Scalable geometric density estimation

Ye Wang
Dept. of Statistics
Duke University

Antonio Canale
Dept. of Economics & Statistics
and Collegio Carlo Alberto
University of Turin

David Dunson
Dept. of Statistics
Duke University

Abstract

It is standard to assume a low-dimensional structure in estimating a high-dimensional density. However, popular methods, such as probabilistic principal component analysis, scale poorly computationally. We introduce a novel empirical Bayes method that we term geometric density estimation (GEODE) and show that, with mild conditions and among all d-dimensional linear subspaces, the span of the d leading principal axes of the data maximizes the model posterior. With these axes pre-computed using fast singular value decomposition, GEODE easily scales to high dimensional problems while providing uncertainty characterization. The model is also capable of imputing missing data and dynamically deleting redundant dimensions. Finally, we generalize GEODE by mixing it across a dyadic clustering tree. Both simulation studies and real world data applications show superior performance of GEODE in terms of robustness and computational efficiency.

1 Introduction

Let $\boldsymbol{y}_i \in \Re^D$, 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. Moreover, careful tuning of bandwidth is needed, since the choice of bandwidth fundamentally impacts performance (Liu et al., 2007). Bayesian nonparametric models (Escobar and West, 1995; Rasmussen, 1999) 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 combat the curse of dimensionality, it is popular to assume that the data concentrate near a low-dimensional linear subspace. Principal component analysis (PCA) is a ubiquitous technique building upon such assumption. Tipping and Bishop (1999b) generalized PCA within a density estimation framework and introduced probabilistic PCA (PPCA). PPCA can be viewed as a special case of the factor analyzer model (FA), which does not assume isotropic error. Carvalho et al. (2008) and Bhattacharya and Dunson (2011) (among many others) have successfully applied FA under the Bayesian paradigm while additionally assuming sparsity. However, FA involves complex computation that does not scale well. PPCA, on the other hand, can be fitted via the expectation maximization algorithm (EM), which is computationally cheaper especially when D is large (Roweis, 1998). EM also offers a straightforward way to accommodate missing data through imputation at each iteration. However, PPCA is not able to scale to massive dimensional problems, and is sensitive to the choice of p. Moreover, the computational cost will explode even in the existence of a tiny proportion of missing data. These computational bottlenecks of PPCA are illustrated in § 4.

Randomized singular value decompositions (SVD) are able to estimate the geometric structure of the data vectors with tiny error at a very small computational cost (Rokhlin et al., 2009). Unfortunately, traditional PPCA cannot utilize this cheaply obtained geometric information. We propose a novel Bayesian model that we term geometric density estimation (GEODE), which leverages on fast SVD algorithms, and hence easily scales to high dimensional problems ($D = 10^6$ in our simulation). We also generalize our model to a

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51.

mixture of GEODE (mGEODE) to account for nonlinear cases via a dyadic clustering tree, and illustrate its performance via real world image data.

The remainder of the paper is organized as follows. We start with a brief review of PPCA and discuss its computational bottlenecks. We then propose GEODE in § 3, and demonstrate its performance via simulation in § 4. mGEODE is proposed in § 5. A detailed discussion on the computational cost is reported in § 6. An image inpainting application is presented in § 7, and a discussion is reported in § 8.

2 PPCA Revisited

Letting $\boldsymbol{C} = \boldsymbol{\Gamma}\boldsymbol{\Gamma}^\top + \sigma^2\boldsymbol{I}$ where $\boldsymbol{\Gamma} \in \Re^{D\times d}$ and $d < D$, the PPCA model can be written as

$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{C}). \tag{1}$$

Letting $\lambda_1, \ldots, \lambda_D$ denote the eigenvalues of the sample covariance matrix ordered descendingly, it can be shown (Tipping and Bishop, 1999b) that the MLE of $\boldsymbol{\Gamma}$ and σ^2 is given as

$$\boldsymbol{\Gamma}_{ML} = \boldsymbol{U}_d(\boldsymbol{\Lambda}_d - \sigma^2\boldsymbol{I})^{1/2}\boldsymbol{R},$$
$$\sigma^2_{ML} = \frac{1}{D-d}\sum_{j=d+1}^{D}\lambda_j,$$

where the column vectors in the $D \times d$ matrix $\boldsymbol{U}_d$ are the principal eigenvectors of the sample covariance matrix, with the corresponding eigenvalues $\lambda_1, \ldots, \lambda_d$ in the $d \times d$ diagonal matrix $\boldsymbol{\Lambda}_d$, and $\boldsymbol{R}$ is an arbitrary $d \times d$ orthogonal rotation matrix.

We use $\boldsymbol{Y}$ to denote the demeaned observation matrix in which each row is a demeaned data vector. According to the above result, one can solve PPCA by applying a SVD on $\boldsymbol{Y}$. Note that depending on whether the sample size N is smaller than D, either all D or all N singular values are needed, with a computational cost of at least $\mathcal{O}(\min\{ND^2, N^2D\})$ even using iterative methods. Iterative methods are likely to be nonrobust if more than a small number of singular values are needed.

Roweis (1998) pointed out that using EM could be computationally cheaper, especially in cases where D is large. Although reported computational cost is $\mathcal{O}(NDd)$, a cost that is comparable to that of our proposed model, the simulation results show that the constant in the cost of EM is much larger. Moreover, it is not clear how the convergence rate of EM varies with respect to D and d, and performance is very sensitive to the choice of d. In practice, d is typically picked by cross validation, whose computational cost is prohibitive when D or N is large.

A fast rank-d SVD (Rokhlin et al., 2009) approximates an exact SVD to its first d leading singular values at a certain error bound with a high probability. The computational cost is $\mathcal{O}(NDd)$. The algorithm is proved to have a better performance when singular values after the dth singular value decay very slowly, which is typically the case in practice. One might ask if fast SVD can be applied to PPCA, since the closed form MLE comes from a SVD on $\boldsymbol{Y}$. Unfortunately, instead of just the first d singular values, one will need all of them in computing σ^2_{ML}. Moreover, since fast rank-d SVD relies on randomization, there is a corresponding approximation error. It is not clear how PPCA performs in the presence of such error.

3 Geometric Density Estimation

In this section, we develop GEODE piece by piece. The correctness of the method is first justified via a theorem, and a shrinkage prior is then specified to facilitate GEODE to automatically identify and delete redundant dimensions. Finally, an efficient Gibbs sampler is designed for posterior computation.

3.1 Model Formulation

Let $\boldsymbol{W}$ be a $D \times d$ matrix with column vectors being orthonormal, $\boldsymbol{\Sigma} = \mathrm{diag}(\alpha_1^2, \ldots, \alpha_p^2)$, $\boldsymbol{C} = \boldsymbol{W}\boldsymbol{\Sigma}\boldsymbol{W}^\top + \sigma^2\boldsymbol{I}$ and $\boldsymbol{S} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{y}_i - \boldsymbol{\mu})(\boldsymbol{y}_i - \boldsymbol{\mu})^\top$. The model is as follows

$$\begin{aligned} \boldsymbol{y} &\sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{C}) \\ \sigma^2 &\sim \mathrm{IG}(a_\sigma, b_\sigma), \end{aligned} \tag{2}$$

where $\mathrm{IG}(a_\sigma, b_\sigma)$ denotes an inverse Gamma distribution with shape parameter a_σ and rate parameter b_σ. The corresponding log-posterior (up to an additive constant) is

$$\mathcal{L} = -\frac{N}{2}\big\{D\ln(2\pi) + \ln|\boldsymbol{C}| + \mathrm{tr}(\boldsymbol{C}^{-1}\boldsymbol{S}) + \frac{2a_\sigma + 2}{N}\ln(\sigma^{-2}) + \frac{2b_\sigma}{N}\sigma^{-2}\big\}.$$

3.2 Empirical Bayes Solution for $\boldsymbol{\mu}$ and $\boldsymbol{W}$

$\boldsymbol{\mu}$ and $\boldsymbol{W}$ carries the geometric information of the data vectors and uniquely define a linear subspace in $\Re^D$, with $\boldsymbol{\mu}$ being the origin. Motivated by computational considerations and from an empirical Bayes perspective, we will first estimate $\boldsymbol{\mu}$ and $\boldsymbol{W}$ relying on a single pass through the data, and then fix them afterwards at these estimated values. In particular, we solve the following optimization problem:

$$(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{W}}) = \arg\max_{\boldsymbol{\mu}, \boldsymbol{W}}\big[\max_{\sigma^2, \boldsymbol{\Sigma}}\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{W}, \sigma^2, \boldsymbol{\Sigma})\big]. \tag{3}$$

The following theorem shows that a closed form solution to (3) exists and can be obtained via a single SVD through the data. The proof is reported in the supplementary material.

Theorem 1. *Let $\lambda_1, \ldots, \lambda_D$ be the eigenvalues of $\boldsymbol{S}$ ordered descendingly, define $e_k = (D-d)\lambda_k - \sum_{j=k}^{k+D-d-1} \lambda_j$ for all $k \leq d+1$ and define $q = \sum_{j=d+1}^{D} \lambda_j - (D-d)\lambda_D$. Suppose*

Condition 1: $d < rank(\boldsymbol{S})$

Condition 2: $(a_\sigma + 1)\lambda_D \leq \frac{N}{2} q$

Condition 3: For all $e_k > 0$, $b_\sigma < \frac{N}{2} e_k$.

Then

$$\hat{\boldsymbol{\mu}} = \bar{\boldsymbol{y}},\ \hat{\boldsymbol{W}} = \boldsymbol{U}_d$$

solves (3), where the column vectors of the $D \times d$ matrix $\boldsymbol{U}_d$ are the d leading right singular vectors of $\boldsymbol{Y}$.

Condition 1 is trivial since it has to be satisfied in the first place in order for the model to make sense. In practice, condition 2 and 3 are easily met when N is large. Theorem 1 shows that the span of the d leading right singular vectors of $\boldsymbol{Y}$ maximizes the posterior of model (2) among all d-dimensional linear subspace in $\Re^D$, regardless of the choice of prior for σ^2.

We term the column vectors of $\hat{\boldsymbol{W}}$ as the *principal axes* of the data and denote them by $\boldsymbol{w}_j$, for $j = 1, \ldots, d$. The theorem yields a practical method for obtaining $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{W}}$, which is summarized as follows.

- $\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{y}_i$.
- Obtain $\boldsymbol{w}_j$, for $j = 1, \ldots, d$, via applying the fast rank-d SVD on $\boldsymbol{Y}$.

This will be the first step in our method. As we do not know the true dimension p of the subspace, we obtain d principal axes, with d chosen to correspond to a conservative upper bound, so that we are confident *a priori* that $d \geq p$. In the sequel, we will define a shrinkage prior on $\boldsymbol{\Sigma}$ that will effectively delete redundant principal axes, and favor the model around the true p. Exploiting this behavior, an adaptive Gibbs sampler will be proposed.

3.3 Prior for $\boldsymbol{\Sigma}$ and Learning of p

For the diagonal elements of $\boldsymbol{\Sigma}$, we could simply fix them at the values that maximize (3), and from the proof of Theorem 1 we know that such values are $\alpha_j^2 = \lambda_j - \sigma^2$, for all $j = 1, \ldots, d$. In this way our model only contains a single unknown σ^2. However this is problematic since our inference would rely heavily on the accuracy of the SVD on $\boldsymbol{Y}$. Moreover, with all except σ^2 fixed *a priori*, the model uncertainty tends to be severely underestimated.

Hence, instead of fixing α_j^2's, we will learn them by giving them a carefully designed prior distribution. This prior distribution also facilitates GEODE to automatically delete redundant principal axes.

Equipped with $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{W}}$ and with another pass through the data, we can obtain for all $i = 1, \ldots, N$ sufficient statistics $A_i = (\boldsymbol{y}_i - \hat{\boldsymbol{\mu}})^\top (\boldsymbol{y}_i - \hat{\boldsymbol{\mu}})$ and $\boldsymbol{Z}_i = \hat{\boldsymbol{W}}^\top (\boldsymbol{y}_i - \hat{\boldsymbol{\mu}})$, with $Z_i^{(j)}$ denoting the jth element of $\boldsymbol{Z}_i$. We then apply a random variable transformation $u_j = (1 + \sigma^{-2}\alpha_j^2)^{-1}$, for $j = 1, \ldots, d$. With basic algebra, the likelihood of GEODE is then

$$f(\boldsymbol{y}_i) \propto (\sigma^2)^{-D/2} \prod_{j=1}^{d} u_j^{1/2} \exp\Big\{ -\frac{1}{2}\sigma^{-2} \times \big[A_i - \sum_{j=1}^{d} (1-u_j)(Z_i^{(j)})^2\big] \Big\}.$$

The derivation is reported in the supplementary material.

Delete redundant principal axes. We rely on the following geometric intuition. It is easy to check that $\boldsymbol{w}_j^\top \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{w}_j^\top \boldsymbol{\mu}, \alpha_j^2 + \sigma^2)$. Hence α_j^2 is the signal variance along the direction $\boldsymbol{w}_j$, which should be decreasing for $j = 1, \ldots, p$ and be zero for $j = p+1, \ldots, d$. This motivates us to penalize α_j^2 by increasingly shrinking towards zero as j increases, which is equivalent to shrinking u_j increasingly for larger j. To accomplish this adaptive shrinkage, we propose a multiplicative exponential process prior that adapts the prior of Bhattacharya and Dunson (2011). Letting $\delta_j = \prod_{k=1}^{j} \tau_k$, the prior is given for $j = 1, \ldots, d$ as follows:

$$u_j \sim \mathrm{Ga}_{(0,1)}(\delta_j + 1, 1)$$
$$\tau_j \sim \mathrm{Exp}_{[1,\infty)}(a_\tau)$$

where $\mathrm{Ga}_{(0,1)}(\delta_j + 1, 1)$ denotes a Gamma distribution with shape parameter $\delta_j + 1$ and rate parameter 1 truncated within $(0, 1)$ and $\mathrm{Exp}_{[1,\infty)}(a_\tau)$ denotes an Exponential distribution with parameter a_τ truncated within $[1, \infty)$. δ_j and τ_j are the global and the local shrinkage parameter for α_j^2, respectively. Since $\tau_j \geq 1$ for $j = 1, \ldots, d$, $\delta_j = \prod_{k=1}^{j} \tau_k$ is increasing with respect to j. As a result, u_j is stochastically approaching one, since the truncated gamma density concentrates around one as δ_j increases. In practice, we fix $a_\sigma = 2$, $b_\sigma = 2$, and $a_\tau = 0.05$ as a default.

3.4 Posterior Computation

To avoid paying a heavy computational price for choosing a conservative upper bound $d \geq p$, we automatically delete redundant principal axes as computation proceeds. To this end, we adopt an adaptive Gibbs sampler related to that developed by Bhattacharya and Dunson (2011). To be specific, we let $\{1, \ldots, d\} = \mathcal{A} \cup \mathcal{R}$, with $\mathcal{A} \cap \mathcal{R} = \varnothing$. The sets $\mathcal{A}$ and $\mathcal{R}$ index the *active* and *removed* axes, respectively. At iteration t of the sampler, with probability $p(t) = \exp(c_0 + c_1 t)$, we refine sets $\mathcal{A}$ and $\mathcal{R}$, where $c_0 = -1$ and $c_1 = 0.005$ are chosen to favor frequent adaptation early in the chain and exponentially fast decay in frequency. In the refinement step, we remove all axes in $\mathcal{A}$ having less than $\text{tol} = 10^{-2}$ impact, and if no such axis exists, move the small element of $\mathcal{R}$ back to $\mathcal{A}$. We will stop the adaptation after a pre-specified stopping time so that the Gibbs sampler will converge to the posterior distribution associated with one selected set of principal axes. Before the stopping time, the Gibbs sampler is jumping around different target distributions and all samples will be thrown away as burn-ins.

The algorithm implementing GEODE can be summarized as follows:

Step 1 (preprocessing): Obtain $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{W}}$ as described in § 3.2 and compute sufficient statistics A_i and $\boldsymbol{Z}_i$ for $i = 1, \ldots, N$.

Step 2 (Gibbs sampler): Set $\mathcal{A} = \{1, \ldots, d\}$ and $\mathcal{R} = \varnothing$. Iterate until obtaining T posterior samples:

1. Update u_j for all $j \in \mathcal{A}$ according to $\text{Ga}_{(0,1)}(\hat{a}_j, \hat{b}_j)$, where $\hat{a}_j = \prod_{k<=j, k\in\mathcal{A}} \tau_k + N/2$ and $\hat{b}_j = 1 + \frac{1}{2}\sigma^{-2} \sum_{i=1}^{n} (\boldsymbol{Z}_i^{(j)})^2$.
2. Update τ_j for all $j \in \mathcal{A}$ according to $\text{Exp}_{[1,\infty)}(\hat{\lambda}_j)$, where $\hat{\lambda}_j = a_\tau - \ln(\prod_{k>j-1, k\in\mathcal{A}} u_k)$
3. Update σ^{-2} according to $\text{Ga}(\hat{c}, \hat{d})$, where $\hat{c} = a_\sigma + DN/2$, $\hat{d} = \frac{1}{2}\sum_{i=1}^{N} \left[A_i - \sum_{j\in\mathcal{A}} (1-u_j)(\boldsymbol{Z}_i^{(j)})^2\right] + b_\sigma$.
4. If after the stopping time, go directly to the next iteration. Otherwise,
   - if before the stopping time, then with probability $1-p(t)$ go directly to the next iteration and otherwise go to step 5.
   - if at the stopping time, move all $j \in \mathcal{A}$ such that $r_j^t = \left(\alpha_j^t\right)^2 / \max_{j\in\mathcal{A}} \left(\alpha_j^t\right)^2 < tol$ from $\mathcal{A}$ to $\mathcal{R}$ and go to next iteration.
5. Move all $j \in \mathcal{A}$ such that $r_j^t = \left(\alpha_j^t\right)^2 / \max_{j\in\mathcal{A}} \left(\alpha_j^t\right)^2 < tol$ from $\mathcal{A}$ to $\mathcal{R}$. If no such j exists, then move the smallest j from $\mathcal{R}$ to $\mathcal{A}$.

The derivation of the conditional posteriors is in the supplementary material. The preprocessing part only involves two passes through the data, with a computational cost linear in D, while the cost of the Gibbs sampler is independent of D. This makes it easy to scale to high dimensional problems. The superior computational performance of GEODE is illustrated in the next section via simulations and a detailed discussion on the computational cost is reported in § 6.

3.5 Missing Data Imputation

Bayesian models can easily utilize partially observed data by probabilistically imputing the missing features based on their conditional posterior distribution. Moreover, prediction can also be viewed as a missing data imputation problem. We propose several scalable missing data strategies for GEODE, and discuss the appropriateness of these strategies in different missing data scenarios.

Notations $\boldsymbol{y}_M$ and $\boldsymbol{y}_O$ are introduced as the missing part and the observed part of $\boldsymbol{y}$ respectively. Let $\boldsymbol{\mu}_M$ and $\boldsymbol{W}_M$ denote the parts of $\boldsymbol{\mu}$ and $\boldsymbol{W}$ corresponding to $\boldsymbol{y}_M$, and let $\boldsymbol{\mu}_O$ and $\boldsymbol{W}_O$ denote the parts corresponding to $\boldsymbol{y}_O$. The following proposition enables efficient sampling from the conditional posterior distribution $p(\boldsymbol{y}_M|\boldsymbol{y}_O, \boldsymbol{\Theta})$, where $\boldsymbol{\Theta}$ denotes all the unknown parameters in the model.

Proposition 1. *Introduce augmented data $\boldsymbol{\eta} \in \Re^d$ such that $(\boldsymbol{y}|\boldsymbol{\eta}, \boldsymbol{\Theta}) \sim \mathcal{N}(\boldsymbol{\mu} + \boldsymbol{W}\boldsymbol{\eta}, \sigma^2\boldsymbol{I})$ and $(\boldsymbol{\eta}|\boldsymbol{\Theta} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$. Then we have the conditional distribution with $\boldsymbol{\eta}$ marginalized out equal $(\boldsymbol{y}|\boldsymbol{\Theta}) \sim \mathcal{N}(\boldsymbol{\mu} + \boldsymbol{W}\boldsymbol{\Sigma}\boldsymbol{W}^\top, \sigma^2\boldsymbol{I})$. Furthermore, we have*

$$\begin{aligned} \boldsymbol{\eta}|\boldsymbol{y}_O, \boldsymbol{\Theta} &\sim \mathcal{N}(\hat{\boldsymbol{\mu}}_\eta, \hat{\boldsymbol{C}}_\eta), \\ \boldsymbol{y}_M|\boldsymbol{\eta}, \boldsymbol{y}_O, \boldsymbol{\Theta} &\sim \mathcal{N}(\boldsymbol{\mu}_M + \boldsymbol{W}_M\boldsymbol{\eta}_i, \sigma^2\boldsymbol{I}), \end{aligned}$$

where $\hat{\boldsymbol{C}}_\eta = \left(\boldsymbol{\Sigma}\boldsymbol{W}_O^\top\boldsymbol{W}_O/\sigma^2 + \boldsymbol{I}\right)^{-1}\boldsymbol{\Sigma}$ and $\hat{\boldsymbol{\mu}}_\eta = \hat{\boldsymbol{C}}_\eta\boldsymbol{W}_O^\top(\boldsymbol{y}_O - \boldsymbol{\mu}_O)/\sigma^2$.

Corollary 1. *For any $\boldsymbol{\Theta}$, the following are true*

$$\begin{aligned} \boldsymbol{y}_O|\boldsymbol{\Theta}, \boldsymbol{\mu}_O, \boldsymbol{W}_O &\sim \mathcal{N}(\boldsymbol{\mu}_O, \boldsymbol{W}_O\boldsymbol{\Sigma}\boldsymbol{W}_O^\top + \sigma^2\boldsymbol{I}), \\ \boldsymbol{y}_M|\boldsymbol{y}_O, \boldsymbol{\Theta}, \boldsymbol{\mu}_O, \boldsymbol{W}_O &\sim \mathcal{N}(\hat{\boldsymbol{\mu}}_M, \hat{\boldsymbol{C}}_M), \end{aligned} \tag{4}$$

where $\hat{\boldsymbol{\mu}}_M = \boldsymbol{\mu}_M + \boldsymbol{W}_M\hat{\boldsymbol{\mu}}_\eta$ and $\hat{\boldsymbol{C}}_M = \boldsymbol{W}_M\hat{\boldsymbol{C}}_\eta\boldsymbol{W}_M^\top + \sigma^2\boldsymbol{I}$.

Proofs are reported in the supplementary material.

Let D_M denote the number of missing features in $\boldsymbol{y}$. Equipped with Proposition 1 and Corollary 1, we propose the following three imputation strategies:

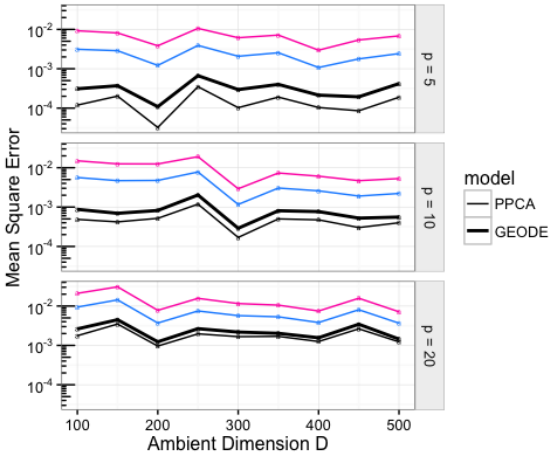

Figure 1: Comparing the MSE of estimating σ^2 under different D and p. Results for PPCA are color coded with black denoting $d = p$, blue denoting $d = p + 5$, and purple denoting $d = p + 10$.

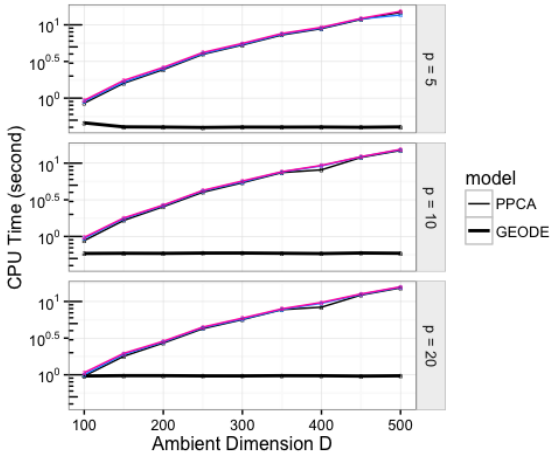

Figure 2: Comparing CPU times fitting the model under different D and p. Results for PPCA are color coded, with black denoting $d = p$, blue denoting $d = p + 5$, and purple denoting $d = p + 10$.

- Small D_M ($\leq d$): sample using (4) is preferred to sampling using Proposition 1 due to numerical concerns. The computational complexity is $\mathcal{O}(d^3)$.
- Moderately large D_M ($> d$): sample via the data augmentation technique provided in Proposition 1. The computational complexity is $\mathcal{O}(d^3 + D_M d)$.
- Large D_M: sample via data augmentation in the first few steps of the Gibbs sampler, and later on fix the value of $\boldsymbol{y}_M$ to its last update. When D_M is large, we cannot afford to run a Gibbs sampler with each step having a complexity linear in D_M.

4 Simulation Studies

In this section, we compare GEODE to its counterpart PPCA in terms of accuracy, robustness and computational efficiency via simulations. All experiments are conducted in Matlab version 2015a on an OS X laptop with a double 3.1 GHz Intel(R) Core(TM) i7 processor. PPCA is fitted using Matlab function `ppca` under the statistics and machine learning toolbox. This function implements an EM algorithm for PPCA, which handles missing data (Roweis, 1998; Ilin and Raiko, 2010). All results reported are obtained by averaging over 10 replicated experiments.

Moderately large D without missing data: In this first simulation study, we let D vary from 100 to 500 and fix N to be 500. We test both methods on three different intrinsic dimensions, i.e., $p \in \{5, 10, 20\}$. To test the robustness of PPCA to the choices of d, we let d take values in $\{p, p + 5, p + 10\}$ while fixing $d = 30$ for GEODE. We evaluate the performance of both methods in terms of mean square error (MSE) in estimating σ^2. The comparison of the two models is presented in Figure 1. The thin black lines denote the MSE of PPCA, with a correct guess of p, i.e., $d = p$. Though they seem to be better than GEODE, the difference decreases as p increases. However, when incorrectly choosing d, which is common in practice, the performance of PPCA (denoted by the thin colored lines) drops dramatically, and is much worse than GEODE.

The corresponding CPU time of fitting both models is reported in Figure 2. The computational cost of PPCA grows fast in D, while the cost of GEODE grows much slower and is dominated by the Gibbs sampler part. This explains why in Figure 2 the cost of GEODE seems not to grow in D. To check how well GEODE dynamically delete the redundant principal axes, for $j = 1, \ldots, d$, we calculate the average proportion of j's being inside $\mathcal{R}$ within all iterations where adaptation takes place. These proportions are visualized in Figure 3. It can be easily seen that GEODE is able to quickly identify the redundant axes. Hence, we can conclude from the simulations that GEODE performs almost as well but slightly worse than the best PPCA can achieve, but with a much smaller computational cost. Moreover, GEODE automatically select p starting from any crude guess d, while the performance of PPCA is highly sensitive to the choice of d.

Moderately large D with missing data: Though the EM algorithm of PPCA offers a straightforward way to impute missing data, even a very small proportion of missingness explodes its computational cost. In this simulation study, we fix $N = 500$, and randomly select

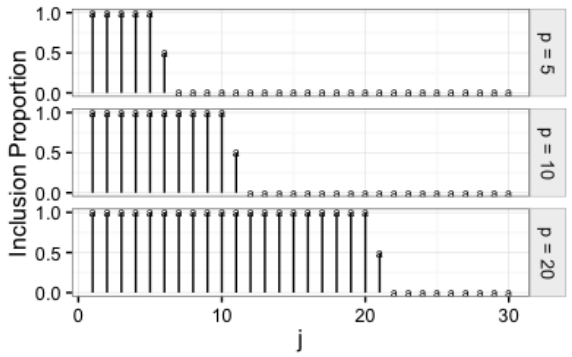

Figure 3: Proportion of inclusion for j within all adaptations averaged across all replicates.

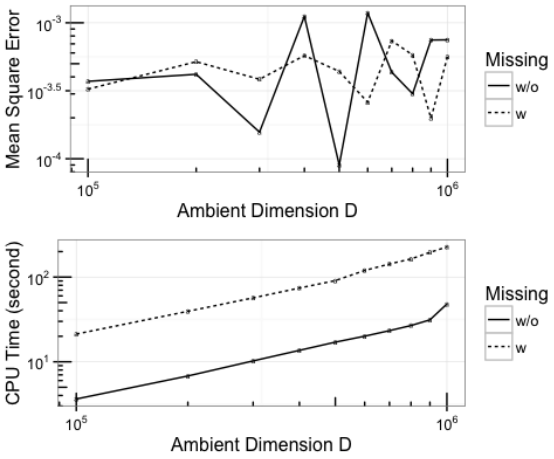

Figure 4: **Top**: MSE in estimating σ^2; **Bottom**: the CPU time fitting GEODE.

25 observations with 5 features missing. We fit PPCA with $d = p$ and run simulations for $D = 100$ and $D = 200$. In estimating σ^2, GEODE generates almost as good results as PPCA, with a CPU time less than 4 seconds for both cases. However, the CPU time of fitting PPCA is 177 seconds for $D = 100$ and 578 seconds for $D = 200$.

Massive D: To evaluate the scalability of GEODE to massive dimensions, we redo the previous two experiments on GEODE with D varying from 10^5 to 10^6. Note that $D = 10^6$ is the largest that we can test on our computer due to storage limits (with $N = 500$ and $D = 10^6$, a single $\boldsymbol{Y}$ takes more than 3 GB storage). For illustration purposes, we fix $p = 5$. The MSE and the CPU time are reported in Figure 4. It is clear that GEODE remains computationally feasible even when $D = 10^6$, while providing very good performance.

5 Mixture of GEODE

Mixture of PPCA (Tipping and Bishop, 1999a) extends PPCA to characterize non-Gaussian data. However, it inherits the computational drawbacks of PPCA. Mixture of factor analyzers (MFA) avoids the isotropic error constraint of mixture of PPCA, but the corresponding EM algorithm (Ghahramani and Hinton, 1996) suffers a similar computational bottleneck. Bayesian MFA is a straightforward Bayesian implementation in small dimensional problems (Diebolt and Robert, 1994; Richardson and Green, 1997), but faces problems in scaling beyond a few 100 dimensions. Inspired by the empirical Bayes idea of GEODE in linear cases, we propose to learn and fix a multiscale set of potential principal component axes in a first stage. In the second stage, we mix across these principal component axes according to their likelihood in a Bayesian paradigm. The ability to learn the intrinsic dimension p's (we allow different spaces having different dimensions) and the ability to characterize uncertainty are both inherited from the linear case.

5.1 Multiscale Principal Axes

To acquire a multiscale set of principal directions, we first adopt METIS (Karypis and Kumar, 1998) to obtain a dyadic clustering tree of the dataset. This is partly motivated by the compressive sensing technique developed by Allard et al. (2012), which efficiently compresses the data by locally finding the best linear subspaces to approximate these dyadic clusters. Moreover, a tree structure allows the model to adapt to different local smoothness by mixing across fine scales and coarse scales. A dyadic clustering tree of the data $\{\boldsymbol{y}_i\}_{i=1}^N$ is defined as follows.

Definition 1. *With $s = 0, \ldots, L$ denoting the scale index and $h = 1, \ldots, 2^s$ denoting the node index within scale s, a level-L **dyadic clustering tree** of $\{\boldsymbol{y}_i\}_{i=1}^N$ is a family of index sets $\mathcal{D}_{sh} \subseteq \{1, \ldots, N\}$ such that*

- *for every s, $\bigcup_{h=1}^{2^s} \mathcal{D}_{sh} = \{1, \ldots, N\}$;*
- *for $s \leq s'$ and $1 \leq h' \leq 2^{s'}$, either $\mathcal{D}_{s'h'} \subseteq \mathcal{D}_{sh}$ or $\mathcal{D}_{s'h'} \cap \mathcal{D}_{sh} = \varnothing$;*
- *for $s < s'$ and $1 \leq h' \leq 2^{s'}$, there exists a unique $h = 1, 2, \ldots, 2^s$ such that $\mathcal{D}_{s'h'} \subseteq \mathcal{D}_{sh}$.*

The tree is denoted by $\{\mathcal{D}_{sh}\}_L$.

METIS generates the tree-structure clustering by partitioning a weighted graph constructed from the data. Following the suggestion by Allard et al. (2012), we add an edge between each data point and its k nearest neighbors and set the weight between any $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ to

be $e^{-\|\boldsymbol{y}_i-\boldsymbol{y}_j\|_2^2/\delta}$. δ is chosen adaptively at each point $\boldsymbol{y}_i$ as the distance between $\boldsymbol{y}_i$ and its $\lfloor k/2 \rfloor$ nearest neighbor. In practice, we fix k to be 30 and constrain the leaf size $|\mathcal{D}_{Lh}|$ to be greater than 10, for $h = 1, \ldots, 2^L$. The depth of the tree L depends on the sample size N, and is automatically decided by METIS. Performance of METIS is illustrated in the supplementary material through multiple simulations.

Equipped with a level-L dyadic clustering tree $\{\mathcal{D}_{sh}\}_L$, the corresponding multiscale principal axes are defined as follows:

Definition 2. *The **multiscale principal axes** of $\{\boldsymbol{y}_i\}_{i=1}^N$ with respect to a level-L dyadic clustering tree $\{\mathcal{D}_{sh}\}_L$ are defined as a family of centroids $\hat{\boldsymbol{\mu}}_{sh} \in \Re^D$ and a family of orthogonal matrices $\hat{\boldsymbol{W}}_{sh} \in \Re^{D\times d}$ such that for all s and h*

- $\hat{\boldsymbol{\mu}}_{sh} = \frac{1}{|\mathcal{D}_{sh}|}\sum_{i\in\mathcal{D}_{sh}} \boldsymbol{y}_i$;
- $\hat{\boldsymbol{W}}_{sh} = \boldsymbol{U}_{sh}$, *where the d column vectors in $\boldsymbol{U}_{sh}$ are the leading d right singular vectors of $\boldsymbol{Y}_{sh}$. $\boldsymbol{Y}_{sh}$ is a $|\mathcal{D}_{sh}| \times D$ matrix with each row representing a demeaned data vector from $\mathcal{D}_{sh}$.*

The multiscale principal axes are denoted as $\{\hat{\boldsymbol{\mu}}_{sh}, \hat{\boldsymbol{W}}_{sh}\}_L$.

In practice, we use fast rank-d SVD to compute $\hat{\boldsymbol{W}}_{sh}$.

5.2 Model Formulation

Equiped with the multiscale principal axes $\{\hat{\boldsymbol{\mu}}_{sh}, \hat{\boldsymbol{W}}_{sh}\}_L$, the mixture of GEODE model (mGEODE) is given by

$$\boldsymbol{y} \sim \sum_{sh} \pi_{sh}\mathcal{N}(\hat{\boldsymbol{\mu}}_{sh}, \boldsymbol{C}_{sh}) \tag{5}$$

where $\boldsymbol{C}_{sh} = \hat{\boldsymbol{W}}_{sh}\boldsymbol{\Sigma}_{sh}\hat{\boldsymbol{W}}_{sh}^\top + \sigma_s^2\boldsymbol{I}$ and $\boldsymbol{\Sigma}_{sh}$ is a $d \times d$ positive diagonal matrix, for $s = 0, \ldots, L$ and $h = 1, \ldots, 2^s$. For all s and h, $\boldsymbol{\Sigma}_{sh}$ and σ_s^2 are given the same prior distribution as in GEODE. We assume isotropic error variance σ_s^2 for each scale s to enable clusters from the same scale to share information.

We then finish the formulation of mGEODE by choosing a prior for the multiscale mixing weights π_{sh}. 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 variance 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, Canale and Dunson (2016) proposed a multiresolution stick-breaking process generalizing usual "flat" stick-breaking (Sethuraman, 1994). In particular, let

$$S_{sh} \sim \text{Be}(1, a_S),\ R_{sh} \sim \text{Be}(b_R, b_R) \tag{6}$$

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

$$\pi_{sh} = S_{sh}\prod_{r<s}(1 - S_{r\ g_{shr}})T_{shr} \tag{7}$$

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 $T_{shr} = 1 - R_{r\ g_{shr}}$. Canale and Dunson (2016) showed that $\sum_{s=0}^{\infty}\sum_{h=1}^{2^s}\pi_{sh} = 1$ almost surely for any $a_S, b_R > 0$. This result makes the defined weights a proper set of multiscale mixing weights. As a_S increases, finer scales are favored, resulting in a highly non-Gaussian density.

In practice, we only consider a truncated finite-depth multiscale mixture with depth being L. Let $\{\tilde{\pi}_{sh}\}_{s\le L}$ denote the truncated weights, which are identical to $\{\pi_{sh}\}$ except that the stopping probabilities at scale L are set equal to one to ensure $\sum_{s=1}^{L}\sum_{h=1}^{2^s}\tilde{\pi}_{sh} = 1$.

5.3 Posterior Computation

The posterior sampling for the mGEODE is almost identical to the GEODE, except for the newly introduced variables (s_i, h_i), S_{sh} and R_{sh}. The conditional posterior of (s_i, h_i) is given by

$$\begin{aligned} &p(s_i = s, h_i = h) \\ \propto &\pi_{sh}\mathcal{N}(\boldsymbol{\mu}_{sh}, \boldsymbol{W}_{sh}\boldsymbol{\Sigma}_{sh}\boldsymbol{W}_{sh}^\top + \sigma_s^2\boldsymbol{I}). \end{aligned}$$

The conditional posteriors of S_{sh} and R_{sh} are given by

$$\begin{aligned} S_{sh} &\sim \text{Beta}(1 + n_{sh}, a_S + v_{sh} - n_{sh}), \\ R_{sh} &\sim \text{Beta}(b_R + r_{sh}, b_R + v_{sh} - n_{sh} - r_{sh}), \end{aligned}$$

where v_{sh} is the number of observations passing through node (s,h), n_{sh} is the number of observations stopping at node (s,h), and r_{sh} is the number of observations that continue to the right after passing through node (s,h).

The remaining Gibbs steps are very similar to GEODE and are provided in the supplementary material.

6 Computational Aspects

GEODE Letting T denote the total number of Gibbs sampler iterations, the computational cost can be split as follows.

Figure 5: The first row shows the original images, second row shows the images with pixels missing, and the third row shows the reconstructed images.

Construction of principal axes: The complexity of fast rank-d SVD is $\mathcal{O}(NDd)$.

Comstruction of sufficient statistics: The complexity of computing $A_i = \tilde{\boldsymbol{y}}_i^\top \tilde{\boldsymbol{y}}_i$ for all i is $\mathcal{O}(ND)$ and the complexity of computing $\boldsymbol{Z}_i = \boldsymbol{W}^\top \tilde{\boldsymbol{y}}_i$ for all i is $\mathcal{O}(NDd)$. Hence, the overall complexity is $\mathcal{O}(NDd)$.

Gibbs sampler: The cost is dominated by updating σ^2 and $\boldsymbol{u}$, whose complexities are both $\mathcal{O}(NTd)$.

Hence the overall complexity of the GEODE is $\mathcal{O}(NDd + NTd)$.

mGEODE Letting K denote the number of nearest neighbours in constructing the weighted graph, the computational cost can be split as follows.

Construction of weighted graph: The complexity of ANN in finding K nearest neighbours is $\mathcal{O}(DN \log N)$ (Arya et al., 1998). The complexity of computing the weights for the graph is $O(KND)$.

Graph partition: The cost of METIS $\mathcal{O}(KN \log N)$.

Construction of multiscale principal axes: For each node (s, h), the cost of applying the fast rank-d SVD is $\mathcal{O}(|\mathcal{D}_{sh}|Dd)$. We have $|\mathcal{D}_{sh}| = \mathcal{O}(2^{-s}N)$ and there are 2^L such $\mathcal{D}_{sh}$'s. Summing them all with $L < \log_2 N$ we obtain a total cost of $\mathcal{O}(N \log NDd)$.

Construction of sufficient statistics: As in the linear case, for each node (s, h), the complexity is $\mathcal{O}(|\mathcal{D}_{sh}|Dd)$. Similar to deriving the complexity for multiple principal axes, the complexity for constructing the sufficient statistics is $\mathcal{O}(N \log NDd)$.

Gibbs sampler: The complexity of the sampler is dominated by updating (s_i, h_i) for all i and updating $\boldsymbol{u}_{sh}$ for all nodes, whose complexities are both $\mathcal{O}(NT2^L d)$.

Hence, the overall complexity of mGEODE is $\mathcal{O}(N \log NDd + NT2^L d)$.

Moreover, the Gibbs sampler converges fast with superb mixing. MCMC diagnostic based on potential scale reduction factor and effective sample size can be found in the supplementary materials. In practice we fix $T = 3000$ with the number of burn-in fixed at 1000 and the stopping time fixed at 800. No thinning is needed.

7 Application

Given that GEODE has already been carefully studied in § 4, we devote this section to illustrate the performance of mGEODE through an image inpainting application.

The Frey faces data (Roweis et al., 2002) contains 1965 20×28 video frames of a single face with different expressions. mGEODE is trained on 1000 images for less than 2 minutes and then reconstruct (predict) the rest 965 damaged images. The reconstruction is done in less than 10 minutes. The mean absolute reconstruction error of mGEODE is 7.04, which outperforms the error of 7.40 reported by Titsias and Lawrence (2010). 14 reconstructions are shown in Figure 5. In this application, we set $d = 20$. Increasing d moderately had essentially no impact on the results.

8 Discussion

In high dimensional applications, PPCA is ubiquitously used since it not only provides an low-dimensional embedding, but also a density estimation. Unfortunately, the dominating algorithm fitting PPCA is not scalable to high dimensions. We tackle this problem by proposing a empirical Bayes model novelly built upon a fast SVD technique. The proposed GEODE showed excellent performance in scaling computationally, while providing a valid characterization of uncertainty in predictions. It also showed excellent performance in inferring the subspace dimension and handling missing data. We also propose a mixture of GEODE model which mixes local GEODE models across a dyadic clustering tree. This mGEODE model showed excellent performance in a real world data application.

References

W. K. Allard, G. Chen, and M. Maggioni. Multi-scale geometric methods for data sets ii: Geometric multi-resolution analysis. *Applied and Computational Harmonic Analysis*, 32(3):435–462, 2012.

S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, 45(6):891–923, 1998.

A. Bhattacharya and D. B. Dunson. Sparse Bayesian infinite factor models. *Biometrika*, 98(2):291–306, 2011.

A. Canale and D. B. Dunson. Multiscale Bernstein polynomials for densities. *Statistica Sinica*, 2016. In press, doi: 10.5705/ss.202015.0163.

C. M. Carvalho, J. Chang, J. E. Lucas, J. R. Nevins, Q. Wang, and M. West. High-dimensional sparse factor modeling: applications in gene expression genomics. *Journal of the American Statistical Association*, 103(484), 2008.

J. Diebolt and C. P. Robert. Estimation of finite mixture distributions through bayesian sampling. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 56(2):363–375, 1994.

M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

Z. Ghahramani and G. E. Hinton. The em algorithm for mixtures of factor analyzers. Technical report, Technical Report CRG-TR-96-1, University of Toronto, 1996.

A. Ilin and T. Raiko. Practical approaches to principal component analysis in the presence of missing values. *The Journal of Machine Learning Research*, 11:1957–2000, 2010.

G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.

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

C. E. Rasmussen. The infinite Gaussian mixture model. In *NIPS*, volume 12, pages 554–560. MIT; 1998, 1999.

S. Richardson and P. J. Green. On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(4): 731–792, 1997.

V. Rokhlin, A. Szlam, and M. Tygert. A randomized algorithm for principal component analysis. *SIAM Journal on Matrix Analysis and Applications*, 31(3): 1100–1124, 2009.

S. Roweis. Em algorithms for pca and spca. *Advances in neural information processing systems*, pages 626–632, 1998.

S. T. Roweis, L. K. Saul, and G. E. Hinton. Global coordination of local linear models. In *NIPS*, volume 2, pages 889–896. MIT; 1998, 2002.

J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

W. Shen, S. T. Tokdar, and S. Ghosal. Adaptive Bayesian multivariate density estimation with Dirichlet mixtures. *Biometrika*, 100(4):623–640, 2013.

M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural computation*, 11(2):443–482, 1999a.

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

M. Titsias and N. Lawrence. Bayesian Gaussian process latent variable model. In *the International Conference on Articial Intelligence and Statistics*, 2010.