Bayesian Inference on Matrix Manifolds for Linear Dimensionality Reduction

Andrew Holbrook Alexander Vandenberg-Rodes Babak Shahbaba

Department of Statistics
University of California, Irvine, California, USA

***Abstract*—We reframe linear dimensionality reduction as a problem of Bayesian inference on matrix manifolds. This natural paradigm extends the Bayesian framework to dimensionality reduction tasks in higher dimensions with simpler models at greater speeds. Here an orthogonal basis is treated as a single point on a manifold and is associated with a linear subspace on which observations vary maximally. Throughout this paper, we employ the Grassmann and Stiefel manifolds for various dimensionality reduction problems, explore the connection between the two manifolds, and use Hybrid Monte Carlo for posterior sampling on the Grassmannian for the first time. We delineate in which situations either manifold should be considered. Further, matrix manifold models are used to yield scientific insight in the context of cognitive neuroscience, and we conclude that our methods are suitable for basic inference as well as accurate prediction. All datasets and computer programs are publicly available at http://www.ics.uci.edu/~babaks/Site/Codes.html.**

I. Introduction

The advent of Markov chain Monte Carlo in its many forms popularized Bayesian inference in the last decades of the 20^{th} century. More recently, hybrid Monte Carlo (HMC) [29] has enabled efficient simulation from models with increasing numbers of parameters and deeper hierarchies. Whereas HMC has extended fast Bayesian inference to higher dimensional models, high dimensional data analysis remains a lasting challenge for the statistical learning community. Linear dimensionality reduction is the most well established genre of dimensionality reduction tools. Famous examples from this toolkit include principal component analysis, canonical correlation analysis, and linear discriminant analyses (PCA, CCA, and LDA, respectively).

PCA is a non-probabilistic linear dimensionality reduction technique in which the eigenvalue decomposition of the empirical covariance matrix is considered. Generative models include probabilistic PCA (PPCA) and factor analysis. Both of these methods model high dimensional data as generated by a multivariate Gaussian distribution with covariance the sum of a low-rank matrix and a diagonal matrix (restricted to a multiple of the identity under PPCA). Both have maximum likelihood (ML) as well as Bayesian implementations. The prevalent treatment of Bayesian PCA analysis is rather complicated: the columns of the "loading matrix" are modeled as independent multivariate Gaussian distributions each with its own variance hyper-prior [6].

Corresponding author Andrew Holbrook may be contacted at aholbroo@uci.edu.

In the following contribution we parsimoniously model the factor matrix as a single manifold valued parameter. Similar ideas in the context of factor analysis have been proposed in [10], [21], but we take a more general viewpoint. We unify a collection of models, including factor analysis and *supervised PPCA* [36], along with exponential-family versions such as *exponential* PCA [24], [28] and *supervised* EPCA [18], [23], [27]. We distinguish between models parameterized by the Grassmannian and those defined on the Stiefel manifold.

Our present approach is greatly informed by [8], in which the differential geometric framework of embedding geodesic Monte Carlo is established and implemented on the sphere and Stiefel manifold. We also benefit from the matrix manifold optimization literature. Chief among these are [1], [15] and [2]. Finally, follow the work of [13], who argue for the relevance of matrix manifold optimization to the field of linear dimensionality reduction. The present results are largely extensible to non-probabilistic approaches discussed therein.

II. Bayesian linear dimensionality reduction

The most prevalent dimensionality reduction methods fall into the factor analysis framework. Such models specify the N observed continuous data points $y_1, \dots, y_N \in \mathbb{R}^d$ as

$$y_j = Fz_j + \mu + \epsilon_j, \tag{1}$$

where $z_j \in \mathbb{R}^k$ are the latent factors, F is the $d \times k$ factor loading matrix, and ϵ_j are iid $\mathrm{N}_d(0, \Psi)$ with Ψ diagonal covariance matrix. Typically, the parameters F, μ, Ψ are optimized, either by EM or using closed form expressions available when Ψ is restricted to be a multiple of the identity [34]. If we place $N(0, I_k)$ priors on z_j, this latent factor is easily integrated out, leaving the sampling distribution, conditioned on F, μ, Ψ, as

$$y_j \sim N(\mu, FF^T + \Psi). \tag{2}$$

This formulation assumes that the data lies close to an affine subspace spanned by the column vectors of F. There are, however, a wide continuum of subspaces that approximately span the data. Picking a single subspace can dramatically understate variation in the data and lead to over-fitting. One approach is to instead use the Bayesian framework to obtain a *posterior* distribution over matrices F which best explain

the data. By thus integrating over high probability subspaces one may avoid over-fitting in a natural way. This was the approach taken in [28], in which the authors further generalized the factor analysis model to allow for observations y_j from any exponential family distribution, allowing for principled dimension reduction for discrete data as well as continuous.

Despite its pleasing formulation, generating samples from the high-dimensional posterior distributions of F proves to be especially difficult. Even using the very efficient hybrid Monte Carlo (see below), the model in [28] takes thousands of samples just to reach a high-probability region of the posterior.

The fundamental problem is that the model is highly over-parameterized on account of the rotational symmetry of the spherical Gaussian distribution. For any orthogonal $k \times k$ matrix V, we have

$$(FV^T)(FV^T)^T = FF^T, \tag{3}$$

Furthermore, if the entries of F are iid normal, then FV^T has the same distribution as F. This causes significant problems for any MCMC sampler, as the resulting log-posterior $\log P(F, \mu, \Psi|\mathbf{y})$ is constant along $p(p-1)/2$-dimensional highly curved contours generated by the action of the orthogonal group. This high degree of curvature in the log-posterior can cause extremely low acceptance rates during sampling.

One solution to the lack of identifiabiliy is to specify the factor loading matrix F as upper triangular with positive entries on the diagonal [9], [26]. This specification has well-known problems due primarily to the ordering of the variables implied by the upper-triangular loading matrix [10]. In what follows, we will explore an alternative approach and discuss its application in a more general dimensionality reduction framework.

A. Reparameterizing with the Stiefel manifold

Under the assumption of dimension reduction ($d > k$), the singular value decomposition (SVD) $F = U\Lambda V^T$ can be modified so that U and V are, respectively, $d \times k$ and $k \times k$ matrices with orthonormal columns, while Λ is a $k \times k$ diagonal matrix with non-negative entries (the singular values) in decreasing order. Assuming the singular values are all distinct, F is uniquely specified by U, Λ, and V. Now, recalling that $z \sim N(0, I_k)$ implies that $V^T z \sim N(0, I_k)$, we ignore the superfluous rotation by V^T and reparameterize (1) into

$$y_j = U\Lambda z_j + \mu + \epsilon_j. \tag{4}$$

The collection of $d \times k$ matrices with orthogonal columns, denoted by $\mathcal{O}_{d,k}$, is known as the real *Stiefel manifold*, which is a (compact) Riemannian manifold of dimension $dk - \frac{k(k+1)}{2}$. Here, as in later models, we place a uniform prior distribution over $U \in \mathcal{O}_{d,k}$. We furthermore specify μ and u_j to have independent mean-zero Normal priors, and the diagonal scale matrix Λ has diffuse, positive valued priors on its entries.

We remark that similar models were considered in [10], [21], which model the SVD of the data, instead of the SVD of the factor matrix. In particular, [21] assumes that $Z \in \mathcal{O}_{N,k}$, where $Z^T = [z_1 \cdots z_N]$. The conditional distributions of each orthonormal column $P(U_j|U_{-j}, \Lambda, Z, Y)$ and $P(Z_j|Z_{-j}, \Lambda, U, Y)$ are shown to follow a von Mises-Fisher distribution, making the model amenable to Gibbs sampling. Relaxing this assumption to $Z_1, \ldots, Z_N \sim N(0, I_k)$, as we do with (4), is not that different, at least a-priori. In most situations we have $N \gg k$, (even if the data dimension $d \approx N$) and the high-dimensional independent Gaussian random variables are orthogonal in prior expectation.

This approach – directly sampling the matrix parameter U over the Stiefel manifold $\mathcal{O}_{d,k}$ – generalizes to the models mentioned in the introduction, as we will show in the next section. Our motivation is simple: *when the relevant dimension reduction methods involve orthogonal projections, one should directly model the orthogonal projection, not an over-parameterized version.* In the case of factor analysis, i.e. when the entries of Ψ are allowed to take on distinct values, it is often appropriate to parameterize by another manifold instead.

B. Separated covariance model with the Grassmann manifold

From (4) it is clear that the PPCA/factor analysis model is a fully generative probability model for the observations $\mathbf{y}$, with the form

$$y_j \sim N(\mu, U\Lambda\Lambda U^T + \Psi), \tag{5}$$

with $U \in \mathcal{O}_{d,k}$ and Λ a diagonal $k \times k$ matrix.

The covariance matrix for y_j can be expressed in a somewhat different fashion, as

$$y_j \sim N(\mu, \Phi(UU^T + \Psi)\Phi), \tag{6}$$

where Φ is a diagonal scale matrix for the observations. This separation strategy is similar to the approaches of [25] in the context of generalized estimating equations and [5] in the context of Bayesian covariance modeling. Model (6) is distinct insofar as $UU^T + \Psi$ will almost surely never be a correlation matrix. The (non-negative) diagonal matrix Ψ of residual variances is known as the *uniquenesses* in the factor analysis literature [22]. While in classical factor analysis the loading matrix U can be an arbitrary d-by-k matrix, we will here constrain U to have orthogonal columns. That way, UU^T is the unique projection matrix onto the subspace on which the standardized data $\Phi^{-1}(y_j - \mu)$ approximately lie.

We remark that this model is still unidentifiable up to right rotations $U \mapsto UV^T$ with $V \in \mathcal{O}_{k,k}$, however, this can be solved by considering U as an element of the quotient space $\mathcal{O}_{d,p}/\mathcal{O}_{k,k}$, which is invariant to right-rotations. This quotient space is known as the *Grassmann* manifold $G_{d,k}$, which is in one-to-one correspondence with the set of p-dimensional subspaces of $\mathbb{R}^d$. In Section III we show how to perform inference on the Grassmann manifold while holding right rotations constant. Next, we will discuss a generalization of this approach to a broader class of models.

C. Extension to exponential family PCA

A significant limitation of standard PCA appears when one tries to apply it to binary, count, or categorical data. Taking

a cue from generalized linear models, [12] models each data point $y_{j,i}$ as coming from an exponential family distribution:

$$P(y|\theta) = h(y)\exp(y\theta - b(\theta)). \tag{7}$$

Here the natural parameter θ is related to the mean $\mu = \mathbf{E}(y \mid \theta)$ through the canonical link function: $\theta = g(\mu)$, where $g^{-1}(\theta) = b'(\theta)$. Dimension reduction is then applied to the natural parameter θ, which for many distributions of interest (e.g. Bernoulli or Poisson) can take any value on the real line, unlike mean μ.

For simplicity we describe the case of binary data. With $X_j \in \{0,1\}^d$, we have the canonical logit link function $g(p) = \log \frac{p}{1-p}$ and

$$X_{j,i} \sim Bernoulli(p_{j,i}), \quad \text{with} \tag{8}$$

$$p_{j,i} = g^{-1}\left(\sum_{\ell=1}^{k} U_{i,\ell}\lambda_\ell z_{j,\ell} + \mu_i\right), \tag{9}$$

where the $X_{j,i}$ are conditionally independent given the parameters $(U, \Lambda, \mathbf{z}, \mu)$. We give parameters $U, \Lambda, \mathbf{z}, \mu$ the same priors as in (4) and recall that $U \in \mathcal{O}_{d,k}$.

In [12], $\Lambda = I_k$ and μ is set to zero, but in particular U is an unconstrained $d \times k$ matrix, with all parameters learned via (penalized) maximum likelihood. The Bayesian versions in [24], [28] have a slightly different parameterization, but U is again not restricted to the Stiefel manifold.

D. Extension to PCA regression and classification models

Suppose now that *paired* data (X, y) is collected. Often the goal is to fit a joint model for the data, such that future X data can be used to predict y – the supervised learning problem. Two classical linear dimensionality reduction methods for this case are partial least squares (PLS) and linear discriminant analysis (LDA). In the case that the y_j are class labels, LDA finds a projection $M \in \mathcal{O}_{d,k}$ of the X data such that the ratio of the between-class variance Σ_B to within-class variance Σ_W is maximized [13]:

$$\underset{M \in \mathcal{O}_{d,k}}{\arg\max} \frac{\operatorname{tr}(M^T \Sigma_B M)}{\operatorname{tr}(M^T \Sigma_W M)}. \tag{10}$$

With continuous data y, on the other hand, partial least squares is concerned with finding orthogonal projections of X and y to a common latent subspace such that their covariance is maximized.

Here, we focus on classification problems, specifically, variants of LDA. Often the data X_j are high-dimensional binary or count valued data, which motivates a generalized linear model framework as with exponential PCA, for simultaneously modeling X and y. We consider the following model:

$$X_j \sim p(x \mid \theta_j), \text{ an exponential family vector} \tag{11}$$

$$\theta_j = g_X^{-1}(U\Lambda z_j + \mu) \tag{12}$$

$$y_j \sim p(y \mid \eta_j), \text{ an exponential family r.v.} \tag{13}$$

$$\eta_j = g_y^{-1}(\beta^T z_j + \beta_0) \tag{14}$$

We keep the prior specification for U, Λ, Z, μ in (12) as before – only the β coefficients are new.

Later, we will discuss an application of this model to count data coming from neural spike trains, where we specify X_j as Poisson with canonical link function $g(x) = \log x$, and model the response variable y – the behavioral response from a finite set of possible outcomes – using (multinomial) logistic regression.

An example of this type of model is supervised logistic PCA [36], which was applied to genomic data. The parameters there were learned by maximum likelihood, with U restricted to the Stiefel manifold. As we saw before with exponential PCA, existing Bayesian versions such as [23] do not restrict U to have orthonormal columns.

In contrast to prior Bayesian treatments, we model U directly as a random element on the Stiefel and Grassmann manifolds and, in order to do so, employ the embedding geodesic Monte Carlo of [8].

III. Bayesian inference and the geodesic Monte Carlo

Given data $y_1, \ldots, y_N \in \mathbb{R}^n$, it is often useful to specify a generative model in the form of a likelihood function, $p(y|q)$. This is the forward model. In the following we allow $q \in \mathcal{M}^m$ to be an m-dimensional manifold valued vector that parameterizes the likelihood. Endowing q with prior distribution $p(q)$ renders the posterior distribution

$$\pi_{\mathcal{H}}(q) = p(q|y) = \frac{p(y|q)p(q)}{\int p(y|q)p(q)\mathcal{H}^m(dq)}\,. \tag{15}$$

The integral is often referred to as the evidence and may be interpreted as the probability of observing data y given the model. Here the prior distribution is defined with respect to the Hausdorff measure

$$\mathcal{H}^m(dq) = \sqrt{|G(q)|}\lambda^m(dq)\,. \tag{16}$$

This measure is exactly the Lebesgue measure λ^m scaled by metric based volume element $\sqrt{|G(q)|}$, where G is the Riemannian metric on $\mathcal{M}$. Let $\pi_{\mathcal{H}}(q)$ denote the posterior density with respect to the Hausdorff measure.

For most interesting models the evidence integral is intractable and high dimensional models do not lend themselves to numerical integration. Non-quadrature sampling techniques such as importance sampling or even random walk MCMC are similarly cursed.

A. Hybrid Monte Carlo

Hybrid Monte Carlo is an effective sampling tool for high dimensional models with thousands of parameters. Riemannian manifold HMC [17] is an extension with connections to Newton's method. Embedding geodesic Monte Carlo is a further extension and is the basis for HMC on the Grassmann and Stiefel manifolds.

HMC is often referred to as Hamiltonian Monte Carlo. One builds a Hamiltonian system from the posterior distribution

$\pi(q)$ and an augmenting Gaussian variable p. The negative-log transform turns the probability distribution functions into a potential energy function $U(q)$ and corresponding kinetic term $K(p)$. Thus q and p become the position and momentum of Hamiltonian function

$$\begin{aligned} H(q,p) &= U(q) + K(p) \\ &= -\log \pi(q) + \frac{1}{2} p^T p \ . \end{aligned} \tag{17}$$

In order to draw samples from $\pi(q)$, the system is numerically advanced according to Hamilton's equations:

$$\begin{aligned} \frac{dq}{dt} &= \frac{\partial H}{\partial p} \\ \frac{dp}{dt} &= -\frac{\partial H}{\partial q} \ . \end{aligned} \tag{18}$$

In the case that variable q takes values on a non-Euclidean space, the above representations are insufficient as curvature is not taken into account. On the other hand, if certain facts about the manifold of interest are known, one may overcome this difficulty by isometrically embedding the manifold into Euclidean space.

Let d be the standard Euclidean metric and denote the metric preserving embedding $x : (\mathcal{M}, G) \to (\mathbb{R}^n, d)$. This map renders a similar Hamiltonian to (6) above:

$$H(x,v) = -\log \pi_{\mathcal{H}}(x) + \frac{1}{2} v^T v \ . \tag{19}$$

Instead of Gaussian momentum $p \in \mathbb{R}^m$, we augment by Gaussian velocity $v \in T_x\mathcal{M}$, the tangent space to the embedded manifold at x. Note that we write $\pi_{\mathcal{H}}$ now since q is manifold valued and $\pi_{\mathcal{H}}(x) = \pi_{\mathcal{H}}(u)$ since $\mathcal{H}^m$ is invariant under isometric embeddings.

One may forward integrate the corresponding Hamiltonian equations by splitting the Hamiltonian into potential and kinetic terms [33]. The solution to the potential term is given by $x(t) = x(0)$ and

$$v(t) = v(0) + t\, \Pi_{T_x\mathcal{M}}\big(\nabla_x \log \pi_{\mathcal{H}}(x)|_{x=x(0)}\big) \ . \tag{20}$$

Here $\Pi_{T_x\mathcal{M}}$ is the orthogonal projection onto the the tangent space to the embedded manifold at x. For the Stiefel and Grassmann manifolds, this map is available in closed form. The solution to the kinetic term is given by the unique geodesic (with respect to the Levi-Civita connection) starting at point $x(0)$ and with initial velocity $v(0)$.

The embedding geodesic Monte Carlo algorithm is presented in Algorithm 1. To implement HMC on an embedded manifold only three quantities require evaluation: the log posterior density $\log \pi_{\mathcal{H}}$ and its gradients; the orthogonal projection from the ambient space $\mathbb{R}^n$ onto tangent space $T_x\mathcal{M}$; and geodesic flow associated with initial velocity $v \in T_x\mathcal{M}$. Since the metric G only appears in the Jacobian term of the prior density, there is no need to compute the metric G when a uniform prior distribution is available. Both the Grassmann and Stiefel manifolds admit a uniform distribution. They also have analytic geodesic flows as well as closed form projections onto the tangent space at any point x. Thus they are suitable candidates for embedding geodesic Monte Carlo.

Algorithm 1 Embedding geodesic Monte Carlo [8]

$v \sim N(0, I_n)$
$v \leftarrow \Pi_{T_x\mathcal{M}}(v)$
$h \leftarrow \log \pi_{\mathcal{H}}(x) - \frac{1}{2} v^T v$
$x^* \leftarrow x$
for $\tau = 1, \ldots, T$ **do**
  $v \leftarrow v + \frac{\epsilon}{2} \nabla_{x^*} \log \pi_{\mathcal{H}}(x^*)$
  $v \leftarrow \Pi_{T_x\mathcal{M}}(v)$
  Progress (x^*, v) along the geodesic flow defined by initial velocity v.
  $v \leftarrow v + \frac{\epsilon}{2} \nabla_{x^*} \log \pi_{\mathcal{H}}(x^*)$
  $v \leftarrow \Pi_{T_x\mathcal{M}}(v)$
end for
$h^* \leftarrow \log \pi_{\mathcal{H}}(x^*) - \frac{1}{2} v^T v$
$u \sim U(0,1)$
if $u < \exp(h^* - h)$ **then**
  $x \leftarrow x^*$
end if

B. Two matrix manifolds

The Grassmann manifold, denoted $G_k(\mathbb{R}^d)$ or $G_{d,k}$, is the space of k-dimensional subspaces of $\mathbb{R}^d$. On this manifold, each point is a linear subspace of $\mathbb{R}^d$ as well as an equivalence class of all d-by-k matrices the columns of which span the subspace. The Stiefel manifold $\mathcal{O}_{d,k}$ is the space of orthonormal matrices of height d and width k. See [11] for an overview of the two manifolds and classical statistical inference. Both manifolds are smooth and compact, and both have been used to great success in non-Bayesian dimensionality reduction [13].

Due to their compactness, both manifolds admit a uniform distribution with respect to the Hausdorff measure. This fact simplifies geodesic Monte Carlo since the uniform measure is constant and cancels in the accept-reject step. [8] provides formulas for projection and co-geodesic flow of the Stiefel manifold, but (to the best of our knowledge) HMC has never been performed on the Grassmannian. We provide the necessary tools to do so.

C. Projection and flow on the Grassmann manifold

For any point $X \in G_{d,k}$, the tangent space to $G_{d,k}$ at X consists of the rank $(d-k)$ subspace orthogonal to X. That is, if we let X_1 be an orthonormal class representative with columns spanning X (i.e., $[X_1] = X$), then the projection onto the tangent space at X is given by the simple formula

$$\begin{aligned} \Pi_{T_X G_k(\mathbb{R}^n)} &= \big(I_n - X_1 (X_1^T X_1)^{-1} X_1^T\big) \\ &= \big(I_n - X_1 X_1^T\big) \ . \end{aligned} \tag{21}$$

From here on we conflate point $X \in G_{d,k}$ with any orthonormal matrix X_1 satisfying $[X_1] = X$ and vice-versa. Given an orthonormal representative $X(0) \in G_{d,k}$, any vector $\dot{X}(0) \in T_X G_{d,k}$ determines a unique geodesic path with respect to which $\dot{X}(0)$ acts as initial velocity. In order to compute this path, we require the singular value decomposition

$\dot{X}(0) = U\Sigma V^T$. Once we have this decomposition, the geodesic path is given by

$$\Big(X(t), \dot{X}(t)\Big) = \Big(X(0)V, X(0)\Big) \times \begin{pmatrix} \cos\Sigma t & -\sin\Sigma t \\ \sin\Sigma t & \cos\Sigma t \end{pmatrix} \begin{pmatrix} V^T \\ \Sigma V^T \end{pmatrix} . \quad (22)$$

We attain (22) by differentiating formula (2.65) of [15]. This formula includes the familiar rotation matrix applied element-wise to the elements of Σ. It is easy to see that $X(t)^T X(t) = I_k$. That is, the geodesic formula advances orthonormal class representative to orthonormal class representative. In (22) V acts as a random right rotation on $X(0)$, but the geodesic remains well-defined even when V is fixed to any orthogonal matrix [15]. Due to the fact that $X(t)$ is orthonormal, allowing V to vary causes both Grassmann and Stiefel geodesic Monte Carlo to perform similarly. When V is fixed, however, geodesic Monte Carlo is performed directly over subspaces of $\mathbb{R}^d$ and mere changes of basis are never considered.

In order to implement geodesic Monte Carlo on the Grassmannian, one need only select any orthonormal d-by-k matrix, X_0, then advance Algorithm 1 by implementing (21) in algorithm lines 2 and 7, and implementing (22) in line 8.

IV. Convergence and posterior summaries using the projection Frobenius metric

As model dimensionality increases, it becomes increasingly difficult to calibrate parameters and assess model fit to data. This is particularly true when a model is built for learning higher moments and latent factors. When assessing model effectiveness with simulated data, it is important to have a measure of closeness to truth, while for real data it is important to understand the uncertainty of parameter estimates, whether through confidence intervals or posterior distributions.

In the context of the above factor analysis type models, one often has a collection of samples $U_1, \ldots, U_n$ of the factor loading matrices, each of which provides a different subspace $\mathrm{Range}(U_j)$ on which the data is assumed to approximately lie. Crucially, we are interested in understanding the variability in these *subspaces*, instead of the variability of the matrix elements. This is because for any $V \in \mathcal{O}_{k,k}$ the matrix UV^T describes the same subspace – multiplying by V^T merely rotates the basis vectors within it.

Hence the Grassmann manifold $G_{d,k}$ (being the space of linear subspaces) is the very space within which we would like to characterize variability. In the following we explore this manifold's projection Frobenius (pF) metric, which is easily used for diagnosing convergence of the MCMC chain, as well as for assessing the variability of the posterior distribution of U. In addition to being high dimensional, the posterior distribution of matrix representative U is typically multi-modal (if not unidentifiable) on account of the symmetry under rotations by V. Therefore traceplots of its entries are hard to diagnose for convergence. On the other hand, metrics on the Grassmann manifold are agnostic to such rotations. Thus the pF distance between the samples and a reference point proves much more informative.

A. Projection Frobenius distance on the Grassmann manifold

There are a variety of metrics that have been defined on $G_{d,k}$, all of which are easily calculated in terms of the *principle angles* between k-dimensional linear subspaces X and Y in $\mathbb{R}^d$. See [15] for short discussion. These angles are defined with respect to the SVD of the matrix representatives: if we have $X^T Y = U \cos\Theta V^T$ with $U, V \in \mathcal{O}_{k,k}$ and Θ a non-negative diagonal matrix, then the principle angles are the singular values $\theta_1, \ldots, \theta_k \in [0, \frac{\pi}{2}]$ [15].

Letting $\|\cdot\|_F$ denote the Frobenius norm, the pF distance between X and Y is defined as

$$d_{pF}(X,Y) = \frac{1}{2}\|XX^T - YY^T\|_F = \sqrt{\sum_{j=1}^{k} \sin^2\theta_j} , \quad (23)$$

while the closely related *geodesic* distance is

$$d_g(X,Y) = \sqrt{\sum_{j=1}^{k} \theta_j^2} . \quad (24)$$

Note that $d_{pF}(\cdot,\cdot)$ has maximal distance $\sqrt{k}$.

B. The projection Frobenius mean

Given a collection of samples $U^{(1)}, \ldots, U^{(N_s)}$ from the Grassmann manifold, we would like to assess traceplots of the distances between these samples and a reference point – typically the posterior mean. Since the Grassmann manifold is not a vector space, we use the idea of a *Karcher* mean, which is a point $U^{(0)}$ minimizing the average distance to the samples:

$$U^{(0)} = \underset{U \in G_{d,k}}{\arg\min} \sum_{j=1}^{N_s} d(U, U^{(j)}). \quad (25)$$

Under the pF metric we will refer to the Karcher mean as the *pF mean*. The pF mean has a simple closed form formula requiring only a single SVD, unlike the Karcher mean using the geodesic metric [19].

For illustration, in Figure 1 we show samples $U^{(s)} \in G_{16,3}$ of the orthogonal loading matrix for the exponential-family PCA experiment discussed in the following section. We first compute the chordal mean $U^{(0)}$ of the latter samples $U^{(5001)}, \ldots, U^{(10000)}$, and then plot the pF distance (23) between $U^{(0)}$ and all the samples $U^{(s)}$, $s = 1, \ldots, 10000$. This technique is interpretable even if samples are drawn using Stiefel geodesic Monte Carlo. One simply identifies the point on the Stiefel manifold, an orthogonal matrix, with the subspace its columns span.

Note that after about sample 30, all loading matrix samples lie within a pF distance 0.8 of the pF mean. We caution against low-dimensional intuition: despite the pF distance between points on $G_{16,3}$ being at most $\sqrt{3}$, for $r < 1$ the metric ball

$$B(r) = \{U \in G_{d,k} \ : \ d_{pF}(U, U_0) < r\} \quad (26)$$

has exponentially small volume (in d and k) under the uniform measure [14]. E.g., in our example $|B(0.8)| \approx 10^{-11}$.

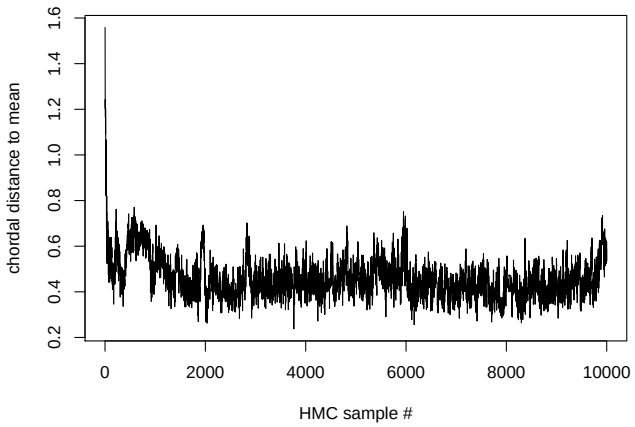

Fig. 1. Traceplot of pF distance between the U samples and their pF mean for the synthetic data example.

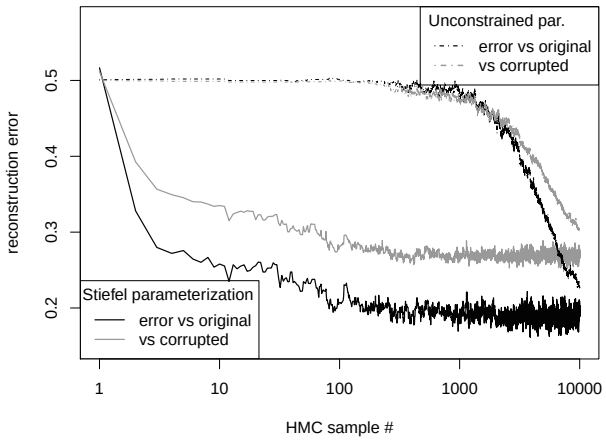

Fig. 3. Traces of reconstruction error under HMC sampling of the exponential PCA model, comparing the Stiefel manifold parameterization to the unconstrained model of [28].

V. Results

A. Synthetic data

We illustrate here the difference between an over-parameterized factor analysis model and the model (8), (9) with loading matrix constrained to lie on the Stiefel manifold. Recall the simulated data from [28]: three 16-bit strings were chosen at random, and each repeated 200 times to form 600 binary vectors. Each bit was flipped independently with probability 20% (10% in [28]), giving a corrupted set of vectors $y_1, \ldots, y_{600}$. We then randomly sample half (4800/9600) of the entries of $\mathbf{y}$ and set them as missing data. (see Figure 2). Next we fit this data to the binary logistic regression formulation of (8), (9) with just three latent dimensions, and compare with the model of [28] with unconstrained loading matrix U. The first question is how quickly the models fit the corrupted and half-missing data; second, whether the low-rank assumption allows for accurate reconstruction of the original data, despite only 50% of the corrupted data being available. The results are shown in Figure 3. Note that the HMC sampler for the Stiefel manifold model immediately reconstructs most of the missing and corrupted data: after a mere 100 Stiefel HMC samples, the *per sample* error rate has reached an equilibrium of about 20%. The unconstrained model needs over 10,000 samples to attain a similar accuracy. Both HMC samplers are tuned to achieve an acceptance rate of 60-80%, and with $L = 80$ steps per sample. By averaging over just samples 100,101,...,500 the Stiefel model obtains an error rate of 10.7%. The right most panel of Figure 2 shows this Bayesian reconstruction. Averaging over the remaining 9,500 samples only drops the error rate to 9.4%.

B. Real data

Next, we apply our models to three separate data sets: the first is the 18-dimensional *Tobamovirus* data from [31], [34]; the second is the $52-$dimensional metabolite dataset of [32]; and the third is 53-dimensional neural spike train data from an experiment conducted at (university name, here). We use the first data set to compare predictive accuracy of the Grassmann-manifold factor analysis model (6) against that of its maximum likelihood counterpart. For the second dataset we measure imputation performance of the PPCA model (1) with missing data, specifically comparing geodesic Monte Carlo to the Variational Bayes method. For the neural firing data we employ the supervised Poisson-Logistic model of (11) – (14), and compare to other popular supervised classification algorithms. Gradients and log-probabilities are computed using [16]. Computation is also performed using [20], [35].

1) Tobamovirus data: We compare the Bayesian-Grassmannian factor analysis model (6) to its maximum likelihood counterpart with respect to predictive accuracy. In order to do so, we implemented a modified leave-one-procedure (LOO) on the Tobamovirus dataset. The Tobamovirus data features 38 observations of vectors in $\mathbb{R}^{18}$. For 38 iterations of LOO we trained both the Bayesian and the MLE models on 37 observations. Next, we randomly divided the hold-out observation into elements to predict (predictands) and elements upon which to condition (predictors). We then computed the Gaussian conditional mean for the predictands, given the predictors. We chose the mean absolute (L^1) distance of predictand from conditional mean as prediction criterion. Among other things, the conditional mean is a function of covariance matrix Σ. Therefore, if one model has better prediction with respect to conditional mean, we may conclude that its low-dimensional representation $\Sigma = U\Lambda U^T + \sigma^2 I_d$ (in the case of PPCA) is superior. In addition to comparing mean LOO error, we show the methods' sensitivity to sparse predictors. Figure 4 shows the results, with the Bayesian-Grassmannian implementation having lower prediction error as the number of predictors decreases. The Bayesian model was initialized at the MLE

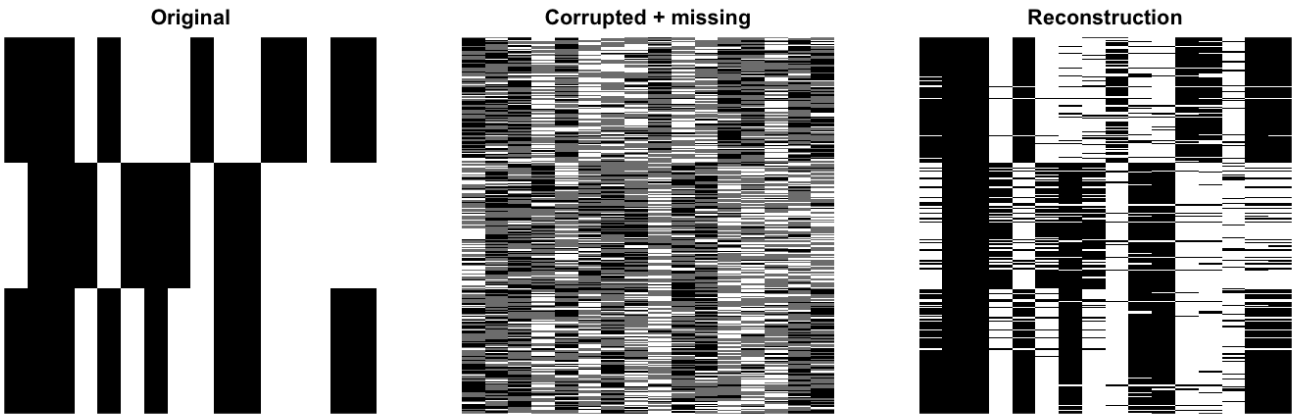

Fig. 2. From left to right: Original samples of bit vectors; After randomly corrupting 20% and setting 50% as missing (grey pixels); Reconstruction averaging over 401 Stiefel HMC samples

.

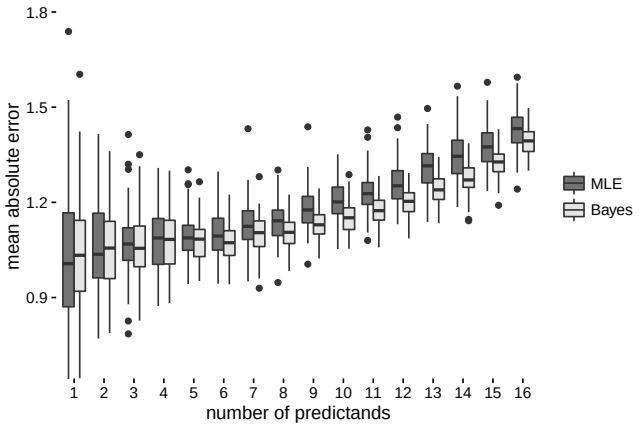

Fig. 4. Tobamovirus data, comparing the Grassmann manifold Bayesian model to standard MLE model. The 18 elements of the hold-out vector are assigned to be predictors and predictands, with the displayed prediction error an average over 100 random assignments. As the number of predictands increases (and number of predicting elements decreases) element-wise errors increase, and the Bayesian implementation outperforms amid greater uncertainty.

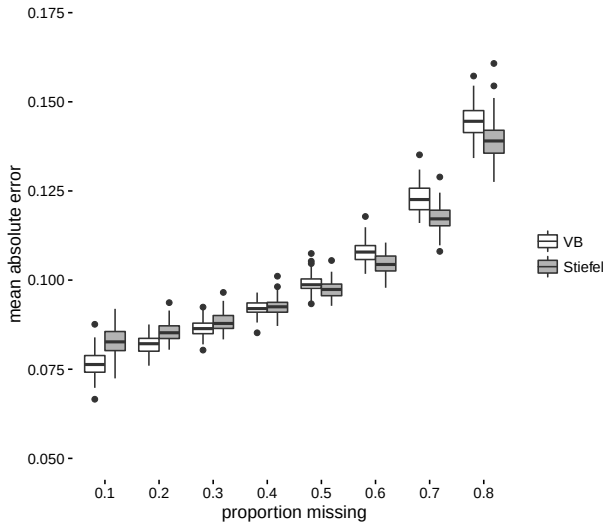

Fig. 5. Imputation performance on metabolite dataset, comparing Variational Bayes to Stiefel HMC. As uncertainty increases, the fully Bayesian treatment excels. Both methods are presented at optimal number of latent factors. For Variational Bayes, 52, for the Stiefel HMC model, there are diminishing returns after 7 latent dimensions

estimates, with U orthogonalized. We used a scant 200 samples for each LOO interation (without thinning), as traceplots of the pF distance (23) of samples U to the MLE showed extremely low autocorrelation.

2) Metabolite data: We compare the performance of the Bayesian-Stiefel PPCA model (1) to Variational Bayes [7], [30], when used for infilling missing data. We consider the *metabolite* dataset of [32]. The dataset consists of 154 vectors of length 52, which are log-ratios of the concentrations over time (compared to a baseline) of 154 metabolites in a cold-stress experiment. We randomly assigned a percentage of the datapoints (independently at random) to be missing, and calculated the mean absolute reconstruction error for the two models. We varied the percentage of missing data to be 10%, 20%,. . . , 80%, and plot the results of 100 trials for each percentage in Figure 5. Both methods perform better as one increases the number of latent dimensions (due to the built-in Automatic Relevance Determination through the priors on the scales Λ). For the Variational Bayes method we choose the best performance by fixing the number of latent dimensions to the maximum of 52. For the Stiefel sampling model we made do with just 7 latent dimensions. Despite this handicap, the Stiefel sampling model does significantly better than Variational Bayes when 50% or more of the data is missing. This result accords with that of Tobamovirus data: as uncertainty increases, fully Bayesian treatments excel.

3) Neural spike data: The neural spike data comes from a non-spatial sequential memory experiment on rodents [3]. Neural activity was recorded in the hippocampi of 6 rats, who had previously been trained on a particular "correct" sequence (A, B, C, D, E) of odors. Each trial involves the rat smelling one of the five odors through a port. The rat signals whether the odor is *in sequence* (InSeq) or *out of sequence* (OutSeq). It does this by choosing to withdraw its nose from the port either after or before one second, signalling InSeq or OutSeq respectively.

Note that with the "correct" sequence (A, B, C, D, E), each presented odor is InSeq. Whereas with the sequence (A, B, C, C) the first three odors are InSeq while the last odor (the repeated odor "C") is OutSeq. About 88% of the trials were in sequence. In this section we only look at data from a single session featuring rat Super Chris. The session consists of 249 trials lasting anywhere from 0.48 to 1.74 seconds each. The data features spike counts from 53 neurons and a binary indicator for whether the present odor is InSeq (1) or OutSeq (0). In order to minimize differences in motor neuron activity across trials, the spike counts for each trial are the total number of spikes in the 0.4 second interval immediately preceding port withdrawal. We are interested in a supervised learning problem: can we decode the rat's response (InSeq vs OutSeq) from the spike data alone?

We use the Poisson-Logistic joint model (11) – (14). Spike count vectors $X_t \in \mathbb{R}^{53}$ are modeled as conditionally independent Poisson, with log rate vector $U\Lambda z_t + \mu$ assumed to have rank 5 – that is, the latent factors z_t are in $\mathbb{R}^5$. The binary in-sequence, out-of-sequence variable y_t is modeled using logistic regression on the latent factors z_t. Hence the latent factors play two roles: they explain the majority of variation in spike counts *and* they predict for sequence status.

The logistic regression parameters β are of particular interest. These coefficients directly relate patterns in spike count to whether a sequence is correctly ordered. Their distributions may support the scientific hypothesis that the rat hippocampus is a place where sequential learning is performed. Here *learning* is meant to suggest a global phenomenon, one involving relationships between individual neurons and groups thereof. Figure (6) affirms the hypothesis in a specific sense: if a coefficient has a significantly non-zero posterior sample, then intensity of the relevant factor corresponds to the increased or decreased odds of sequential correctness.

Figure (6) features 500 draws from the posterior distributions of the logistic coefficients associated with the five latent factors. We simulated 10,000 samples, discarded the first half, and thinned nine of every ten draws. The first parameter has a distinctly non-zero posterior, while the rest do not. We infer that the first latent factor has a statistically significant association with sequential correctness. Moreover, the strictly negative posterior suggests that this association is in fact negative.

Besides providing interpretable regression coefficients, the joint model outperforms competitors with respect to prediction accuracy. Table 1 shows prediction error rates for a number of methods under (0-1) loss. All methods use spike count data or a reduction thereof to predict sequential correctness. As 88% of odors are presented in-sequence, uniformly predicting in-sequence earns the low error rate of 0.12. PLS-DA predictions are made by first performing PLS-DA [4] for dimensionality reduction then running one of the respective prediction methods on the reduced data. Only the Bayesian joint model is able to correctly predict a significant fraction of out-of-sequence odors, achieving an error rate below 0.08.

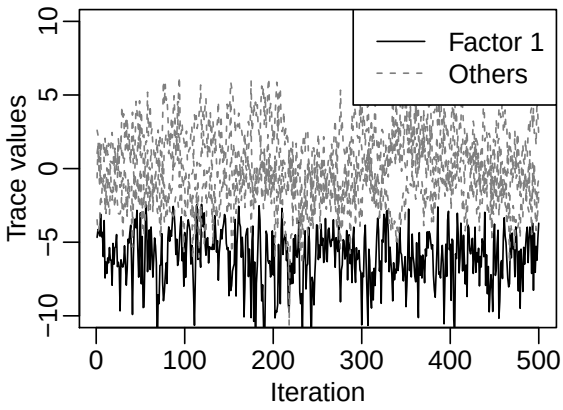

Fig. 6. The posterior trace plots of logistic regression parameters associated with the first latent factor (black) and remaining minor factors. Created using 500 geodesic Monte Carlo samples

TABLE I
10-FOLD CROSS-VALIDATION ERROR

Method	0-1 Error
Bayesian joint model	0.076
Random forest	0.103
PLS-DA SVM	0.111
PLS-DA 5NN	0.119
PLS-DA LDA	0.124

VI. DISCUSSION

We used geodesic Monte Carlo on the Stiefel and Grassmann manifolds to extend Bayesian analysis to linear dimensionality reduction models for high-dimensional data. By reparameterizing earlier versions of (exponential-family) PPCA and factor analysis, we demonstrated dramatically more efficient sampling. We showed how to perform geodesic Monte Carlo on the Grassmann manifold and demonstrated use of the Grassmannian pF distance for diagnosing convergence of both Stiefel and Grassmann manifold-valued parameters. We compared our manifold parameterized models to maximum likelihood counterparts and state-of-the-art Bayesian implementations, with favorable results. And in the the context of neural spike trains we demonstrated how the manifold parameterization allows for efficient Bayesian analysis of more complicated supervised dimensionality reduction tasks, resulting in superior prediction accuracy on held out data.

The above applications are in no way comprehensive.

Indeed one may use Bayesian inference on the Stiefel and Grassmann manifolds to probabilize many of the methods found in [13]. These new implementations will not necessarily resemble past iterations of probabilistic linear dimensionality reduction. We have shown that the geodesic Monte Carlo does not restrict dimensionality reduction to simple models but allows for inclusion into broader joint or graphical models. In turn, Bayesian dimensionality reduction becomes a tool for scientific inference as well as prediction.

REFERENCES

[1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. Riemannian geometry of grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematica*, 80(2):199–220, 2004.

[2] P.A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2009.

[3] Timothy A Allen, Daniel M Salz, Sam McKenzie, and Norbert J Fortin. Nonspatial sequence coding in ca1 neurons. *The Journal of Neuroscience*, 36(5):1547–1563, 2016.

[4] Matthew Barker and William Rayens. Partial least squares for discrimination. *Journal of chemometrics*, 17(3):166–173, 2003.

[5] John Barnard, Robert McCulloch, and Xiao-Li Meng. Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage. *Statistica Sinica*, pages 1281–1311, 2000.

[6] Christopher M Bishop. Bayesian pca. *Advances in Neural Information Processing Systems*, 11:382–388, 1999.

[7] Christopher M Bishop. Variational principal components. In *Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470)*, volume 1, pages 509–514. IET, 1999.

[8] Simon Byrne and Mark Girolami. Geodesic monte carlo on embedded manifolds. *Scandinavian Journal of Statistics*, 40(4):825–845, 2013.

[9] Carlos M Carvalho, Jeffrey Chang, Joseph E Lucas, Joseph R Nevins, Quanli Wang, and Mike West. High-dimensional sparse factor modeling: applications in gene expression genomics. *Journal of the American Statistical Association*, 2012.

[10] Joshua CC Chan, Roberto Leon-Gonzales, and Rodney W Strachan. Invariant inference and efficient computation in the static factor model. 2013.

[11] Yasuko Chikuse. *Statistics on Special Manifolds*. Springer, 2003.

[12] Michael Collins, Sanjoy Dasgupta, and Robert E Schapire. A generalization of principal components analysis to the exponential family. In *Advances in neural information processing systems*, pages 617–624, 2001.

[13] John P Cunningham and Zoubin Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *Journal of Machine Learning Research*, 2015.

[14] Wei Dai, Youjian Liu, and Brian Rider. Quantization bounds on grassmann manifolds and applications to mimo communications. *Information Theory, IEEE Transactions on*, 54(3):1108–1123, 2008.

[15] Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[16] A Gelman. Rstan: the r interface to stan. 2014.

[17] Mark Girolami and Ben Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

[18] Yuhong Guo. Supervised exponential family principal component analysis via convex optimization. In *Advances in Neural Information Processing Systems*, pages 569–576, 2009.

[19] Mehrtash Harandi, Richard Hartley, Chunhua Shen, Brian Lovell, and Conrad Sanderson. Extrinsic methods for coding and dictionary learning on grassmann manifolds. *International Journal of Computer Vision*, 114(2-3):113–136, 2015.

[20] P Hoff. rstiefel: Random orthonormal matrix generation on the stiefel manifold. *R package version 0.9, URL http://CRAN. R-project. org/package= rstiefel*, 2012.

[21] Peter D Hoff. Model averaging and dimension selection for the singular value decomposition. *Journal of the American Statistical Association*, 2012.

[22] Richard Arnold Johnson, Dean W Wichern, et al. *Applied multivariate statistical analysis*, volume 4. Prentice hall Englewood Cliffs, NJ, 1992.

[23] Arto Klami, Seppo Virtanen, and Samuel Kaski. Bayesian exponential family projections for coupled data sources. *arXiv preprint arXiv:1203.3489*, 2012.

[24] Jun Li and Dacheng Tao. Simple exponential family pca. *Neural Networks and Learning Systems, IEEE Transactions on*, 24(3):485–497, 2013.

[25] Kung-Yee Liang and Scott L Zeger. Longitudinal data analysis using generalized linear models. *Biometrika*, 73(1):13–22, 1986.

[26] Hedibert Freitas Lopes and Mike West. Bayesian model assessment in factor analysis. *Statistica Sinica*, pages 41–67, 2004.

[27] Meng Lu, Jianhua Z Huang, and Xiaoning Qian. Supervised logistic principal component analysis for pathway based genome-wide association studies. In *Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, pages 52–59. ACM, 2012.

[28] Shakir Mohamed, Zoubin Ghahramani, and Katherine A Heller. Bayesian exponential family pca. In *Advances in Neural Information Processing Systems*, pages 1089–1096, 2009.

[29] Radford M Neal. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2:113–162, 2011.

[30] Shigeyuki Oba, Masa-aki Sato, Ichiro Takemasa, Morito Monden, Ken-ichi Matsubara, and Shin Ishii. A bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19(16):2088–2096, 2003.

[31] Brian D Ripley. *Pattern recognition and neural networks*. Cambridge university press, 2007.

[32] Matthias Scholz, Fatma Kaplan, Charles L Guy, Joachim Kopka, and Joachim Selbig. Non-linear pca: a missing data approach. *Bioinformatics*, 21(20):3887–3895, 2005.

[33] Babak Shahbaba, Shiwei Lan, Wesley O Johnson, and Radford M Neal. Split hamiltonian monte carlo. *Statistics and Computing*, 24(3):339–349, 2014.

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

[35] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2009.

[36] Shipeng Yu, Kai Yu, Volker Tresp, Hans-Peter Kriegel, and Mingrui Wu. Supervised probabilistic principal component analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 464–473. ACM, 2006.