\alpha$ (update of $\sigma^2$ and $\alpha$ is in given in Appendix B). Sampling from directional posteriors is prohibitively slow, so we propose MAP update scheme for the updates on $W$. Alternatively we could use an automated MCMC platforms such as STAN [27] for the inference, but STAN does not deal well with discontinuous likelihood models such as aPPCA. This can be addressed using discrete relaxations such as Maddison et al. [28] or numerical solver extensions such as Nishimura et al. [29]. However, such an approach can be justified only for nonlinear intractable extensions of aPPCA, since the Gibbs sampler with closed form updates is substantially more efficient, hence we derive it here.

The joint data likelihood of both aPPCA models we propose takes the form:

$$P(Y, W, X, Z | \sigma, \alpha) = \prod_{n=1}^N \left( P(Y_n | W, x_n, z_n, \sigma) \prod_{k=1}^K P(x_{k,n} | \sigma) P(z_{k,n} | \alpha) \right) \quad (13)$$

We can check whether the MCMC sampler has converged using standard tests such as Raftery and Lewis [30] directly on Equation [15]. Comparing the two aPPCA models from Section 4, the only difference is in $P(Z)$. We will see that this will affect the posterior update of $W$, but the rest of inference algorithm is otherwise identical across both models.

Posterior of $W$

In order to comply with the orthogonality constraint on $W$, i.e. $w_i \perp w_j \forall i \neq j$, we have to use a distribution with support on the Stiefel manifold (see [31] for a good introduction). One option explored in Elvira et al. [19] would
be using conjugate Bingham prior \[^{32}\] independently on the columns of \(W\) leading to independent von Mises-Fisher posterior over each column where re-scaling is required after each sample to maintain orthogonality. Empirical trails suggest that this results in very poor mixing. To overcome this issue, we propose joint sampling of the columns of \(W\). We place a uniform prior over the Steifel manifold on the matrix \(W\) which allows us to work with a matrix von Mises-Fisher \[^{33}\] posterior:

\[
P (W | Y, X, Z, \sigma) = {}_0F_1^{-1} \left( \emptyset, \frac{D}{2}, AA^T \right) \exp \left( \text{tr} (AW) \right)
\]  

(14)

where \(A = \frac{1}{\sigma^2} (X \odot Z) Y^T\) and \(_0F_1^{-1} (\cdot)\) is a hypergeometric function \[^{34}\]. The normalization 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 et al. \[^{35}\] proposed a Metropolis-Hastings scheme to generate samples from Equation (14), the resulting posterior over \(W\) converges faster than the Bingham-von-Mises-Fisher posterior, but can be further sped up by numerical optimization methods. Here, we propose updating the matrix \(W\) by maximizing the posterior from Equation (14) over the Steifel manifold, i.e. keeping the orthogonality assumption \(w_i \perp w_j \; \forall i \neq j\). An efficient implementation can be achieved using the PYMANOPT toolbox \[^{36}\], for optimization over manifolds with different geometries; this step is outlined in Appendix C.

**Posterior of \(X\)**

The posterior distribution over the latent variable \(x_{k,n}\), for which its respective \(z_{k,n} = 1\), is sampled from a Gaussian:

\[
P (x_{k,n} | w_k, y_n, z_n) = N \left( x_{k,n}, \frac{y_n^T w_k}{\sigma^2 + 1}, \frac{\sigma^2}{\sigma^2 + 1} \right)
\]  

(15)

where \(w_k\) is the \(k\)th column of the matrix \(W\).

### 4.1.1 Bayesian nonparametric aPPCA

If each component \(z_{k,n}\) from the binary vectors \(z_n\) is independently drawn from a Bernoulli distribution with the \(K\) mixing parameters \(\{p_k\}_{k=1, \ldots, K}\), each independently drawn from a Beta distribution, then as the number of latent features \(K \to \infty\), one can show that the conjugate prior over the matrix \(Z^T = [z_1, \ldots, z_N]^T\) is the Beta process \[^{37}\]. By integrating out the mixing parameters, one is able to work with a simpler IBP marginal process. Under the IBP prior, the indicator matrix \(Z\) is \((K \times N)\)-dimensional 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 \(K\) follows a Poisson distribution with mean \(\alpha \sum_{n=1}^{N} \frac{1}{N}\); for large \(N\), \(K \approx \alpha \ln (N)\). The prior for the matrix \(Z^T\) under the IBP is:

\[
P (Z^T | \alpha) \propto \exp (-\alpha H_N) \alpha^K \left( \prod_{k=1}^{K} \frac{(m_k-1)!(N-m_k)!}{N!} \right)
\]  

(16)

where \(H_N = \sum_{n=1}^{N} \frac{1}{n}\) and \(m_k = \sum_{n=1}^{N} z_{k,n}\).

In the Bayesian nonparametric aPPCA, we placed an IBP prior over the indicator matrix \(Z\); this assumes that after a finite \(N\) number of observations only a finite \(K\) number of one-dimensional subspaces are active. This results in the first \(K\) rows of \(Z\) having non-zero entries, and the remaining being all zeros. By design, \(K\) cannot exceed the dimension of the data \(D\) and this leads to truncation of the IBP such that \(K\) has an upper limit of \(K^{\max}\); where \(K \leq K^{\max} \leq D\), therefore in Bayesian nonparametric aPPCA, \(Z\) 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. Following Knowles and Ghahramani \[^{2}\], we sample the matrix \(Z\) in two stages which include sampling “existing features” and “new features”; in both cases
Algorithm 2 Pseudocode for inference in Bayesian nonparametric aPPCA using Gibbs sampling.

**Input:** $Y$, $\Theta$, MaxIter, $K$

**Initialise:** Sample a random $(K_{\text{max}} \times N)$ binary matrix $Z$ and initialize $W$ using PCA

**for** iter ← 1 to MaxIter

**for** $n$ ← 1 to $N$

**for** $k$ ← 1 to $K$

Sample $z_{k,n}$ using (17)

Sample $\kappa \sim \text{Poisson} \left( \frac{\tau}{N} \right)$

Accept $\kappa$ new features with probability (18) and update $K$ accordingly

**for** $n$ ← 1 to $N$

**for** $k$ ← 1 to $K$

if $z_{k,n} = 1$

Sample $x_{k,n}$ using (15)

Sample $W$ using (14)

Sample $\{ \sigma^2, \alpha \}$ from Appendix B

the latent variables $x_{k,n}$ are marginalized out. The posterior distribution over the existing features $z_{k,n}$ is Bernoulli distributed:

$$P(z_{k,n} | \ldots) = \text{Bern} \left( \frac{m_{k,-n} \cdot \exp \left( \frac{1}{2\sigma^2(\sigma^2+1)} \left( y_n^T w_k \right) \left( \frac{\sigma^2}{\sigma^2+1} \right)^2 \right)}{m_{k,n} \cdot \exp \left( \frac{1}{2\sigma^2(\sigma^2+1)} \left( y_n^T w_k \right) \left( \frac{\sigma^2}{\sigma^2+1} \right)^2 + 1 \right)} \right) \left( y_n^T w_k \right) \left( \frac{\sigma^2}{\sigma^2+1} \right)^{\frac{3}{2} + 1}$$

where we omit the dependence on $W$ and $\sigma^2$ and $m_{k,-n} = \sum_{i \neq n} z_{k,i}$.

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

$$P(z_{K+j,n} | \ldots) = \text{Bern} \left( \frac{\exp \left( \frac{1}{2\sigma^2(\sigma^2+1)} \sum_{k=K+1}^{K+\kappa} \left( y_n^T w_k \right)^2 \right) \left( \frac{\sigma^2}{\sigma^2+1} \right)^{\frac{3}{2} + 1}}{\exp \left( \frac{1}{2\sigma^2(\sigma^2+1)} \sum_{k=K+1}^{K+\kappa} \left( y_n^T w_k \right)^2 \right) \left( \frac{\sigma^2}{\sigma^2+1} \right)^{\frac{3}{2} + 1}} \right)$$

for $j = 1, \ldots, \kappa$ new features.

### 4.1.2 Parametric aPPCA

In many common PPCA applications, constraints on the latent feature dimensionality occur naturally. In 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 input features, the output dimensionality is usually fixed to $D$ (the dimensionality of the input). In these scenarios the multivariate hypergeometric model for $Z$ allows explicit control over the number of latent subspaces $L$ used to decompose each single observation. $K$ 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 different subset of $L$ subspaces, selected from a total of $K$ subspaces. So, any single point is actually represented by lower dimensional spaces subsets of $\mathbb{R}^L$. Note that the orthogonality assumption $w_i \perp w_j \forall i \neq j$ for the columns of $W$ implies that $K \leq D$.

In the parametric aPPCA we place a flexible multivariate hypergeometric prior on the latent $Z$. The hypergeometric prior allows updates of $Z$ across $N$ in parallel, since the number of observed data points assigned to a latent subspace no longer implies higher probability of assigning a new data point to that subspace, i.e. no reinforcement effect. Instead,
We generated synthetic data according to the model $Y$ which has each column drawn from a multivariate Gaussian with mean zero and covariance matrix $\sigma^2I_K$; $\sigma^2$ is scalar parameter set to 1 and $E$ is a $(D \times 1000)$ noise matrix which has each column drawn from a multivariate Gaussian with mean zero and covariance matrix $\sigma^2I_D$; $\sigma$ is scalar parameter set to 0.1. We compare the mean absolute reconstruction error of the following models:

- Factor analysis (FA): $Z$ is all 1’s
- Infinite sparse FA (isFA): $Z$ is modelled with an IBP prior, see (16)
- Finite sparse FA (fsFA): Same as the isFA, except $K$ is fixed.
- Adaptive FA (aFA): $Z$ is modelled with a multivariate hypergeometric prior (7).

To ensure fairness we generate multiple datasets, each with different priors over the binary matrix $Z$. Figure 2 shows the different forms of the matrix $Z$ used in the study; where Figure 2(a) was generated using a sparse feature allocation

Algorithm 3 Pseudocode for inference in parametric aPPCA using Gibbs sampling.

**Input:** $Y$, $\Theta$, MaxIter
**Initialise:** Sample a random $(K \times N)$ binary matrix $Z$ and initialize $W$ using PCA

for iter ← 1 to MaxIter
  for $n$ ← 1 to $N$
    Set $I = \{k : z_{k,n} = 1\}$
    for $l$ ← 1 to $L$
      Set $z_{I,l,n} = 0$
      Sample $I_l$ using (19)
    for $n$ ← 1 to $N$
      for $k$ ← 1 to $K$
        if $z_{k,n} = 1$
          Sample $x_{k,n}$ using (15)
    Sample $W$ using (14)
    Sample $\{\sigma^2, \alpha\}$ using Appendix B

for each $n = 1, \ldots, N$, we sample $z_n$ by first finding the $L$ observed data indices $\{l_1, \ldots, l_L\}$ for which $z_n$ is one, then for each $l_i$, we set $z_{n,l_i} = 0$ and sample $l_i$ from the following Categorical distribution:

$$l_i \sim \text{Categorical}\left(\frac{(1 - z_{1,n}) \exp \left((y_n^T w_1)^2\right)}{\sum_k (1 - z_{k,n}) \exp \left((y_n^T w_k)^2\right)}, \ldots, \frac{(1 - z_{K,n}) \exp \left((y_n^T w_K)^2\right)}{\sum_k (1 - z_{k,n}) \exp \left((y_n^T w_K)^2\right)}\right)$$

(19)

where after each draw we set $z_{n,l_i} = 1$. In dimensionality reduction applications we often assume $L$ being two or three, hence $l_1$ 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 the parametric aPPCA is suggested in Algorithm 3.

4.2 Relationship to PCA

If we marginalize the likelihood (from Equation (12)) with respect to the discrete and continuous latent variables $\{x_n, z_n\}$ and take the limit $\alpha \to 0$, the maximum likelihood solution with respect to the transformation matrix $W$ is a scaled version of the $K$ largest eigenvectors of the covariance matrix (like PCA) of the data (multiplied by orthonormal rotation); proof of this can be seen in Appendix A. Furthermore, different priors over the matrix $Z$ result in different variants of the model, giving explicit control over the scale of the different projection axis.

5 Experiments

5.1 Data visualization

Adaptive FA

Synthetic data

We generated synthetic data according to the model $Y = W (X \odot Z) + E$, where $X$ is a $(K \times 1000)$ matrix which has each column drawn from a multivariate Gaussian with mean zero and covariance matrix $I_K$, $Z$ is a random $(K \times 1000)$ binary matrix of five different distributions, the matrix $W$ is $(D \times K)$ with columns drawn from a multivariate Gaussian with mean zero and covariance matrix $\sigma^2 I_K$; $\sigma W$ is scalar parameter set to 1 and $E$ is a $(D \times 1000)$ noise matrix which has each column drawn from a multivariate Gaussian with mean zero and covariance matrix $\sigma^2 I_D$; $\sigma$ is scalar parameter set to 0.1. We compare the mean absolute reconstruction error of the following models:

- Factor analysis (FA): $Z$ is all 1’s
- Infinite sparse FA (isFA): $Z$ is modelled with an IBP prior, see (16)
- Finite sparse FA (fsFA): Same as the isFA, except $K$ is fixed.
- Adaptive FA (aFA): $Z$ is modelled with a multivariate hypergeometric prior (7).

To ensure fairness we generate multiple datasets, each with different priors over the binary matrix $Z$. Figure 2 shows the different forms of the matrix $Z$ used in the study; where Figure 2(a) was generated using a sparse feature allocation.
Figure 2: Different matrices Z used in the synthetic study from Table 1 where black indicates 1 and white indicates 0, (a) Sparse binary matrix, (b) Dense binary matrix, (c) Subspace clustered binary matrix, (d) Sampled using multivariate hypergeometric distribution and (e) Full of ones.

Table 1: Mean absolute reconstruction error of various models over different synthetic data generated using different priors over the \((K \times 1000)\) binary matrix \(Z\). The data used took the form \(Y = W(X \odot Z) + E\), where \(Y\) is a \((D \times 1000)\) observation matrix, \(X\) is a \((K \times 1000)\) latent feature matrix, \(W\) is a \((D \times K)\) factor loading matrix and \(E\) is a \((D \times 1000)\) noise matrix.

| Prior                          | Sparse | Dense | Multivariate hypergeometric | Subspace clustering | Ones |
|-------------------------------|--------|-------|-----------------------------|---------------------|------|
|                               |        |       |                             |                     |      |
| Factor analysis (FA)          | 0.012  | 0.014 | 0.012                       | 0.088               | 0.012 0.015 0.012 0.014 |
| Finite sparse FA              | 0.014  | 0.018 | 0.015                       | 0.090               | 0.016 0.018 0.014 0.019 |
| Infinite sparse FA            | 0.023  | 0.042 | 0.021                       | 0.073               | 0.024 0.046 0.045 0.075 |
| Adaptive FA (aFA) Gibbs       | 0.014  | 0.019 | 0.012                       | 0.047               | 0.013 0.021 0.015 0.019 |
| Adaptive FA (aFA) EM          | **0.011** | **0.013** | **0.011**                   | **0.034**           | **0.012 0.014 0.012 0.014** |

prior, Figure 2(b) was also generated using a dense feature allocation prior, Figure 2(c) was generated assuming subspace clustering (i.e. no overlap in the sets of features describing each subspace); Figure 2(d) was generated using a multivariate hypergeometric distribution and finally Figure 2(e) is simply full of ones. We expect each of the models discussed above to do well on their respective \(Z\) matrix, for example FA should do relatively well when \(Z\) is like Figure 2(e) than if \(Z\) is like Figure 2(a).

The results in Table 1 suggest that aFA (EM) outperforms the other variants of FA, we also saw that the FA does as well as the aFA (EM) when the matrix \(Z\) is full of ones, this is because the FA assumes the matrix \(Z\) is full of ones whereas aFA (EM) is able to adaptively learn them; slightly lower reconstruction of the EM compared to Gibbs aFA is most likely due to convergence issues of the Gibbs sampler.

**MNIST data**

We apply adaptive FA (aFA) on 2500 digits from the MNIST digit dataset [38]. In Figure 4, we show the factor sharings across the odd digits which are calculated based on the proportion of factors shared between all digit pairs. The larger
Figure 3: Estimated latent feature allocation, for sparse (left) and dense (right) latent feature Gaussian synthetic data. The x-axis denotes the proportion of points associated with a feature and the y-axis is the feature number (indicator). The red line shows the true feature distribution. The remaining lines contrast the ability to recover the generating distribution using the nonparametric isFA and the proposed aFA models.

and darker circles indicate more sharing. As expected, observations depicting the same digits have the most shared factors; 1’s and 7’s also share significant structure as well as 5’s and 9’s. The number of unique factors $K$, as well as, the number $L$ of observation factors explicitly control the sparsity of the factors. This implies that aFA can infer both sparse factor matrices as in [39], as well as dense ones. In Figure 4, we display the factor proportions for the model trained on MNIST.

Figure 4: Left: Factor sharing grid between digits: circles are sized depending on the number of features shared between digit pairs denoted on the x-axis and y-axis; color enforces this effect where darker circles indicate more sharing and brighter circles - less. Right: Distribution of feature allocation process: y-axis denotes the proportion of data sharing current factor; x-axis indicates the factor number where they have been ordered based on most popular (left), least popular with small number of data allocated (right).

Adaptive PPCA

We now apply the parametric aPPCA for data visualization using 10,000 samples from the MNIST digit dataset [38]. We use a simple (with 2 MLP layers) variational autoencoder (VAE) [40] to reduce the dimension of the digits from 784-dimensional to 10-dimensional, then we apply aPPCA with $K = 3$ and $L = 2$ to visualize the digits in the latent space. The results can be seen in Figure 5. aPPCA with $K = 3$ and $L = 2$ leads to three 2-dimensional subspaces: subspace 1 is spanned by features 1 and 2; subspace 2 by features 2 and 3; subspace 3 by features 1 and 3. Furthermore, Figure 5 shows that aPPCA separates most of the distinctly different geometries where each subspace highlights a different aspect of the input space. For example, Figure 6 shows that when compared to subspace 1, subspace 2 and 3 capture more ‘thin’ digits, subspace 2 and subspace 3 capture different variations of the digits; for example the digit seven in subspace 3 has a ‘dash’ whereas in subspace 2 it has no dash. The visualization reduces the crowding effect of PCA and produces multiple 2D plots which jointly summarize the information about the observed data.
Figure 5: Plot of the three, 2-dimensional subspaces discovered by parametric aPPCA on 10,000 samples from the MNIST digit dataset

Figure 6: Behavior of different MNIST digits in different subspaces.

5.2 Data pre-processing

With the emergence of flexible manifold embedding and other nonlinear algorithms, PCA is less widely-used as a visualization tool, but remains in heavy use for data whitening. This is often used pre-processing step which 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, PCA works by “rotating” the data in observation space, retaining dimensionality unlike with visualization applications.
Figure 7: Supervised classification accuracy on MNIST handwritten digit recognition using a multilayer perceptron with one hidden layer. The dashed line is the benchmark of classifying MNIST using a multilayer perceptron on the original data. On the x-axis we show the number of reduced dimensions (i.e. principal component projections) used for different instances of the same classifier trained now on reduced dimensionality MNIST. The y-axis indicates the out-of-sample accuracy scale, given 10-fold cross-validation was adopted.

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 that, we compare a classifier train on raw data with the same classifier on the first few principal component projections of the data where the principal components 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 dataset, before training a multilayer perceptron. We train a simple multilayer perceptron with one hidden layer with a softmax activation function on a 9000-image subset of the 784-dimensional MNIST dataset with 1000 images reserved for testing. We compare the performance of the same classifier network when (1) trained on the original 784-dimensional pre-processed data, (2) trained on lower dimensional projection of the data using PCA (3) trained on locally whitened data by aPPCA (K-dimensional). The classifier is a multilayer perceptron in all 3 scenarios. Figure 7 shows the classification accuracy of these three different pre-processing approaches as we vary K, i.e. the number of principal components to which we project down the data. For aPPCA, we have kept $L = K - 1$ for simplicity. We have also seen increases in performance if multiple, separate classifiers are trained on each L-dimensional subspace, but these are not reported since it is not fair comparison with PCA.

A key feature of the aPPCA algorithm when used for localized data whitening is that it estimates more robust subspaces which can be seen in the smaller number of subspaces (i.e. principal components or columns of $W$) required for training of the same classifier, to achieve better out-of-sample performance. The multilayer perceptron trained on PCA whitened data requires more subspaces in training to achieve comparable out-of-sample results.

5.3 Blind source separation in fMRI

Functional magnetic resonance imaging (fMRI) is a technique for the non-invasive study of brain functions. fMRI can act as indirect measure of neuronal activation in the brain, by detecting blood oxygenation level dependent (BOLD) contrast [41]. BOLD relies on the fact that oxygenated (diamagnetic) and deoxygenated (paramagnetic) blood have different magnetic properties. When neurons fire there is a resultant 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 3-dimensional images. However, data can be also represented as a 2-dimensional matrix using vectorized voxel matrices against time (voxels by time). In this representation each matrix column contains all voxels from the brain image (or the subset selected for analysis) from a single time point. Although useful, fMRI data often suffers from low image contrast-to-noise ratio, it is biased from subject head motions, scanner drift (i.e. due to overheating of the equipment) and from signals from irrelevant physiological sources (cardiac or pulmonary). Therefore, direct analysis of raw fMRI measurements is rare [42] 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
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 localization. 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 t-statistic map that shows the voxels that have significant correlation with the task. Map is family-wise error (FWE) corrected at $p < 0.05$ at voxel threshold $p < 0.001$.

of fMRI series and localization of active spatial regions of the brain are variants of linear dimensionality reduction methods such as PCA, FA and ICA [43, 44, 45, 42, 46]. Typically, of main interest is then analysis of a representative subset of the inferred components or factors respectively, instead of the use of raw data.

A key problem with this approach is that these linear methods assume that the components/factors are a linear combination of all of the data, i.e. with other words PCA, FA and ICA would assume that all components are active for the full duration of the recording. Common implementations for fMRI series [47, 48] might adopt thresholding the inferred components or using sparse alternatives of the decomposition techniques. However, these still can lead to biased decomposition into components and we are likely to overestimate the firing area of the brain for some components and completely overlooking functional areas of the brain active for shorter periods of time. Here, we show that using our proposed adaptive linear methods, are better motivated model to alleviate this problem and can infer better localized
Figure 9: We also show the time-course associated with the component (same for PCA and aPPCA) and how it correlates with the predicted BOLD activity during the presentation of videos and recall task. Note that there is some mixing in that component 1 is not completely uncorrelated with the recall task.

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 parametric aPPCA to fMRI data collected from a single participant while exposed to continuous visual stimuli. fMRI data was initially realigned to correct for subject motion and registered to a group template (Montreal Neurological Institute Template). Using 3T Siemens scanner, a whole brain image with voxel resolution of $2 \times 2 \times 2$ mm was acquired each 0.8 seconds. The data had 215,302 voxels and 989 time points. aPPCA decomposition was performed by treating time points as features, which is standard 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 8.

Direct quantitative evaluation of pre-processing tools of fMRI data is an open problem, due to the lack of clear definition of the 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 highest error with highest standard deviation (i.e. average root mean square error (RMSE) of 16.5, standard deviation of RMSE of 4.8) was achieved using PPCA. aPPCA reconstruction gradually reduces these figures depending on the ratio of $K$ and $L$ used with best scoring reconstruction having average RMSE of 14.1 and standard deviation of RMSE (across voxels) of 3.0. The lower standard deviation of error across the voxels supports our hypothesis of better preserved local region information. Due to the simplicity of the imaging setup, both methods were able to identify components highly correlated to the stimuli, see Figure 9. 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 common analysis 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 voxels association with specific components (i.e. $Z$ switches off voxels not part of a component) in a principled fashion as a part of a probabilistic model. In addition, the user has explicit control over the contrast voxels used in different components (ratio of $K$ and $L$) and this can be useful for achieving better spatial localization, without heuristic thresholding.

6 Summary and conclusions

In this report, we have studied generic discrete latent variable augmentation for ubiquitous linear Gaussian methods applied for feature learning, whitening and dimensionality reduction applications. We have proposed a nonparametric latent feature PCA model which learns a flexible set of principal components all of which are computed as a linear combination of only subsets of the input data. The report details some shortcomings with existing Bayesian nonparametric linear Gaussian methods and demonstrates that flexible alternatives can be derived using hypergeometric distributions. This leads to novel aFA and aPPCA models which be trained efficiently, which overcome the inherent over-partitioning in Beta process and allows for more flexible regularization of the model capacity, compared to Beta-Bernoulli models.
The proposed models can be extended to many other related methods such as: generalized linear Gaussian models, Gaussian process latent variable models (GPLVMs), kernel PCA methods, and others. Dai et al. [11] 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, can in principle allow for richer model of the manifold and more compact modelling using 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 first on a wide range of synthetic latent feature Gaussian data sets and then also on MNIST and brain imaging fMRI data. Our synthetic study shows that a wide range of feature allocation distributions can be captured with a multivariate hypergeometric model. We have applied aPPCA to MNIST variational autoencoder projections, to show that in can be used to identify images sharing clear graphics features. aFA was applied to nearly raw digits to show that images of visually similar digits share more factors together. The report concludes with an application of aPPCA to a widely-encountered problem in brain imaging with fMRI, and demonstrate 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 localization of active brain regions.
A Adaptive PCA

In this section we demonstrate that the proposed parametric APPCA model from Section 4 is indeed a generalization of the ubiquitous PCA and using small variance asymptotics [23]. Let us first start by marginalizing out the discrete and continuous latent variables \( \{x_n, z_n\} \) which are not of explicit interest in conventional PCA approach. To compute the marginal likelihood of \( y_n \) we compute the expectations:

\[
E_{\mathcal{P}(x_n, z_n)}[y_n] \quad \text{and} \quad E_{\mathcal{P}(x_n, z_n)}\left[ (y_n - E[y_n]) (y_n - E[y_n])^T \right]
\]

where we use \( E[\cdot] = E_{\mathcal{P}(x_n, z_n)}[\cdot] \) for notational convenience. We express the moments of the marginal likelihood starting with the posterior mean of the marginal, \( E[y_n] \):

\[
E[y_n] = E[W(x_n \odot z_n) + \mu + \epsilon_n]
= W(E[x_n] \odot E[z_n]) + \mu + E[\epsilon_n]
= W(0 \odot \rho) + \mu + 0
= \mu
\]

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

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

For the variance of the marginal, we can write:

\[
E\left[ (y_n - E[y_n]) (y_n - E[y_n])^T \right] = E\left[ (W(x_n \odot z_n) + \epsilon_n) (W(x_n \odot z_n) + \epsilon_n)^T \right]
= W\rho W^T + \sigma^2 I_D
\]

Finally, using the obtained expression for \( E[y_n] \) and \( E\left[ (y_n - E[y_n]) (y_n - E[y_n])^T \right] \), combined with the Gaussian likelihood of \( y_n \) resulting in a linear Gaussian model, we can write the marginal likelihood as:

\[
P\left( y_n \mid W, \rho, \sigma \right) = \frac{1}{(2\pi)^{D/2} \left| C \right|^{-1/2}} \exp \left( -\frac{1}{2} y_n^T C^{-1} y_n \right)
= \frac{1}{(2\pi)^{D/2} \left| C \right|^{-1/2}} \exp \left( -\frac{1}{2} y_n^T C^{-1} y_n \right)
\]

where we used \( C = W\rho W^T + \sigma^2 I_D \) 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 \( \rho \) which can be scalar shared across each dimension 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 \( \rho \) are full of ones, which occurs when the matrix \( Z \) is full of ones implying all observations are active in all \( K \) number of one-dimensional subspaces.

The complete data log-likelihood of the collapsed model is:

\[
\mathcal{L} = \sum_{n=1}^{N} \ln \left( P\left( y_n \mid W, \rho, \sigma \right) \right)
= -\frac{N}{2} \left( D \ln (2\pi) + \ln |C| + tr \left( C^{-1} S \right) \right)
\]

where \( S = \frac{1}{N} YY^T \). To find the maximum likelihood estimate for \( W \), we differentiate the likelihood and solve:

\[
\frac{d\mathcal{L}}{dW} = -\frac{N}{2} \left( 2C^{-1} W \rho - 2C^{-1} SC^{-1} W \rho \right) = 0
\]
\[ C^{-1} W \rho = C^{-1} S C^{-1} W \rho \]
\[ W^{ML} \rho = S C^{-1} W^{ML} \rho \]

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

\[ W^{1/2} = U L V^T \]

which leads to:

\[ C^{-1} W^{1/2} = U (L^2 + \sigma^2 I_K)^{-1} V^T \]

then:

\[ S C^{-1} W^{1/2} = W^{1/2} \]
\[ S U L (L^2 + \sigma^2 I_K)^{-1} V^T = U L (L^2 + \sigma^2 I_K) L \]

which implies that \( u_j \) is the eigenvector of \( S \) with eigenvalue of \( \lambda_j = \sigma^2 + \ell_j^2 \). Therefore all potential solutions for \( W^{ML} \) may be written as

\[ W^{ML} = U_K (K_K - \sigma^2 I_K)^{1/2} R \rho^{-1/2} \]

where

\[ k_{jj} = \begin{cases} \lambda_j & \text{eigenvalue of } u_j \\ \sigma^2 & \text{otherwise} \end{cases} \]

Where \( R \) is \((D \times K)\) orthonomal matrix. The weighting term \( \rho \) allows to explicit control over the scale of the different projection axis. \( \rho \) controls if we should place more or less importance on the role of the input to the projection axis, which is meant to reflect our posterior belief of re-scaling due to not all data points sharing all subspaces. Appropriate scaling with \( \rho \) can address a well known pitfalls of PCA such as: the disproportionate crowding of the projections due to outliers or multi-modalities; the sphericalization of the projection.

**B Updating hyperparameters**

**Updating \( \sigma^2 \)**

We place a inverse-Gamma prior on \( \sigma^2 \) with parameters \( \{ \gamma, \vartheta \} \)

\[ p(\sigma^2|\gamma, \vartheta) = \frac{\vartheta^{\gamma}}{\Gamma(\gamma)} (\sigma^2)^{-\gamma-1} \exp \left[ -\frac{\vartheta}{\sigma^2} \right] \]

then the posterior distribution over \( \sigma^2 \) is

\[ p(\sigma^2|\gamma, \vartheta, Y, W, X, Z) = \frac{\vartheta^{\gamma}}{\Gamma(\gamma)} (\sigma^2)^{-\gamma-1} \exp \left[ -\frac{\vartheta}{\sigma^2} \right] \]
\[ \times \frac{1}{(2\pi\sigma^2)^{ND/2}} \exp \left( -\frac{1}{2\sigma^2} \sum_{n=1}^{N} \left[ (y_n - W (x_n \odot z_n))^T (y_n - W (x_n \odot z_n)) \right] \right) \]
\[ \propto (\sigma^2)^{-\gamma+ND/2-1} \]
\[ \times \exp \left( -\frac{1}{\sigma^2} \frac{1}{2} \text{tr} \left[ (Y - W (X \odot Z))^T (Y - W (X \odot Z)) + \vartheta \right] \right) \]
which is still a inverse-Gamma distribution with parameters $\gamma^{\text{post}} = \gamma + \frac{ND}{2}$ and $\vartheta^{\text{post}} = \frac{1}{2} \text{tr} \left[ (Y - W(X \odot Z))^T (Y - W(X \odot Z)) \right] + \vartheta$

Updating $\alpha$

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

$$ p(\alpha|\lambda, \mu) = \frac{\mu^\lambda}{\Gamma(\lambda)} \frac{1}{\Gamma(\lambda)} \exp\left[-\mu \alpha\right] $$

then the posterior distribution over $\alpha$ is

$$ p(\alpha|\lambda, \mu, Y, W, X, Z) = \frac{\mu^\lambda}{\Gamma(\lambda)} \frac{1}{\Gamma(\lambda)} \exp\left[-\mu \alpha\right] \times \exp\left(-\alpha H_N\right) \alpha^K \times \left( \prod_{k=1}^{K} \frac{(m_k - 1)! (N - m_k)!}{(N - 1)!} \right) \alpha^{\lambda + K - 1} \exp\left(-\alpha (H_N + \mu)\right) $$

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

C  Projection matrix update using PYMANOPT

For both variants of the aPPCA, the matrix $W$ is updated numerically by minimising the negative-log of of Equation (14) over the Stiefel manifold with respect to the matrix $W$. Figure 10 shows the implementation of this using the PYMANOPT toolbox [36].
References

[1] J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, “Bayesian factor regression models in the “large p, small n” paradigm,” *Bayesian statistics*, vol. 7, pp. 733–742, 2003.

[2] D. Knowles and Z. Ghahramani, “Infinite sparse factor analysis and infinite independent components analysis,” in *International Conference on Independent Component Analysis and Signal Separation*. Springer, 2007, pp. 381–388.

[3] J. Paisley and L. Carin, “Nonparametric factor analysis with beta process priors,” in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 777–784.

[4] H. Zou, T. Hastie, and R. Tibshirani, “Sparse principal component analysis,” *Journal of computational and graphical statistics*, vol. 15, no. 2, pp. 265–286, 2006.

[5] I. T. Jolliffe, N. T. Trendafilov, and M. Uddin, “A modified principal component technique based on the lasso,” *Journal of computational and Graphical Statistics*, vol. 12, no. 3, pp. 531–547, 2003.

[6] M. E. Tipping and C. M. Bishop, “Probabilistic principal component analysis,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[7] K. Pearson, “Liii. on lines and planes of closest fit to systems of points in space,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[8] M. E. Tipping and C. M. Bishop, “Probabilistic principal component analysis,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[9] H. H. Harman, *Modern factor analysis*. Univ. of Chicago Press, 1960.

[10] P. Comon, “Independent component analysis, a new concept?” *Signal processing*, vol. 36, no. 3, pp. 287–314, 1994.

[11] Z. Dai, J. Hensman, and N. Lawrence, “Spike and slab gaussian process latent variable models,” *arXiv preprint arXiv:1505.02434*, 2015.

[12] K. R. Campbell and C. Yau, “Probabilistic modeling of bifurcations in single-cell gene expression data using a bayesian mixture of factor analyzers,” *Wellcome open research*, vol. 2, 2017.

[13] L. Parsons, E. Haque, and H. Liu, “Subspace clustering for high dimensional data: a review,” *Acm Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 90–105, 2004.

[14] R. Vidal, “Subspace clustering,” *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011.

[15] Z. Ghahramani, G. E. Hinton et al., “The em algorithm for mixtures of factor analyzers,” Technical Report CRG-TR-96-1, University of Toronto, Tech. Rep., 1996.

[16] Z. Ghahramani and M. J. Beal, “Variational inference for bayesian mixtures of factor analysers,” in *Advances in neural information processing systems*, 2000, pp. 449–455.

[17] P. Rai and H. Daumé, “The infinite hierarchical factor regression model,” in *Advances in Neural Information Processing Systems*, 2009, pp. 1321–1328.

[18] C. Elvira, P. Chainais, and N. Dobigeon, “Bayesian nonparametric principal component analysis,” *arXiv preprint arXiv:1709.05667*, 2017.

[19] G. Di Benedetto, F. Caron, and Y. W. Teh, “Non-exchangeable feature allocation models with sublinear growth of the feature sizes,” *arXiv preprint arXiv:2003.13491*, 2020.

[20] Y. W. Teh and D. Gorur, “Indian buffet processes with power-law behavior,” in *Advances in neural information processing systems*, 2009, pp. 1838–1846.

[21] D. A. Van Dyk and T. Park, “Partially collapsed gibbs samplers: Theory and methods,” *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 790–796, 2008.

[22] J. Chesson, “A non-central multivariate hypergeometric distribution arising from biased sampling with application to selective predation,” *Journal of Applied Probability*, vol. 13, no. 4, pp. 795–797, 1976.

[23] T. Broderick, B. Kulis, and M. Jordan, “Mad-bayes: Map-based asymptotic derivations from bayes,” in *International Conference on Machine Learning*, 2013, pp. 226–234.

[24] Y. P. Raykov, A. Boukouvalas, M. A. Little et al., “Simple approximate map inference for dirichlet processes mixtures,” *Electronic Journal of Statistics*, vol. 10, no. 2, pp. 3548–3578, 2016.

[25] Y. Raykov, “A deterministic inference framework for discrete nonparametric latent variable models: learning complex probabilistic models with simple algorithms,” Ph.D. dissertation, Aston University, 2017.
A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian data analysis*. CRC press, 2013.

B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell, “Stan: A probabilistic programming language,” *Journal of statistical software*, vol. 76, no. 1, 2017.

C. J. Maddison, A. Mnih, and Y. W. Teh, “The concrete distribution: A continuous relaxation of discrete random variables,” *arXiv preprint arXiv:1611.00712*, 2016.

A. Nishimura, D. Dunson, and J. Lu, “Discontinuous hamiltonian monte carlo for discrete parameters and discontinuous likelihoods,” *arXiv preprint arXiv:1705.08510*, 2017.

A. E. Raftery and S. M. Lewis, “[practical markov chain monte carlo]: comment: one long run with diagnostics: implementation strategies for markov chain monte carlo,” *Statistical science*, vol. 7, no. 4, pp. 493–497, 1992.

H. D. Tagare, “Notes on optimization on stiefel manifolds,” in *Technical report, Technical report*. Yale University, 2011.

C. Bingham, “An antipodally symmetric distribution on the sphere,” *The Annals of Statistics*, pp. 1201–1225, 1974.

C. Khatri and K. V. Mardia, “The von mises–fisher matrix distribution in orientation statistics,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 95–106, 1977.

C. S. Herz, “Bessel functions of matrix argument,” *Annals of Mathematics*, pp. 474–523, 1955.

C. J. Fällaize and T. Kypraios, “Exact bayesian inference for the bingham distribution,” *Statistics and Computing*, vol. 26, no. 1-2, pp. 349–360, 2016.

J. Townsend, N. Koep, and S. Weichwald, “Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 4755–4759, 2016.

N. L. Hjort et al., “Nonparametric bayes estimators based on beta processes in models for life history data,” *The Annals of Statistics*, vol. 18, no. 3, pp. 1259–1294, 1990.

Y. LeCun, C. Cortes, and C. J. Burges, “The mnist database of handwritten digits, 1998,” URL http://yann. lecun. com/exdb/mnist, vol. 10, p. 34, 1998.

J. Paisley and L. Carin, “Nonparametric factor analysis with beta process priors,” in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 777–784.

D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.

E. Zarahn, G. K. Aguirre, and M. D’Esposito, “Empirical analyses of bold fmri statistics,” *NeuroImage*, vol. 5, no. 3, pp. 179–197, 1997.

R. H. Pruim, M. Mennes, D. van Rooij, A. Llera, J. K. Buitelaar, and C. F. Beckmann, “Ica-aroma: A robust ica-based strategy for removing motion artifacts from fmri data,” *Neuroimage*, vol. 112, pp. 267–277, 2015.

V. D. Calhoun, T. Adali, L. K. Hansen, J. Larsen, and J. J. Pekar, “Ica of functional mri data: an overview,” in *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation*. Citeseer, 2003.

J. Taglia, S. Ryali, T. Chen, K. Supekar, W. Cai, and V. Menon, “Bayesian switching factor analysis for estimating time-varying functional connectivity in fmri,” *Neuroimage*, vol. 155, pp. 271–290, 2017.

C. F. Beckmann and S. M. Smith, “Tensorial extensions of independent component analysis for multisubject fmri analysis,” *Neuroimage*, vol. 25, no. 1, pp. 294–311, 2005.

P. A. Højen-Sørensen, O. Winther, and L. K. Hansen, “Analysis of functional neuroimages using ica with adaptive binary sources,” *Neurocomputing*, vol. 49, no. 1-4, pp. 213–225, 2002.

M. J. McKeown, L. K. Hansen, and T. J. Sejnowsk, “Independent component analysis of functional mri: what is signal and what is noise?” *Current opinion in neurobiology*, vol. 13, no. 5, pp. 620–629, 2003.

V. D. Calhoun, J. Liu, and T. Adali, “A review of group ica for fmri data and ica for joint inference of imaging, genetic, and erp data.” *Neuroimage*, vol. 45, no. 1, pp. S163–S172, 2009.