# Bayes High-Dimensional Density Estimation Using Multiscale Dictionaries

by

Ye Wang

Department of Statistics
Duke University

Date: _____

Approved:

_____
David Dunson, Supervisor

_____
Surya Tokdar

_____
Charles Becker

Thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in the Departments of Statistical Science and Economics
in the Graduate School of Duke University

2014

# ABSTRACT

## Bayes High-Dimensional Density Estimation Using Multiscale Dictionaries

by

Ye Wang

Department of Statistics
Duke University

Date: _____

Approved:

_____
David Dunson, Supervisor

_____
Surya Tokdar

_____
Charles Becker

An abstract of a thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in the Departments of Statistical Science and Economics in the Graduate School of Duke University

2014

# Abstract

Although Bayesian density estimation using discrete mixtures has good performance in modest dimensions, there is a lack of statistical and computational scalability to high-dimensional multivariate cases. To combat the curse of dimensionality, it is necessary to assume the data are concentrated near a lower-dimensional subspace. However, Bayesian methods for learning this subspace along with the density of the data scale poorly computationally. To solve this problem, we propose an empirical Bayes approach, which estimates a multiscale dictionary using geometric multiresolution analysis in a first stage. We use this dictionary within a multiscale mixture model, which allows uncertainty in component allocation, mixture weights and scaling factors over a binary tree. A computational algorithm is proposed, which scales efficiently to massive dimensional problems. We provide some theoretical support for this method, and illustrate the performance through simulated and real data examples.

# Contents

# List of Tables

# List of Figures

# List of Abbreviations and Symbols

## Symbols

| | |
|---|---|
| $\Re$ | The set of real numbers. |
| $\Re^D$ | The set of real valued $D$ dimensional vectors. |
| $\|A\|_F$ | Frobenius norm. |
| $\ln$ | Natural logarithm. |
| $d_{TV}(P_1, P_2)$ | Total variation distance between $P_1$ and $P_2$. |

## Abbreviations

| | |
|---|---|
| MSE | Mean squared error. |
| EN | Elastic net. |
| PLSR | Partial least squares regression. |
| GMRA | Geometric multiresolution analysis (Allard et al., 2012). |
| METIS | Serial Graph Partitioning and Fill-reducing Matrix Ordering (Karypis and Kumar, 1998). |
| ANN | Approximate nearest neighbour algorithm (Arya et al., 1998). |
| RF | Random forest. |

# Acknowledgements

When I arrived at Duke to pursue a Master degree in Statistical and Economical modeling, pursuing a PhD degree in statistics was not on the top of my list. I would like to express thanks to everyone in the department of statistical science, you all have been my mentors and friends for giving me the inspiration, understanding, and ability to learn the science of Bayesian statistics.

I would like to thank my advisor, professor David Dunson, for his invaluable guidance and uncanny intuition when I started this project, and his patience with me when my draft turned out to be terribly written again and again and his encouragement when my simulation results overthrew my proposed model. I would not have been completing this thesis if not for him.

I thank my coauthor on this project, professor Antonio Canale, for his review and feedbacks even when he was super busy. I thank professor Mike West, for his kind wonderful course "Probability models" and his guidance in PhD application and future planning. I thank my committee members, professor Surya Tokdar, professor Charles Becker for serving as my committee members. I also thank my former and current programme advisors, professor Surya Tokdar, professor Charles Becker and professor Jerry Reiter, for their patience in listening to my stories and for their assistance and encouragements during my depression.

A special thanks to my family for their funding my graduate study, which is quite a heavy burden. Words cannot express how grateful I am. I would also like to thank Diane

Bryson, David Klemish who supported me in writing, and incented me to strive towards my goal.

# 1

# Introduction

Let $y_i = (y_{i1}, \ldots, y_{iD})^T$, for $i = 1, \ldots, n$, be a sample from an unknown distribution having support in a subset of $\Re^D$. We are interested in estimating its density when $D$ is large, and the data have a low-dimensional structure with intrinsic dimension $p$ such that $p \ll D$. Kernel methods work well in low dimensions, but face challenges in scaling up to large $D$ settings. In particular, optimally one would allow separate bandwidth parameters for the different variables to accommodate differing smoothness, but then there is the issue of how to choose the high-dimensional vector of bandwidths or alternatively the kernel covariance matrix. Clearly, cross validation involves an intractable computation cost and plugging in arbitrary values is not recommended, since bandwidth choice fundamentally impacts performance (Liu et al., 2007).

Bayesian nonparametric models (Escobar and West, 1995; Rasmussen, 1999; Fokoué and Titterington, 2003) provide an alternative approach for density estimation, specifying priors for the bandwidth parameters allowing adaptive estimation without cross-validation (Shen et al., 2013). However, inference is prohibitively costly. To scale up nonparametric Bayes inference, one can potentially rely on a maximum posteriori (MAP) estimation (Ghahramani et al., 1996) or variational Bayes (VB) (Ghahramani and Beal, 1999). Is-

sues with MAP include difficulties in efficient estimation in high-dimensions, with the EM algorithm tending to converge slowly to a local mode, and lack of characterization of uncertainty. Although VB provides an approximation to the full posterior instead of just the mode, it is well known that posterior uncertainty is substantially underestimated (Wang and Titterington, 2004) and in being implemented with EM, VB inherits the computational problems of MAP estimation. To improve performance, one direction is to reduce effective dimensionality through imposing constraints on the multivariate density. This can be accomplished via copula models; for example, using a Gaussian copula to characterize dependence while letting the marginals be flexible (Bedford and Cooke, 2002; Joe, 2005; Lopez-Paz et al., 2013), or learning a graphical dependence structure that restricts certain variables to be conditionally independent (Jordan, 2004).

Manifold learning methods (Roweis and Saul, 2000; Tenenbaum et al., 2000; Lawrence, 2005) provide computationally efficient and geometrically dimension reduction, motivating an alternative way to characterize the density via a low-dimensional embedding. While most of these methods have focused on visualization, manifold Parzen windows (Bengio and Vincent, 2004) is a notable exception that has attempted to combine density estimation and manifold learning. The model applies dimension reduction and fits a Gaussian "pancake" to the neighbourhood area of each data point, integrating local geometric information into a kernel density estimator. However, overfitting might come in when every data point is associated, by the same weight, with a Gaussian. Moreover, the model can be sensitive to the prior choice of intrinsic dimension $p$, and only provides a point estimate. Motivated by this work, we designed an empirical Bayes nonparametric density estimator that combines density estimation and manifold learning, characterizes the uncertainty, scales up to problems with massive dimensions and is capable of automatically learning the intrinsic dimension. We extended the idea of Bengio and Vincent (2004) by learning a set of multiscale geometric dictionaries at a first stage. The model is illustrated through simulated and real data examples.

To relate this thesis to research articles, I reference the following paper:

> Wang Y., Canale A. and Dunson D. B. (2014) "Bayes High-Dimensional Density Estimation Using Multiscale Dictionaries" Manuscript, Department of Statistical Science, Duke University.

The remainder of the paper is organized as follows. In Chapter 2 we propose an empirical Bayes dictionary learning approach for scaling up Bayesian factor analysis, allowing rapid estimation of a large covariance matrix for Gaussian data. In Chapter 3 we generalize this approach to a multiscale mixture model, enabling estimation of a high-dimensional density. Chapter 4 considers real data applications to image inpainting data, and Chapter 5 contains a discussion.

# 2

# Empirical Bayes dictionary learning in factor models

We initially focus on the case in which $y_i \sim N_D(\mu, \Omega)$ is assumed, with $\mu = (\mu_1, \ldots, \mu_D)^T$ a mean vector and $\Omega$ a $D \times D$ covariance matrix. Assuming data are centered prior to analysis, we focus on the challenging problem of estimating the high-dimensional covariance $\Omega$. When $D$ is large, potentially with $D \gg n$, we need to incorporate dimensionality reduction in estimating $\Omega$. Analytic factorizations that let $\Omega = \Lambda\Lambda^T + \sigma^2 I$ are intimately related to principal components analysis (Tipping and Bishop, 1999), and have been highly successful in applications. The effective number of unknown parameters in $\Omega$ can be massively reduced by assuming $\Lambda$ is a $D \times p$ matrix with $p \ll D$, while additionally assuming $\Lambda$ is sparse and so has many elements near zero. Carvalho et al. (2008) and Bhattacharya and Dunson (2011) (among many others) have successfully applied such sparse factor models in the Bayesian paradigm, but face problems in scaling to really large $D$ and considering extensions to more intricate models that avoid Gaussian assumptions. To simplify computation and construct a building block for our final model, we propose an empirical Bayes approach that avoids directly placing priors on all the free parameters in the analytic factorization via the use of dictionary learning.

## 2.1 Formulation

Our dictionary learning-based factorization model (DLF) is

$$y_i \sim N_D(\mu, \Phi\Sigma\Phi^T + \sigma^2 I) \tag{2.1}$$

where $\Sigma$ is a non-negative diagonal scaling matrix and $\Phi$ is a $D \times d$ orthonormal matrix. Both $\mu$ and $\Phi$ are estimated in a first stage, with priors then placed on $\Sigma$ and $\sigma^2$. When $\sigma^2$ is small, the data are concentrated near a $p$ dimensional subspace. We treat $d$ as an upper bound for the unknown $p$, with the prior for $\Sigma$ allowing adaptive removal of unnecessary dimensions.

Model (2.1) can be equivalently expressed as

$$y_i - \mu = \Phi\eta_i + \epsilon_i \tag{2.2}$$

where $\eta_i \in \Re^d$, for $i = 1, 2, \ldots, n$, are latent variables drawn independently from $N_d(0, \Sigma)$ and $\epsilon_i \in \Re^D$, for $i = 1, 2, \ldots, n$, is a residual noise vector drawn from $N_D(0, \sigma^2 I)$. Viewing $\eta_i$ as the coordinates on a $d$-dimensional linear subspace, the columns of $\Phi$ form a basis (or dictionary) for the subspace, with span$(\Phi)$ denoting the subspace spanned by the columns of $\Phi$. As a simple choice, we fix $\mu = n^{-1}\sum_{i=1}^n y_i$, though regularized choices can also be considered, and focus on learning the dictionary $\Phi$. In particular, we want to learn a dictionary that allows accurate approximation of the covariance $\Omega$, while reducing dimensionality. Before describing the algorithm proposed, we start with some definitions and preliminaries.

**Definition 1.** *A $D \times d$ orthonormal matrix $\Phi_0$ is called a **d-dictionary** if it solves*

$$\min_{\Phi} \quad E(\|\Omega - \Phi\Sigma\Phi^T\|_F)$$
$$s.t. \quad \Sigma = \mathrm{diag}(\alpha_1^2, \ldots, \alpha_d^2)$$

*where $d$ is the upper bound of the intrinsic dimension subject to $D \geqslant d$.*

Let the $D \times D$ matrix $C$ be the sample covariance matrix of data $\{y_i\}_{i=1}^n$ defined above, and let $C = \Phi_C \Sigma_C \Phi_C^T$ be the singular value decomposition of $C$. Partition $\Phi_C$ and $\Sigma_C$ as follows:

$$\Phi_C = [\Phi_{SVD}^d \; \Phi_r], \Sigma_C = \begin{bmatrix} \Sigma_{SVD}^d & 0 \\ 0 & \Sigma_r \end{bmatrix}$$

where $\Sigma_{SVD}^d$ is $d \times d$, $\Sigma_r$ is $(D-d) \times (D-d)$, $\Phi_{SVD}^d$ is $D \times d$ and $\Phi_r$ is $D \times (D-d)$.

**Theorem 1.** *$\Phi_{SVD}^d$ is a d-dictionary.*

The above theorem can be easily proved using Eckart–Young–Mirsky theorem and the fact that sample covariance is an unbiased estimator of the covariance.

We apply the fast rank-$d$ SVD algorithm (Rokhlin et al., 2009) to learn $\Phi_{SVD}^d$ at the first stage, with the computational cost being $O(nDd)$. It remains to specify a prior for the scaling matrix $\Sigma$ and the residual variance $\sigma^2$.

Using some linear algebra tricks, the likelihood function of model (2.1) can be simplified as

$$\begin{aligned} L(y_{1:n}) \;\; \propto \;\; & (\sigma^2)^{-Dn/2} \prod_{m=1}^{d} u_m^{n/2} \times \\ & \exp\left\{ -\frac{1}{2}\sigma^{-2} \sum_{i=1}^{n} [A_i - \sum_{m=1}^{d} (1-u_m)(Z_i^{(m)})^2] \right\}, \end{aligned} \tag{2.3}$$

where $u_m = (1 + \sigma^{-2}\alpha_m^2)^{-1}, m = 1, \ldots, d$, $Z_i = \Phi^T \tilde{y}_i$, with $Z_i^{(m)}$ denoting its $m$th element, $A_i = \tilde{y}_i^T \tilde{y}_i$, and $\tilde{y}_i = y_i - \mu$.

We first specify a prior for the "full" model where $d = D$. The "full" model is given by

$$y_i \sim N_D(\mu, \Phi_{SVD}^D \Sigma (\Phi_{SVD}^D)^T + \sigma^2 I).$$

When $p$ is small, which is approximately true in practice, the information contained in the last $D - p$ columns of $\Phi_{SVD}^D$ (columns of $\Phi_{SVD}^D$ are ordered to be descending in their

singular values) is trivial and treated as noise. We use a speciallly tailored prior that shrinks $\alpha_m^2$ to zero more aggressively as $m$ grows; this reduces MSE by pulling the small signals aggressively towards zero. This is equivalent to shrinking $u_m$ increasingly for larger $m$. To accomplish this adaptive shrinkage, we propose a multiplicative exponential process prior that adapts the prior of Bhattacharya and Dunson (2011), while placing an inverse-gamma prior on $\sigma^2$:

$$\begin{aligned} \sigma^{-2} &\sim \text{Ga}(a_\sigma, b_\sigma) \\ u_m &\sim \text{Ga}_{(0,1)}(\delta_m + 1, 1) \\ \delta_m &= \prod_{k=1}^{m} \tau_k \\ \tau_k &\sim \text{Exp}_{[1,\infty)}(a) \end{aligned} \tag{2.4}$$

where $\tau_k$, for all $k = 1, \ldots, d$, are independent truncated exponential random variables, $\delta_m$ and $\tau_m$ are the global and the local shrinkage parameter for the $m$th column, respectively. Since $\tau_k \geqslant 1$ for all $k = 1, \ldots, D$, $\delta_m = \prod_{k=1}^{m} \tau_k$ is increasing with respect to $m$. As a result, $u_m$ is stochastically approaching one since the truncated gamma density concentrates around one as $\delta_m$ increases

Although we specify priors for $u_m$ for $m = 1, \ldots, D$, for large $D$ it is wasteful to conduct computation for the full model, because as $m$ increases $u_m$ is shrunk very strongly to one, and the excess dimensions are effectively discarded. Hence, we propose to truncate the model by setting $u_m = 1$ ($\alpha_m^2 = 0$) for $m > d$, with $d$ an upper bound on the number of factors. The following theorem shows that that approximation error of the truncated prior decreases exponentially in $d$.

**Theorem 2.** *Let* $\Omega = \Phi_{SVD}^D \Sigma (\Phi_{SVD}^D)^T + \sigma^2 I$ *and* $\Omega_d = \Phi_{SVD}^d \Sigma_d (\Phi_{SVD}^d)^T + \sigma^2 I$*, where* $\Sigma_d = \text{diag}(\alpha_1^2, \ldots, \alpha_d^2)$*. Then for any* $\epsilon > 0$*,*

$$Pr\{d_\infty(\Omega, \Omega_d) > \epsilon\} < \frac{6ba^d}{\epsilon(1-a)}$$

*for* $d > 2\log\{b/\epsilon(1-a)\}/\log(1/a)$*, where* $d_\infty(\Omega, \Omega_d)$ *is defined as* $\|\Omega - \Omega_d\|_\infty$*.* $\|A\|_\infty$ *calculates the maximum absolute row sum of the matrix* $A$*,* $b = E(\sigma^2)$ *and* $a = E(\frac{1}{\tau_1})$ *with* $\sigma^2$ *and* $\{\tau_m\}$ *defined as in (2.4).*

## 2.2 Posterior computation

The usual frequentist method of selecting an upperbound $d$ thresholds the singular values, leading to substantial sensitivity to threshold choice. When $D$ is large, $d$ has to be chosen in advance so that approximation of $\Phi_{SVD}^d$ can be achieved (Rokhlin et al., 2009). $d$ is usually conservatively chosen to ensure $d \geqslant p$, adding a burden to both computation and storage. We avoid this by automatically deleting redundant dictionary elements, and hence decreasing $d$, as computation proceeds. To achieve this we adopt an adaptive Gibbs sampler similar to that developed in Bhattacharya and Dunson (2011). Let $\mathcal{D}$ denote the set of deleted column indices (the deleted pool) and $\mathcal{R}$ denote the set of retained column indices (the remaining pool). The adaptive Gibbs sampler is summarized in Algorithm 1, where $c_0$ and $c_1$ are chosen to ensure frequent adaption at the beginning of the chain and an exponentially fast decay in frequency after that, and $tol$ is a prespecified threshold. We fix $c_0 = -1$, $c_1 = -0.005$ and $tol = 0.001$ as default. The sampling method can be summarized as follows,

1. Update $u_m$ for all $m = 1, 2, \ldots, d$ according to

$$\text{Ga}_{(0,1)}\left(\prod_{k=1}^{m}\tau_k + n/2, 1 + \frac{1}{2}\sigma^{-2}\sum_{i=1}^{n}(Z_h^{(m)})^2\right)$$

2. Update $\tau_m$ according to

$$\text{Exp}_{[1,\infty)}\left(a_\tau - \ln(\prod_{j>m-1} u_j)\right)$$

**Algorithm 1** Adaption at iteration $t$

compute $p(t) = \exp(c_0 + c_1 t)$, generate $g$ from $\mathrm{U}(0,1)$
**if** $g \leqslant p(t)$ **then**
  **for all** $m \in \mathcal{R}$ **do**
    compute $r_m^t = (\alpha_m^t)^2 / \max_{j \in \mathcal{R}}((\alpha_j^t)^2)$
    **if** $r_m^t \leqslant tol$ **then**
      Let $u_m^t = 1$, remove $m$ from $\mathcal{R}$ to $\mathcal{D}$,
    **end if**
  **end for**
  **if** $r_m^t \leqslant tol$ for all $m \in \mathcal{R}$ **then**
    **for all** $m$ **do**
      **if** $m \in \mathcal{D}$ **then**
        $pr(m) \propto r_m^{t-1}$
      **else**
        $pr(m) = 0$
      **end if**
    **end for**
    sample $m$ with probability $pr(m)$
    remove $m$ from $\mathcal{D}$ to $\mathcal{R}$.
  **end if**
**end if**

3. Update $\sigma^{-2}$ according to

$$Ga\left(a_\sigma + \frac{Dn}{2}, \frac{1}{2}\sum_{i=1}^{n}\{A_i - \sum_{j=1}^{d}(1-u_j)(Z_i^{(j)})^2\} + b_\sigma\right)$$

4. Adapt $d$ using Algorithm 1.

Since $\Phi_{SVD}^d$ and $\mu$ are learned at a first stage, $\{A_i\}_{i=1}^n$ and $\{Z_i\}_{i=1}^n$ are sufficient statistics and can be computed before the MCMC, whose computational costs are $O(nD)$ and $O(ndD)$ respectively. Hence, the computational cost of the MCMC is independent of $D$, leading to a small per iteration burden and allowing many samples to be collected.

Furthermore, in a $D$-dimensional problem, traditional factor analyzers will have $(2D - d)d/2 + nd + 1$ free parameters to update in the MCMC algorithm, while DLF only has $d + 1$ parameters to update, i.e, $d$ scaling parameters and one variance $\sigma^2$. Due to the reduced number of parameters and lower posterior dependence in these parameters, our Gibbs sampler for DLF converges and mixes dramatically faster than MCMC algorithms for fully Bayesian sparse factor models. This reduces the number of samples needed; we

Table 2.1: We estimated coverage of 95% predictive intervals out of sample based on 100 simulation replicates. The mean is reported in the first row, and the 95% interval based on 100 simulation replicates is reported in the second row.

| $D = 1000$ | $D = 2000$ | $D = 3000$ |
|---|---|---|
| 0.946 | 0.938 | 0.940 |
| (0.9, 0.99) | (0.88, 0.98) | (0.89, 0.98) |

run the sampler 1,000 iterations, with the first 500 as a burn-in. Experimental results show convergence occurs very fast, typically.

## 2.3 Simulation experiment

To assess the performance of the proposed Bayesian DLF (B-DLF) model, we conducted a simulation study. We simulated 100 independent samples of size $n = 600$ from three different scenarios involving increasing dimension, namely $D = 1000, 2000, 3000$ and fixed $p = 10$. The data are simulated as $y \sim N(0, \Lambda\Lambda^T + \sigma^2 I)$ where each entry of the $D \times p$ matrix $\Lambda$ is generated from $N(0, 25)$ and $\sigma$ is generated from $N(0, 0.1)$.

We split the samples into training and test subsets containing 500 and 100 observations respectively. We applied our Gibbs sampler to the training data and used the results to predict a randomly selected dimension given the others in the test set as in Müller et al. (1996). We assessed predictive performance using the MSE and 95% coverage. As prior specification we used $a_\sigma = 1/2$, $b_\sigma = 1/2$, $a = 0.05$, and fixed the upper bound to $d = 20$. Our model is compared with EN and PLSR. The B-DLF has a consistently better predictive performance than the competing methods as can be seen from Figure 2.1 showing the predictive MSE of the three competing methods. Coverage of 95% intervals were estimated for each simulated data sets; we present summaries across the 100 data sets in Table 2.1. Clearly the 95% intervals have close to 95% frequentist coverage on average in each case. Hence, we find no evidence that our empirical Bayes method under estimates uncertainty. Our method also consistently learns the true dimension $p$ as can be seen from

Figure 2.2, reporting the posterior mean inclusion probability of each of the $d$ dimensions of the dictionary.

The computational efficiency of B-DLF is illustrated in Figure 2.3. From the left plot, it can be seen that the computational cost of the proposed Gibbs sampler is independent from $D$ and, in the right plot, it can be seen that the overall computational time of our procedure scales well to higher dimensions, comparing favorably with EN and PLSR. B-DLF takes less than 10 seconds when running the algorithms in matlab version 2012a on a 32 bit windows 7 machine with a 5.2 GHz i5-3320M processor.

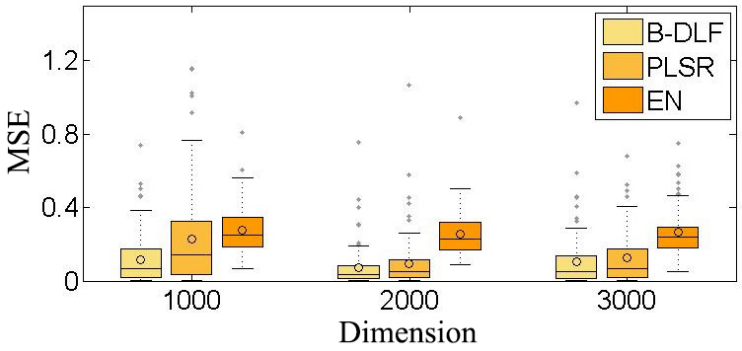


FIGURE 2.1: The predictive MSE of DLF is plotted against that of EN and PLSR.

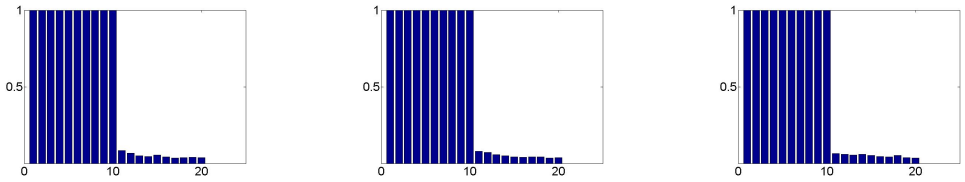


FIGURE 2.2: The averaged inclusion probability of each column of experiment 1, 2 and 3 (from left to right).

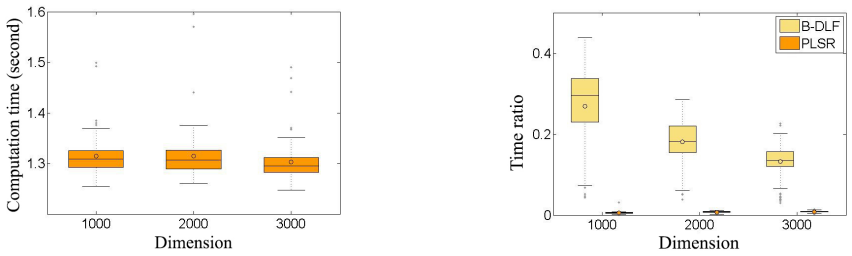


FIGURE 2.3: **Left**: boxplot of the computation time of the Gibbs sampler in B-DLF. **Right**: boxplot of the ratios of the overall computation time of B-DLF and PLSR with respect to EN.

# 3

# Multiscale Mixture of DLF

DLF is restricted to characterizing linear dependence in Gaussian data, while our overarching focus is on non-linear dependence for non-Gaussian data. We address this problem using a multiscale mixture of DLFs, with the coarsest scale characterizing the data as Gaussian distributed about a lower-dimensional hyperplane (this can be accomplished with a single DLF), and finer scales introducing additional Gaussian components about corresponding low-dimensional subspaces. To extend DLF in this manner, we need an appropriate multiresolution dictionary, as well as a way to weight across the different scales adaptively. We learn the dictionary in a first stage, while taking a Bayesian approach to learn a posterior distribution for the weights as well as the scaling parameters in each component at each scale.

## 3.1 Formulation

Borrowing the notations of Allard et al. (2012), $y_i$, for all $i = 1, 2, \ldots, n$, are assumed to have support on $(\mathcal{M}, \mathcal{F}, \mu)$, where $\mathcal{M} \subset \Re^D$, $\mathcal{F}$ is a $\sigma$-field defined on $\mathcal{M}$ and $\mu$ is a probability measure defined on $\mathcal{F}$.

For simplicity, we pick a binary decomposition of the metric space $(\mathcal{M}, \mathcal{F}, \mu)$ as the

multiscale structure of our model. Letting $s = 0, \ldots, \infty$ denote the scale index, $h = 1, \ldots, 2^s$ denote the node index within scale $s$ and $B_r^{\mathcal{M}}(y)$ denote the $\mathcal{F}$-ball inside $\mathcal{M}$ of radius $r > 0$ centered at $y \in \mathcal{M}$, the tree decomposition is defined as follows.

**Definition 2.** *(Allard et al., 2012) A **binary tree decomposition** of a $m$-dimensional metric measure space $(\mathcal{M}, \mathcal{F}, \mu)$ is a family of open sets in $\mathcal{M}$, $\{Cell_{s,h}\}$, called dyadic cells, such that*

*1. for every $s \in \mathcal{Z}$, $\mu(\mathcal{M} \setminus \bigcup_{h=1}^{2^s} Cell_{s,h}) = 0$;*

*2. for $s \leqslant s'$ and $1 \leqslant h' \leqslant 2^{s'}$, either $Cell_{s',h'} \subseteq Cell_{s,h}$ or $\mu(Cell_{s',h'} \cap Cell_{s,h}) = 0$;*

*3. for $s < s'$ and $1 \leqslant h' \leqslant 2^{s'}$, there exists a unique $h = 1, 2, \ldots, 2^s$ such that $Cell_{s',h'} \subseteq Cell_{s,h}$;*

*4. each $Cell_{s,h}$ contains a point $c_{s,h}$ such that $B^{\mathcal{M}}_{c_1 2^{-s}}(c_{s,h}) \subseteq Cell_{s,h} \subseteq B^{\mathcal{M}}_{2^{-s}}(c_{s,h})$, for a constant $c_1$ depending on intrinsic geometric properties of $\mathcal{M}$. In particular, we have $\mu(Cell_{s,h}) \sim 2^{-ds}$.*

A simple 4 level binary tree decomposition of a parabola is visualized in Figure 3.1.

Suppose we have a set of multiscale dictionaries $\{\mu_{s,h}, \Phi_{s,h}\}$, for all $s = 0, 1, \ldots, \infty$ and for all $h = 1, 2, \ldots, 2^s$. The multiscale mixture of DLF (MDLF) model is given by

$$f(y_i) = \sum_{s=0}^{\infty} \sum_{h=1}^{2^s} \pi_{s,h} N_D(y_i; \mu_{s,h}, \Phi_{s,h}\Sigma_{s,h}\Phi_{s,h} + \sigma_s^2 I_D). \tag{3.1}$$

We are interested in the case where the support $(\mathcal{M}, \mathcal{F}, \mu)$ is a compact Riemannian manifold of dimension $p$ isometrically embedded in $\Re^D$. Each local DLF model should provide a local linear approximation to the subspace or manifold the data are concentrated near. Hence, the multiscale dictionaries should reflect local geometric information. With this motivation and to minimize computation time, we use GMRA. Our implementation of GMRA can be summarized as follows.

1. Obtain a binary tree decomposition, $Cell_{s,h}$ for $s = 0, \ldots, \infty$ and $h = 1, \ldots, 2^s$

**Algorithm 2** Multiscale allocation and weights updating within a single iteration

simulate $u_i|y_i, s_i \sim U(0, \pi_{s_i})$
**for all** scale $s$ **do**
  **if** $\pi_s > u_i$ **then**
    **for** $h = 1, 2, \ldots, 2^s$ **do**
      $pr(h_i = h|u_i, y_i, s_i) \propto$
      $\tilde{\pi}_{s_i,h} N_D(y_i; \Phi_{s_i,h}\Sigma_{s_i,h}\Phi_{s_i,h} + \sigma^2_{s_i} I_D)$
    **end for**
    $pr(s_i = s|u_i, y_i) \propto$
    $\sum_{h=1}^{2^s} \tilde{\pi}_{s,h} N_D(y_i; \Phi_{s,h}\Sigma_{s,h}\Phi_{s,h} + \sigma^2_s I_D)$
  **else**
    $pr(s_i = s|u_i, y_i) = 0$
  **end if**
**end for**
sample $s_i$ with probability $pr(s_i = s|u_i, y_i)$
sample $h_i$ with probability $pr(h_i = h|u_i, y_i, s_i)$
**for all** $s$ and $h$ **do**
  sample $S_{s,h} \sim Be(1 + n_{s,h}, a_S + v_{s,h} - n_{s,h})$
  sample $R_{s,h} \sim Be(b_R + r_{s,h}, b_R + v_{s,h} - n_{s,h} - r_{s,h})$
**end for**

using METIS. Note that the proximity matrix needed for graph partition is computed using ANN algorithm.

2. A $d$-dimensional affine approximation in each dyadic cell $Cell_{s,h}$ using fast rank-$d$ SVD algorithm (Rokhlin et al., 2009), yielding $d$-dictionary associated to this cell, denoted $\Phi^d_{s,h}$.

For simplicity, we make $\mu_{s,h}$ the sample mean of $Cell_{s,h}$. The computational cost of the above two steps is

$$O(nD(\log n + p^2)) \tag{3.2}$$

As can be seen, the overall computational cost of dictionary learning is linear in $D$; hence the algorithm is computationally tractable in problems of massive dimension.

Using the multiscale dictionary within $Cell_{s,h}$ for all $s, h$, we can apply the same priors as specified in equation (4) to the scaling matrices $\Sigma_{s,h}$ and residual variance $\sigma^2_s$. Then, all that remains is to choose a prior for the multiscale mixing weights.

This prior should be structured to allow adaptive learning of the appropriate tradeoff between coarse and fine scales. Heavily favoring coarse scales may lead to reduced vari-

ance but also high bias if the coarse scale approximation is not accurate. High weights on fine scales may lead to low bias but high variance due to limited sample size in each fine resolution component. With this motivation, we propose a multiresolution stick-breaking process generalizing usual 'flat' stick-breaking (Sethuraman, 1991). In particular, let

$$S_{s,h} \sim Be(1, a_S),\ R_{s,h} \sim Be(b_R, b_R) \tag{3.3}$$

with $S_{s,h}$ denoting the probability that the observation stops at scale $s$ of a binary tree and $R_{s,h}$ denoting the probability that the observation moves down to the right from node $(s, h)$ conditioning on not stopping at node $(s, h)$. Hence

$$\pi_{s,h} = S_{s,h} \prod_{r<s} (1 - S_{r,g_{shr}}) T_{shr} \tag{3.4}$$

where $g_{shr} = \lceil h/2^{s-r} \rceil$ denotes the ancestors of node $(s, h)$ at scale $r$, $T_{shr} = R_{r,g_{shr}}$ if node $(r+1, g_{sh(r+1)})$ is the right daughter of node$(r+1, g_{shr})$ , otherwise $T_{shr} = 1-R_{r,g_{shr}}$. It can be shown that $\sum_{s=0}^{\infty} \sum_{h=1}^{2^s} \pi_{s,h} = 1$ almost surely for any $a_S, b_R > 0$, where $\pi_{s,h}$ is an infinite sequence of weights defined as in (3.3)-(3.4). This result makes the defined weights a proper set of multiscale mixing weights. The proof is provided in the supplementary materials. As $a_S$ increases, finer scales are favored, resulting in a highly non-Gaussian density.

In practice, it is appealing to approximate model (3.1) by a finite-depth multiscale mixture. Let $L$ denote this depth and let $\{\tilde{\pi}_{s,h}\}_{s \leqslant L}$ denote the truncated weights, which is identical to $\{\pi_{s,h}\}$ except that the stopping probabilities at scale $L$ are set to be equal to one to ensure $\sum_{s=1}^{L} \sum_{h=1}^{2^s} \tilde{\pi}_{s,h} = 1$. The accuracy of the approximation is discussed in the following theorem. The proof is reported in the supplementary material.

**Theorem 3.** *Let*

$$f^L(y) = \sum_{s=1}^{L} \sum_{h=1}^{2^s} \tilde{\pi}_{s,h} N_D(y; \Phi_{s,h} \Sigma_{s,h} \Phi_{s,h} + \sigma_s^2 I_D)$$

*denote the approximation at scale L, let* $P(B) = \int_B f(y)dy$ *and* $P^L(B) = \int_B f^L(y)dy$*, for all* $B \subset \Re^D$ *denote the probability measures corresponding to density* $f(y)$ *and* $f^L(y)$*. Then we have,*

$$d_{TV}(P_L, P) < \left(\frac{a_S}{1+a_S}\right)^L.$$

The above theorem indicates that the approximation error decays at an exponential rate.

## 3.2 Posterior computation

Let $\pi_s = \sum_{h=1}^{2^s} \pi_{s,h}$ denote the total mass assigned at scale $s$, and let $\tilde{\pi}_{s,h} = \pi_{s,h}/\pi_s$. Under this setting we can rewrite model (3.1) as

$$f(y_i) = \sum_{s=0}^{\infty} \pi_s \sum_{h=1}^{2^s} \tilde{\pi}_{s,h} N_D(y_i; \Phi_{s,h}\Sigma_{s,h}\Phi_{s,h}^T + \sigma_s^2 I_D). \tag{3.5}$$

The multiscale allocation is achieved by a multiscale modification of the slice sampler of Kalli et al. (2011). Consider the joint density

$$f(y_i, u_i, s_i) = I_{(u_i < \pi_{s_i})} \sum_{s=0}^{\infty} \pi_{s_i} \sum_{h=1}^{2^{s_i}} \tilde{\pi}_{s_i,h} N_D(y_i; \Phi_{s_i,h}\Sigma_{s_i,h}\Phi_{s_i,h}^T + \sigma_{s_i}^2 I_D). \tag{3.6}$$

The full conditional posterior distributions are

$$u_i | y_i, s_i \sim U(0, \pi_{s_i}) \tag{3.7}$$

$$pr(s_i = s | u_i, y_i) \propto I_{(s:\pi_s > u_i)} \sum_{h=1}^{2^s} \tilde{\pi}_{s,h} N_D(y_i; \Phi_{s,h}\Sigma_{s,h}\Phi_{s,h} + \sigma_s^2 I_D). \tag{3.8}$$

$$pr(h_i = h | u_i, y_i, s_i) \propto \tilde{\pi}_{s_i,h} N_D(y_i; \Phi_{s_i,h}\Sigma_{s_i,h}\Phi_{s_i,h} + \sigma_{s_i}^2 I_D). \tag{3.9}$$

The slice sampler contributes to the computation by allowing the allocation to take place in a subset of all scales of the tree, which can be efficient when we have a deep tree structure.

The algorithm is summarized in Algorithm 2, where $v_{s,h}$ is the number of observations passing through node $(s,h)$, $n_{s,h}$ is the number of observations stopping at node $(s,h)$, and $r_{s,h}$ is the number of observations that continue to the right after passing through node $(s,h)$.

Combining all the techniques discussed above, the Bayesian mutiscale mixture of DLF (B-MDLF) algorithm can be summarized as below.

1. Compute a multiscale dictionary $\{\Phi_{s,h}, \mu_{s,h}\}$ using GMRA.
2. Initialize scaling parameters $\{\alpha^2_{m,s,h}\}$, idiosyncratic variance $\{\sigma^2_s\}$, multiscale weights $\{\pi_{s,h}\}$.
3. Allocate observations and update weights $\{\pi_{s,h}\}$ using Algorithm 2.
4. Update $u_{m,s,h}$ for all $m$, $s$ and $h$ according to
$$\text{Ga}_{(0,1)}\left(\prod_{k=1}^{m}\tau_k^{s,h} + n_{s,h}/2, 1 + \frac{1}{2}\sigma_s^{-2}\sum_{y_i\in C_{s,h}}(Z^{(m)}_{i,s,h})^2\right)$$
5. Update $\tau_m^{s,h}$ for all $m$, $s$ and $h$ according to
$$\text{Exp}_{[1,\infty)}\left(a_\tau - \ln(\prod_{j>m-1}u_{j,s,h})\right)$$
6. Update $\sigma_s^{-2}$ for all $s$ according to
$$\text{Ga}\left(a_\sigma + Dn_s/2, \frac{1}{2}\sum_{y_i\in C_s}\left(A_{i,s,h} - \sum_{j=1}^{d}(1-u_{j,s,h})(Z^{(j)}_{i,s,h})^2\right) + b_\sigma\right)$$
where $C_s$ denotes the set of observations stopping at scale $s$, and $n_s$ denotes the size of $C_s$. The derivation of all the conditional posteriors can be found in the supplement.
7. Adapt $u_{m,s,h}$ using Algorithm 1.
8. Go back to step 3 until the desired iteration number.

## 3.3 Simulation experiments

To assess the performance of the proposed Bayesian MDLF (B-MDLF) model, we conducted a simulation study. We simulated 100 independent samples of size $n = 600$ from a 3-dimensional Swissroll, and then embedded them into a 200-dimensional space by a $200 \times 3$ projection matrix. The projection matrix is randomly picked in each sample. The simulated Swissroll is visualized in Figure 3.2, left panel.

We split the samples into training and test subsets containing 500 and 100 observations respectively. We applied our Gibbs sampler to the training data and used the results to predict a randomly selected dimension given the others. As prior specification we used $a_\sigma = 1/2$, $b_\sigma = 1/2$ and $a = 0.05$, and fixed the upper bound to $d = 10$. A 5-level multi-scale dictionary is used. In both the simulation experiment and the real data analysis, we set the parameters of METIS as suggested by Allard et al. (2012). Our model is compared with RF. The B-MDLF has a consistently better predictive performance than RF as can be seen from the left of Figure 3.3 showing the predictive MSE of each method.

The adaptation performance is also assessed by computing the average inclusion probability of all $d$ dimensions of the dictionary. To be specific, let $\mathcal{R}^t_{s,h}$ denotes the set of retained column indices of node $(s, h)$ at the $t$th iteration, and let $(s^t_i, h^t_i)$ denote the node index of the $i$th observation at the $t$th iteration. Then the inclusion probability of dimension $j = 1, 2, \ldots, 10$ is given by

$$p_j^{inclu} = \frac{1}{n_{adapt} \times N} \sum_{t:\ adapt} \sum_{i=1}^{N} I_{(j \in \mathcal{R}^t_{s_i^t, h_i^t})} \tag{3.10}$$

where $n_{adapt}$ denotes the number of adaptation steps during the MCMC collection interval. The average inclusion probability is easily $> 0.5$ for the first three dimensions, while being much less ($\approx 0.1$) for the remaining dimensions. This suggests good performance in estimating the intrinsic dimension.

To illustrate ability of B-MDLF to learn the nonlinear joint distribution, we randomly

selected one of the 100 simulated data sets, and generated 2,000 samples from the posterior predictive distribution for the data set. To visualize whether these samples were appropriately concentrated near the true manifold, we projected them back to the 3-dimensional subspace using the true projection matrix. The result is shown in Figure 3.2; clearly the samples are appropriately concentrated within only slight additional noise.

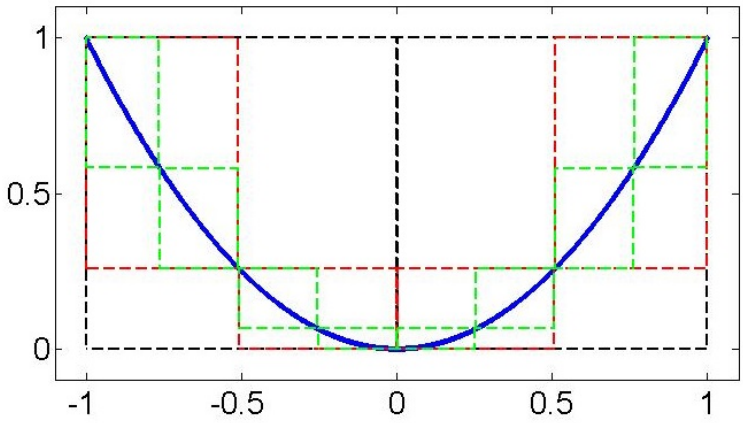


FIGURE 3.1: A 4 level binary tree decomposition of a parabola using METIS, with the black rectangular denoting the second level cells, the red denoting the third level cells and the green denoting the leaf cells.

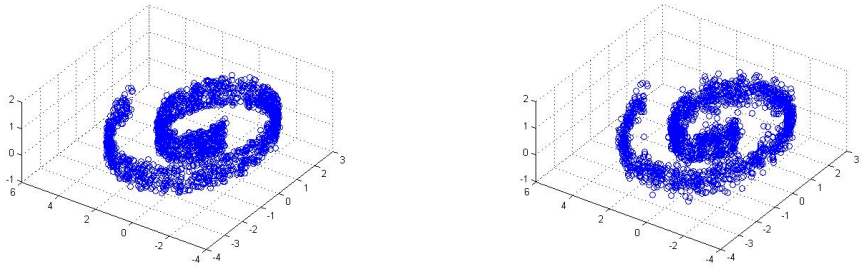

FIGURE 3.2: **Left**: the 3-D scatter plot of the observations in the true linear subspace. **Right**: the 3-D scatter plot of the generations from the learned density projected onto the true linear subspace.

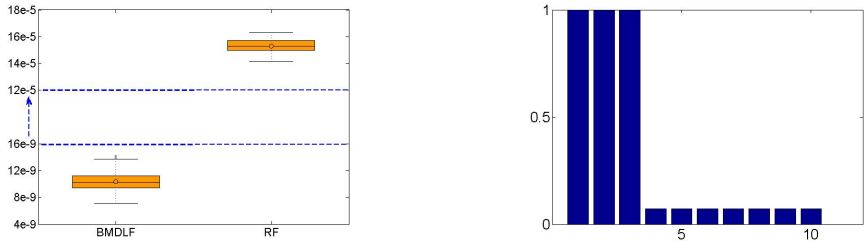


FIGURE 3.3: **Left**: Boxplot of the predictive MSE of B-MDLF and random forest. **Right**: The averaged inclusion probability of each column.

# 4

# Image inpainting

The Frey faces data (Roweis et al., 2002) are 1965 $20 \times 28$ video frames of a single face with different expressions. We randomly split these into 1500 training data and 465 testing data. We applied our Gibbs sampler to the training data, and used the results to reconstruct (predict) the missing top half of the testing data given the bottom half. As prior specification we used $a_\sigma = 1/2$, $b_\sigma = 1/2$ and $a = 0.05$, and fixed the upper bound to $d = 10$. A $6$-level multiscale dictionary is used.

The reconstruction results are compared with those obtained with the Bayesian deep sparse graphical model of Adams et al. (2010) in Figure 4.1. Our B-MDLF outperforms their approach both in terms of prediction and computational efficiency. Indeed, our B-MDLF takes less than 10 minutes while the time reported by Adams et al. (2010) is several hours.

FIGURE 4.1: **Left panel**: results of B-MDLF, with the first row showing original images, second row showing images with the bottom half pixels missing, and the third row showing the reconstructed images. **Right panel**: results shown by Adams et al. (2010), with the left column being the original image, and the right column being the reconstruction from the images with the bottom half missing.

# 5

# Discussion

## 5.1 Summary

In many applications, high-dimensional data with unknown joint distribution are collected. Despite the dramatic importance of learning the joint distribution of such data, few probabilistic methods that scale well to high-dimension and provide good characterization of uncertainty are available. Bayesian nonparametric methods based on mixtures of multivariate Gaussian kernels are widely used, but face major bottlenecks in scaling to higher dimensions. To tackle this problem, we proposed an empirical Bayes density estimator combining manifold learning and Bayesian nonparametric density estimation. One of the building blocks of our model focus on single Gaussian factor decomposition in which variables are linearly related, showing excellent performance in scaling computationally and in generalization error, while providing a valid characterization of uncertainty in predictions. The multiscale mixture generalizations to accommodate unknown density, nonlinear relationships and nonlinear subspaces also had excellent performance in inferring the subspace dimension, estimating the subspace, and characterizing the joint density of the data in the ambient space. The proposed methods are broadly applicable to many learning problems

including regression or classification with missing features.

## 5.2 Remaining problems and future directions

The multi-scale dictionaries learning in the first stage makes the proposed Bayesian density estimator computationally tractable, but it certainly introduces other problems. First of all, it remains unclear about the ability of empirical Bayesian methods in preserving uncertainty compared to fully Bayesian methods. It would be useful to find a more general insight and theory on when empirical Bayes approaches can work well in big data settings. Second, the performance of the proposed method relies heavily on the quality of GMRA. This could be problematic when we have missing features since GMRA relies on complete data, moreover, it becomes challenging when we have streaming data with a changing structure. Future research can potentially address this by designing a mechanism which allows the dictionaries to evolve as the missing data are imputed by the Bayesian paradigm or as new data come in.

Furthermore, the proposed model assumes the data to be continuous, while we might meet categorical data or mixed data or even more structured data, such as curves, objects, graphs in practice. Hence another possible future direction is to generalize the empirical Bayesian idea to adapt more general data types.

# Appendix A

## Proof of the main results

**Theorem** 2

*Proof.* Let $\triangle_d = \Omega - \Phi^d_{SVD}\Sigma_d(\Phi^d_{SVD})^T$. Clearly, $d_\infty(\Omega, \Omega_H) = \max_{1\leqslant r,s\leqslant D} |a^d_{rs}|$, where $a^d_{rs}$ is the $r\ s$ th entry of $\triangle_d$ so that $a^d_{rs} = \sum_{h=d+1}^{D} \alpha_h^2\phi_{rh}\phi_{sh}$. By Cauchy-Schwartz inequality,

$$\left| \sum_{h=H+1}^{D} \alpha_h^2\phi_{rh}\phi_{sh} \right| \leqslant \max_{1\leqslant j\leqslant D} \left( \sum_{h=H+1}^{D} \alpha_h^2\phi_{jh}^2 \right).$$

Since $\Phi^D_{SVD}$ is orthonormal, we have $\phi^2_{rh} \leqslant 1$ for all $r$ and $h$. Hence

$$d_\infty(\Omega, \Omega_d) \leqslant \sum_{h=d+1}^{D} \alpha_h^2.$$

Now for a fixed $\epsilon > 0$, by Chebyshev's inequalities

$$
\begin{aligned}
p\{d_\infty(\Omega,\Omega_d)\leqslant\epsilon\} &\geqslant p\Big\{\sum_{h=d+1}^{D}\alpha_h^2\leqslant\epsilon\Big\}\\
&= E\Big\{p\big(\sum_{h=d+1}^{D}\alpha_h^2\leqslant\epsilon|\tau\big)\Big\}\\
&= 1-E\Big\{p\big(\sum_{h=d+1}^{D}\alpha_h^2>\epsilon|\tau\big)\Big\}\\
&\geqslant 1-E\Big\{\frac{E\big(\sum_{h=d+1}^{D}\alpha_h^2|\tau\big)}{\epsilon}\Big\}.
\end{aligned}
$$

By design we have $u_h \sim \text{Ga}_{(0,1)}(\prod_{t=1}^{h}\tau_t+1,1)$ and $\{u_h\}$ and $\sigma^2$ are conditionally independent, hence

$$
E\Big[\big(\frac{1}{u_h}-1\big)\sigma^2|\tau\Big] = E\Big[\big(\frac{1}{u_h}-1\big)|\tau\Big]E(\sigma^2).
$$

Let $A=\prod_{k=1}^{h}\tau_k$, we have

$$
\begin{aligned}
E\Big[\big(\frac{1}{u_h}-1\big)|\tau\Big] &= \frac{\int_0^1(1/u_h-1)\frac{u_h^A}{\Gamma(A+1)}e^{-u_h}du_h}{\int_0^1\frac{u_h^A}{\Gamma(A+1)}e^{-u_h}du_h}\\
&= \frac{\int_0^1 1/u_h\times u_h^A e^{-u_h}du_h}{\int_0^1 u_h^A e^{-u_h}du_h}-1\\
&= \frac{\int_0^1 u_h^{A-1}e^{-u_h}du_h}{\int_0^1 u_h^A e^{-u_h}du_h}-1\\
&= \frac{\frac{1}{A}u_h^A e^{-u_h}|_0^1+\int_0^1\frac{1}{A}u_h^A e^{-u_h}du_h}{\int_0^1 u_h^A e^{-u_h}du_h}-1\\
&= \frac{e^{-1}}{A\int_0^1 u_h^A e^{-u_h}du_h}-1+\frac{1}{A}.
\end{aligned}
$$

Note that,

$$
\begin{aligned}
A\int_0^1 u_h^A e^{-u_h}du_h &= \frac{A}{A+1}u_h^{A+1}e^{-u_h}|_0^1 + \int_0^1 \frac{A}{A+1}u_h^{A+1}e^{-u_h}du_h \\
&= \frac{A}{A+1}e^{-1} + \frac{A}{A+1}\int_0^1 u_h^{A+1}e^{-u_h}du_h \\
&= \frac{A}{A+1}e^{-1} + \frac{A}{A+1}(\frac{1}{A+2}e^{-1} + \frac{1}{A+2}\int_0^1 u_h^{A+2}e^{-u_h}du_h) \\
&\vdots \\
&= \lim_{K=\infty}\Bigg\{\sum_{k=1}^{K}\frac{\Gamma(A+1)^2}{\Gamma(A)\Gamma(A+k+1)}e^{-1} \\
&\quad + A\Gamma(A+1)F(1;A+K,1)\Bigg\} \\
&= \sum_{k=1}^{\infty}\frac{\Gamma(A+1)^2}{\Gamma(A)\Gamma(A+k+1)}e^{-1} \\
&= \sum_{k=1}^{\infty}\frac{A}{(A+1)(A+2)\ldots(A+k)}e^{-1}
\end{aligned}
$$

where $F(x;a,b)$ is the cdf of $Ga(a,b)$ and $\lim_{a=\infty}F(1;a,1)=0$. Furthermore we have

$$
\begin{aligned}
\sum_{k=1}^{\infty}\frac{\Gamma(A+1)^2}{\Gamma(A)\Gamma(A+k+1)} &= \sum_{k=1}^{\infty}\frac{A}{(A+1)(A+2)\ldots(A+k)} \\
&= \frac{A}{A+1} + \cdots + \frac{A}{(A+1)\ldots(A+k)} + \ldots \\
&\geqslant \frac{A}{A+1} \\
&\geqslant \frac{1}{2}
\end{aligned}
$$

and

$$
\begin{aligned}
1-\sum_{k=1}^{\infty}\frac{\Gamma(A+1)^2}{\Gamma(A)\Gamma(A+k+1)} &= 1-\frac{A}{A+1}-\dots \\
&\leqslant 1-\frac{A}{A+1} \\
&= \frac{1}{A+1} \\
&\leqslant \frac{1}{A}
\end{aligned}
$$

thus we have

$$
\begin{aligned}
\frac{e^{-1}}{A\int_0^1 u_h^A e^{-u_h}du_h}-1+\frac{1}{A} &= \frac{1}{\sum_{k=1}^{\infty}\frac{\Gamma(A+1)^2}{\Gamma(A)\Gamma(A+k+1)}}-1+\frac{1}{A} \\
&= \frac{1-\sum_{k=1}^{\infty}\frac{\Gamma(A+1)^2}{\Gamma(A)\Gamma(A+k+1)}}{\sum_{k=1}^{\infty}\frac{\Gamma(A+1)^2}{\Gamma(A)\Gamma(A+k+1)}}+\frac{1}{A} \\
&\leqslant \frac{1/A}{1/2}+\frac{1}{A} \\
&= \frac{3}{A}
\end{aligned}
$$

Thus $E[(\frac{1}{u_h}-1)|\tau] \leqslant 3/(\prod_{k=1}^{h}\tau_k)$. Based on this inequality, we have

$$
\begin{aligned}
\sum_{h=d+1}^{D} E\Big\{E[(\frac{1}{u_h}-1)\sigma^2|\tau]\Big\} &\leqslant \sum_{h=d+1}^{D} E\bigg(\frac{3}{\prod_{k=1}^{h}\tau_k}\bigg)E(\sigma^2) \\
&= \sum_{h=d+1}^{D} 3ba^h \\
&\leqslant \frac{3ba^d}{1-a}
\end{aligned}
$$

where $b = E(\sigma^2)$ and a = $E(\frac{1}{\tau_1})$. Note that $\tau_h \sim \text{Exp}_{[1,\infty)}(\lambda)$, thus $a < 1$. By Fubini's theorem, $E\{E(\sum_{h=H+1}^{\infty} \alpha_h^2|\tau)\} = \sum_{h=d+1}^{\infty} E\{E[(\frac{1}{u_h} - 1)\sigma^2|\tau]\}$ can be ensured. Now use the inequality $(1 - x/2) > exp(-x)$ if $0 < x \leqslant 1.5$ to get

$$p\{d_\infty(\Omega, \Omega_d) \leqslant \epsilon\} \geqslant \exp\{\frac{-6ba^d}{\epsilon(1-a)}\}$$

if $d > 2log\{b/\epsilon(1-a)\}/log(1/a)$. Hence,

$$p\{d_\infty(\Omega, \Omega_d) > \epsilon\} \leqslant 1 - \exp\{\frac{-6ba^d}{\epsilon(1-a)}\} \leqslant \frac{6ba^d}{\epsilon(1-a)}$$

since $6ba^d/\{\epsilon(1-a)\} < 1$. □

**Theorem** 3

*Proof.* The total variation distance

$$\begin{aligned}
d_{TV}(P_L, P) &= \sup_{B \in \Re^D} |P^L(B) - P(B)| \\
&= \sup_{B \in \Re^D} |\sum_{h=1}^{2^L} \tilde{\pi}_{s,h} N(B; \mu_{s,h}, \Phi_{s,h}\Sigma_{s,h}\Phi_{s,h}^T) - ... \\
&\quad \sum_{s=L}^{\infty} \sum_{h=1}^{2^s} \pi_{s,h} N(B; \mu_{s,h}, \Phi_{s,h}\Sigma_{s,h}\Phi_{s,h}^T)| \\
&\leqslant \max\{\sum_{h=1}^{2^L} \tilde{\pi}_{s,h}, \sum_{s=L}^{\infty} \sum_{h=1}^{2^s} \pi_{s,h}\} \\
&= \max\left\{2^L (\frac{a_S}{1+a_S})^{L-1} \frac{1}{1+a_S} 2^{-L}, \sum_{s=L}^{\infty} 2^s \frac{1}{1+a_S} (\frac{a_S}{2+2a_S})^s\right\}
\end{aligned}$$

$$
\begin{aligned}
d_{TV}(P_L, P) &\leqslant \sum_{s=L}^{\infty} \frac{1}{1+a_S} \left(\frac{a_S}{1+a_S}\right)^s \\
&= \left(\frac{a_S}{1+a_S}\right)^L
\end{aligned}
$$

□

**Theorem 4.** *Letting $\pi_{s,h}$ be an infinite sequence of weights defined as in (7)-(8), then*

$$\sum_{s=0}^{\infty} \sum_{h=1}^{2^s} \pi_{s,h} = 1$$

almost surely for any $a_S$, $b_R > 0$.

*Proof.* For finite $N$ define $\triangle_N = 1 - \sum_{s=0}^{N} \sum_{h=1}^{2^s} \pi_{s,h}$, for which the following inequality holds:

$$\triangle_N = \sum_{h=1}^{N} \prod_{r \leqslant N} \big(1 - S_{r,g_{Nhr}}\big) T_{r-1,g_{Nhr}} \leqslant 2^N \max_{h=1,\ldots,2^N} \prod_{r \leqslant N} \big(1 - S_{r,g_{Nhr}}\big) T_{r-1,g_{Nhr}}. \quad \text{(A.1)}$$

To establish (9), it is sufficient to take the limit of $\triangle_N$ for $N \to \infty$ and show that it converges to 0 a.s. To this end, take the logarithm of the right hand side of (A.1),

$$\log(\triangle_N) \leqslant \max_{h=1,\ldots,2^N} \prod_{r \leqslant N} \log \left\{ 2^N \big(1 - S_{r,g_{Nhr}}\big) T_{r-1,g_{Nhr}} \right\}, \quad \text{(A.2)}$$

and notice that for each $h = 1, \ldots, 2^N$ we have

$$E\left\{ 2^N \big(1 - S_{r,g_{Nhr}}\big) T_{r-1,g_{Nhr}} \right\} = 2^N \frac{a}{a+1} \frac{1}{2^N} = \frac{a}{a+1} \quad \text{(A.3)}$$

Therefore taking $N \to \infty$, by Kolmogorov's three series theorem and Jensen's inequality, the argument of the maximum of (A.2), converges to $-\infty$ a.s. for each $h$. Thus $\triangle_N$ converges to 0 a.s. which concludes the proof. □

# Appendix B

## Posterior derivation

The derivation of conditional posterior of $\sigma_s^{-2}$ is given by

$$
\begin{aligned}
p(\sigma_s^{-2}|-) &\propto (\sigma_s^{-2})^{a_\sigma - 1} \exp(-b_\sigma \sigma_s^{-2}) \prod_{y_i \in C_s} (\sigma_s^2)^{-D/2} \\
&\quad \exp\left\{ -\frac{1}{2}\sigma_s^{-2}\big(A_{i,s,h} - \sum_{j=1}^{d}(1-u_{j,s,h})(Z_{i,s,h}^{(j)})^2\big) \right\} \\
&\propto (\sigma_s^{-2})^{Dn_s/2 + a_\sigma - 1} \\
&\quad \exp\left\{ -\sigma_s^{-2}\big[\frac{1}{2}\sum_{y_i \in C_s}(A_{i,s,h} - \sum_{j=1}^{d}(1-u_{j,s,h})(Z_{i,s,h}^{(j)})^2) + b_\sigma\big] \right\}
\end{aligned}
$$

The derivation of conditional posterior of $u_{m,s,h}$ is given by

$$
\begin{aligned}
p(u_{m,s,h}|-) &\propto \prod_{y_i \in C_{s,h}} u_{m,s,h}^{1/2} \exp\left\{ -\frac{1}{2}\sigma_s^{-2} u_{m,s,h} (Z_{i,s,h}^{(m)})^2 \right\} \\
&\quad u_{m,s,h}^{\prod_{j=1}^{m} \tau_j^{s,h} - 1} \exp\{-u_{m,s,h}\} I_{(0,1)} \\
&\propto u_{m,s,h}^{\prod_{j=1}^{m} \tau_j^{s,h} + n_{s,h}/2 - 1} \\
&\quad \exp\left\{ -\Big[1 + \frac{1}{2}\sigma_s^{-2} \sum_{y_i \in C_{s,h}} (Z_{i,s,h}^{(m)})^2\Big] u_{m,s,h} \right\} I_{(0,1)}
\end{aligned}
$$

The derivation of conditional posterior of $\tau_m^{s,h}$ is given by

$$
\begin{aligned}
p(\tau_m^{s,h}|-) &\propto \Big( \prod_{j>m-1} u_{j,s,h} \Big)^{\tau_j^{s,h}} \exp\{-a_\tau \tau_m^{s,h}\} I_{[1,\infty)} \\
&\propto \exp\left\{ -\Big[a_\tau - ln\Big( \prod_{j>m-1} u_{j,s,h} \Big)\Big] \tau_m^{s,h} \right\}
\end{aligned}
$$

# Bibliography


Adams, R. P., Wallach, H. M., and Ghahramani, Z. (2010), "Learning the Structure of Deep Sparse Graphical Models," *Journal of Machine Learning Research: Workshop and Conference Proceedings (AISTATS)*, 9, 1–8.

Allard, W. K., Chen, G., and Maggioni, M. (2012), "Multi-scale geometric methods for data sets II: Geometric multi-resolution analysis," *Applied and Computational Harmonic Analysis*, 32, 435–462.

Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R., and Wu, A. Y. (1998), "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," *Journal of the ACM (JACM)*, 45, 891–923.

Bedford, T. and Cooke, R. M. (2002), "Vines–a new graphical model for dependent random variables," *The Annals of Statistics*, 30, 1031–1068.

Bengio, Y. and Vincent, P. (2004), "Manifold parzen windows," Tech. rep., CIRANO.

Bhattacharya, A. and Dunson, D. B. (2011), "Sparse Bayesian infinite factor models," *Biometrika*, 98, 291–306.

Carvalho, C. M., Chang, J., Lucas, J. E., Nevins, J. R., Wang, Q., and West, M. (2008), "High-dimensional sparse factor modeling: applications in gene expression genomics," *Journal of the American Statistical Association*, 103.

Escobar, M. D. and West, M. (1995), "Bayesian density estimation and inference using mixtures," *Journal of the American Statistical Association*, 90, 577–588.

Fokoué, E. and Titterington, D. M. (2003), "Mixtures of factor analysers. Bayesian estimation and inference by stochastic simulation," *Machine Learning*, 50, 73–94.

Ghahramani, Z. and Beal, M. J. (1999), "Variational inference for Bayesian mixtures of factor analysers." in *NIPS*, pp. 449–455.

Ghahramani, Z., Hinton, G. E., et al. (1996), "The EM algorithm for mixtures of factor analyzers," Tech. rep., CRG-TR-96-1, University of Toronto.

Joe, H. (2005), "Asymptotic efficiency of the two-stage estimation method for copula-based models," *Journal of Multivariate Analysis*, 94, 401–419.

Jordan, M. I. (2004), "Graphical models," *Statistical Science*, pp. 140–155.

Kalli, M., Griffin, J. E., and Walker, S. G. (2011), "Slice sampling mixture models," *Statistics and Computing*, 21, 93–105.

Karypis, G. and Kumar, V. (1998), "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing*, 20, 359–392.

Lawrence, N. (2005), "Probabilistic non-linear principal component analysis with Gaussian process latent variable models," *The Journal of Machine Learning Research*, 6, 1783–1816.

Liu, H., Lafferty, J. D., and Wasserman, L. A. (2007), "Sparse nonparametric density estimation in high dimensions using the rodeo," in *International Conference on Artificial Intelligence and Statistics*, pp. 283–290.

Lopez-Paz, D., Lobato, J. M. H., and Ghahramani, Z. (2013), "Gaussian process vine copulas for multivariate dependence," .

Müller, P., Erkanli, A., and West, M. (1996), "Bayesian curve fitting using multivariate normal mixtures," *Biometrika*, 83, 67–79.

Rasmussen, C. E. (1999), "The infinite Gaussian mixture model." in *NIPS*, vol. 12, pp. 554–560.

Rokhlin, V., Szlam, A., and Tygert, M. (2009), "A randomized algorithm for principal component analysis," *SIAM Journal on Matrix Analysis and Applications*, 31, 1100–1124.

Roweis, S. T. and Saul, L. K. (2000), "Nonlinear dimensionality reduction by locally linear embedding," *Science*, 290, 2323–2326.

Roweis, S. T., Saul, L. K., and Hinton, G. E. (2002), "Global coordination of local linear models," *Advances in Neural Information Processing Systems*, 2, 889–896.

Sethuraman, J. (1991), "A constructive definition of Dirichlet priors," Tech. rep., DTIC Document.

Shen, W., Tokdar, S. T., and Ghosal, S. (2013), "Adaptive Bayesian multivariate density estimation with Dirichlet mixtures," *Biometrika*, 100, 623–640.

Tenenbaum, J. B., Silva, V. D., and Langford, J. C. (2000), "A global geometric framework for nonlinear dimensionality reduction," *Science*, 290, 2319–2323.

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

Wang, B. and Titterington, M. (2004), "Inadequacy of interval estimates corresponding to variational Bayesian approximations," .