Fully Bayesian Dimensionality Reduction Modeling with GT-PPCA

Anonymous Author(s)
Affiliation
Address
email

Abstract

Probabilistic PCA (PPCA) is a probabilistic generative model used for finding low-dimensional representations of high-dimensional data. While PPCA is highly flexible in theory use in practice poses a number of challenges. This originates from the representations typically employed for the subspace based on orthonormality constraints on a matrix of parameters for PPCA. This hinders building complex PPCA-like models that are tractable and performing fully-Bayesian inference using modern algorithms like NUTS and ADVI. To address this challenge, we introduce a geometrically-motivated representation of orthonormal matrices in terms of unconstrained parameters using Givens Transforms (GT). Our unconstrained approach, which we refer to as GT-PPCA, allows for performing tractable fully-Bayesian inference on a wide-class of dimensionality reduction models. Our apporach allows for employing current inference algorithms such as NUTS and ADVI available in probabilistic programming frameworks like Stan. Our GT-PPCA approach also provides geometric insights into the latent variables allowing us to address important issues arising in Bayesian dimensionality reduction, such as tractable hierarchical modeling of subspaces and identifiability. We demonstrate how our GT-PPCA approach can be used in practice by developing hierarchical subspace models of medical patients and show how our approach can be used to capture the low-rank time-dependent dynamics within social networks. To facilitate use and adoption by the wider community, we also provide example codes in Stan for the GT-PPCA approach.

1 Introduction

Probabilistic PCA (PPCA) [16] posits a probabilistic generative model where high-dimensional data is determined by a linear function of some low-dimensional latent state. This generative model readily allows us to extend PPCA to describe the generative process of our data as we see fit. Notably, the PPCA generative model has been extended to obtain Sparse PCA [7], Bayesian Exponential PCA [12], Mixtures of Factor Analyzers [6], and Canonical Correlation Analysis (CCA) [14, chapt. 12.5]. This ability to extend PPCA is crucial for describing complex datasets arising in domains such as medicine where data may be stratified, consist of multiple views, or may vary over time. However, despite the demonstrated flexibility of PPCA, computational and statistical challenges that stem from the orthonormal matrix parameter of PPCA and related models stand in the way of making it a part of a routine modeling workflow. Ideally, one would be able to rapidly prototype diverse Bayesian dimensionality-reduction models in the framework of a probabilistic programming language like Stan [4], and subsequently infer these models using state-of-the-art, fully-Bayesian inference algorithms like NUTS or ADVI.

We draw on techniques from differential geometry and numerical analysis to introduce the Givens Transform, a novel representation of orthonormal matrices, that we use to easily build complex

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

Bayesian dimensionality-reduction models in Stan. We refer to this approach applied to PPCA as GT-PPCA. We provide Stan code for GT-PPCA and other example models allowing for use and expansion by researchers and scientists out-of-the-box. We also demonstrate how we use GT-PPCA in Stan in our own applied work to model and infer real-world data arising from complex generative processes, a task that was previously difficult and inaccessible to novices due to implementation constraints, as we explain in the next section. Specifically, we show how we use Stan's NUTS algorithm to obtain fully-Bayesian posterior inferences for a hierarchical subspace model of medical patients, as well as a Hidden Markov Model (HMM) that captures the low-rank time-dependent dynamics of a network of school children.

In addition to alleviating implementation challenges of Bayesian dimensionality reduction models, the Givens Transform, which represents orthonormal matrices in terms of a sequence of fundamental rotations through given angles, yields insight into novel and useful ways to work with and interpret these models, which in turn helps address previously unresolved issues. Specifically, the elegant geometric representation lets us see how by limiting the range of the parameters in GT-PPCA, we can naturally avoid issues of un-identifiability that arise when working with orthonormal matrices. GT-PPCA also allows us new and creative ways to generate and use prior distributions on orthonormal matrices, and thus subspaces, a task that had previously been rather complicated due to the difficulty of evaluating densities of orthonormal matrix distributions for even small problem sizes [7]. As we shall discuss in more detail, our GT representation provides a rather natural way to specify prior distributions comparable to the Matrix Langevin prior [13].

Related work While previous methods have been developed for posterior sampling of orthonormal matrices, they can at times suffer from numerical issues, are difficult to implement on large probabilistic graphical models, and are not generally adaptable to the widely available inference schemes such as NUTS or ADVI. Brubaker et al. [2] and Byrne and Girolami [3] developed different approaches to modify the Leap-Frog integrator typically used in Hamiltonian Monte Carlo (HMC), so that Hamiltonian exploration, and thus MCMC samples of posteriors, satisfied any necessary constraints at all times. Specifically, Brubaker et al. [2] uses the SHAKE integrator [10] to simulate Hamiltonian dynamics and generate proposals. The integrator works by repeatedly taking a step forward that may be off the manifold using ordinary leap frog, then projecting back down to the nearest point on the manifold. The projection is done via Newton iterations, which may converge to the wrong local minimum in practice or possibly not converge at all causing divergences [1]. While both of these possibilities jeopardize the ergodicity of a Markov Chain, and the integrity of samples, the former is more dangerous as it occurs without warning. On the other hand, Byrne and Girolami [3] exploit the fact that closed form solutions are known for the geodesic equations in the space of orthonormal matrices in the embedded coordinates, W, and use their derived Embedded Manifold HMC (EMHMC) sampler to perform posterior inference. While this method is completely explicit, requiring no Newton iterations, in practice we found that for larger step sizes, the integrator steps off the Stiefel manifold, due to the numerical imprecision of the matrix exponential function. Because these methods use modified integrators for constrained parameters, in practice they require additional book-keeping to track the support and the integrator of each variable, adding an extra layer of implementation complexity, especially for large complex models with many parameters. This makes it difficult to implement these methods within a probabilistic programming framework such as Stan, which typically does not expose the underlying inference algorithm to the user. Unfortunately, this precludes the rapid prototyping and building of models as well as the flexibility to use different inference algorithms that Stan provides. Lastly, we remark that for inference on orthonormal matrices, these methods can lead to multi-modal posteriors, that can be avoided in a straight-forward way using the Givens transform, as we discuss in sections 4.2 and 5.

Paper outline A brief overview of probabilistic dimensionality reduction is provided in Section 2. The geometry of the Stiefel Manifold is discussed in Section 3 and the Givens Transform (GT) is introduced in Section 4. Empirical studies where GT-PPCA was used in Stan for for building complex probabilistic graphical models are presented in Section 5.

2 Probabilistic Principal Component Analysis (PPCA)

PPCA posits the following generative process for how a sequence of high-dimensional data vectors $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, \cdots, N$ arise from some low dimensional latent representations $\mathbf{z}_i \in \mathbb{R}^p$ ($p < n$):

$$\begin{aligned} p(\mathbf{z}_i) &\sim \mathcal{N}_p(0, I) \\ p(\mathbf{x}_i|\mathbf{z}_i, W, \Lambda, \sigma^2) &\sim \mathcal{N}_n(W\Lambda\mathbf{z}_i, \sigma^2 I). \end{aligned} \tag{1}$$

Here W is an $n \times p$ orthonormal matrix and Λ is a $p \times p$ diagonal matrix with positive elements. For simplicity, we have presented the case where the data has zero mean, but the more general case can also readily be considered [14, chapt. 12.1]. We note that in the limit of σ^2 going to zero, the maximum likelihood estimator $\hat{W}$ of W converges to the matrix given by classical PCA [16]. Inference for PPCA and extensions is usually conducted in practice by obtaining point estimates of W via Expectation Maximization (EM) [14, chapt. 12.2.5].

Extensions to PPCA As mentioned previously, PPCA as a generative model can be flexibly expanded in several ways to produce new models. To build a probabilistic sparse PCA, one can place a Laplace or Cauchy prior over the elements of W. Mohamed et al. [12] showed that we can model non-Gaussian data, $\mathbf{x}_i$, by replacing equation 1 with any distribution from the exponential family whose natural parameters are given by $\mathrm{Expon}(W\Lambda\mathbf{z}_i)$, where $\mathrm{Expon}(\cdot)$ is an appropriate link function. Ghahramani et al. [6] use a mixture of several W matrices for different regions of data space to extend PPCA to the non-linear case. GT-PPCA in the context of a probabilistic programming framework like Stan makes these extensions to the base PPCA model easy to implement, even for non-experts or people with little programming background.

Importance of the Orthonormality Condition The orthonormal constraint on the matrix W plays an important role in obtaining robust methods for making inferences in probabilistic PCA because it alleviates identifiability and numerical issues. If one were to relax the orthonormality constraint and naively conduct inference in the space of all $n \times p$ matrices, the likelihood function would assign identical probability to a whole equivalence class of matrices $W \sim V$ where the span is the same linear subspace $\mathrm{span}\{W\} = \mathrm{span}\{V\}$ (see Murphy [14, chapt. 12.1.3]). In addition to this resulting in an unidentifiable model, in practice this presents a number of major challenges. First, matrices in a given equivalence class are not all equally well-conditioned numerically, thus round-off errors and truncation errors become problematic in practical calculations. Secondly, these issues with the representation further manifest in the log-likelihood objective function where regions of particularly large curvature arise, as pointed out by Holbrook et al. [8]. We note that, while most identifiability issues and numerical issues are alleviated by constraining inference to orthonormal matrices, the PPCA likelihood is equivalent for an orthonormal matrix W and any permutation of the columns of W being negative [14, 8, chapt. 12.1.3]. As such, even the methods of Brubaker et al. [2] and Byrne and Girolami [3] will lead to multi-modal posteriors that can be avoided in a straightforward manner by appealing to insights revealed by the Givens Transform, as shown in Sections 4.2 and 5.

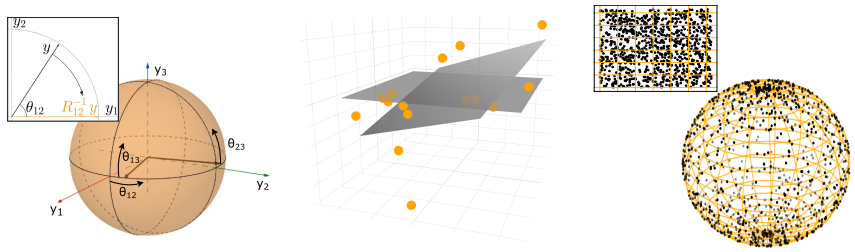
3 Geometry of the Stiefel Manifold

The orthonormality of W in Equation (1) introduces the challenge of having to work with prior and posterior distributions over orthonormal matrices. To make better sense of these types of objects, as well as the space of orthonormal matrices we must analyze their geometric properties.

Just as three-dimensional unit vectors form a sphere in $\mathbb{R}^3$, which is a submanifold of $\mathbb{R}^3$, the set of $n \times p$ orthonormal matrices, form a sub-manifold in the space of general $n \times p$ matrices known as the Stiefel Manifold and denoted $V_{n,p}$[13]. $V_{n,p}$ is formally defined as

$$V_{n,p} := \{Y \in \mathbb{R}^{n \times p} : YY^T = I\}. \tag{2}$$

The elements of $V_{n,p}$ can be thought of as p-frames, a collection of p orthonormal vectors that lie in n-dimensional space. To move about the Stiefel manfiold, one can rigidly rotate the vectors in the p-frame about any combination of axes an arbitrary number of times. The case where $n = 3$ and $p = 1$ corresponds to a sphere, while the case where $n = 3$ and $p = 2$, has an extra angle, θ_{23} that controls how much the second basis vector is rotated about the first (Figure 1a). Put in this



(a) (b) (c)

Figure 1: Visualizing the Givens Transform. (a) To obtain different elements of the Stiefel Manifold we rigidly rotate p-frames. This motivates the connection to Givens Reductions which work by rotating in some plane (inset) (b) PCA finds the orthonormal matrix in the Stiefel Manifold that best describes the subspace the data lie in, although in this case the point estimates is misleads us from the true subspace, which in this case is flat. (c) Even if we sample uniformly in angle coordinates (inset), without a proper measure adjustment samples will not be uniform when transformed to the sphere. In this case, samples are sparse near the equator and congregate near the poles.

perspective, for a three-dimensional set of points forming a flat, pancake-like cloud, PCA can be thought of as finding the best 2-frame that aligns with this cloud Figure 1b.

Even though $n \times p$ orthonormal matrices are typically represented by np elements, the intrinsic dimension of the Stiefel Manifold, $V_{n,p}$, is actually $np - p(p+1)/2$. This arises from the constraints on the columns of the matrix that impose orthonormality. This dimensionality can be seen by observing, that the first column of $Y \in V_{n,p}$ must have norm one and hence has one constraint placed on it. The second column must also have norm one and also must be orthogonal to the first column hence with two constraints placed on it. Continuing to the third column through the n^{th} one arrives at the conclusion that each point of the Stiefel Manifold has only $np-(1+2+\cdots+p) = np-p(p+1)/2$ degrees of freedom. The reduced dimensionality motivates the Givens Transform, which can be thought of as an $np - p(p+1)/2$-dimensional set of coordinates Θ, that represent elements of the Stiefel manifold.

4 Givens Transform (GT) approach to PPCA (GT-PPCA)

Posterior distributions for several types of constrained parameters are routinely inferred in practice using both MCMC and VI by transforming the constrained variables to an unconstrained space. Using a smooth one-to-one mapping, $T : \text{supp}(z) \to \mathbb{R}^D$, where $\text{supp}(z)$ is the support of the constrained random variable z, one can obtain a posterior over the unconstrained parameter that corresponds to the original constrained parameter of interest, then map inferences back to the original constrained space. This procedure requires computing the Jacobian, $J_{T^{-1}}$ of the transformation, to obtain $f_Y(y) = f_z(T^{-1}(y))J_{T^{-1}}(y)$ where Y is an unconstrained random variable with probability density function (PDF) f_Y and f_z is the probability density of z, which for PPCA comes from (1). The extra Jacobian term accounts for how the a unit volume under the transformation changes [9]. Without this extra Jacobian factor, inference between the two spaces is incomparable. As an example, in Figure 1c, we show that uniformly sampling in spherical coordinates (unconstrained space) does not correspond to uniformly sampling on the sphere (constrained space), unless we include an appropriate term accounting for how volumes are warped under the transformation. Intuitively, areas that are near the poles are shrunk far more than areas near the equator, so when mapped back on to the sphere, points will congregate closer to the poles of the sphere than the equator. Perfoming inference in a transformed space is most notably used in ADVI and Stan's HMC routines [4, 9]. To our knowledge no such transform has been proposed for orthonormal matrices.

4.1 Givens Reductions and the Givens Transform

Appealing to the Givens Reduction motivates a transformation from $n \times p$ orthonormal matrices, a constrained space, to unconstrained angles, that we call the Givens Transform. The Givens Reduction is a numerical analysis technique for reducing a square matrix A to upper-triangular form [11]. The

technique works by applying a series of rotation matrices to A such that elements below the diagonal are "zeroed out" starting with the second element of the first column, and moving down the first column before zeroing out the appropriate elemnents of the subsequent columns. For example if A is a 3×3 matrix and its first column is $(0.84, 0.48, 0.26)^T$, the Givens Reduction would would apply a rotation in the $(1, 2)$-plane, R_{12}^{-1} so as to annihilate the second element of this column producing $(0.93, 0.00, 0.26)^T$ (Figure 1a Inset). For an $n \times p$ orthonormal matrix Y, obtaining the Givens Reduction requires multiplication by $(n-1) + (n-2) + \cdots + (n-p) = np - p(p+1)/2$ rotation matrices each with their own respective angles. Since each rotation rigidly rotates the orthonormal columns of Y, they remain orthonormal, resulting in the matrix $I_{n,p}$, whose columns are the first p columns of the $p \times p$ identity matrix:

$$(R_{pn}^{-1} \cdots R_{p,p+2}^{-1} R_{p,p+1}^{-1}) \cdots (R_{2n}^{-1} \cdots R_{24}^{-1} R_{23}^{-1})(R_{1n}^{-1} \cdots R_{13}^{-1} R_{12}^{-1}) Y = I_{n,p}, \tag{3}$$

Because rotations are invertible we can rewrite (3) as

$$Y = (R_{12} \cdots R_{1n}) \cdots (R_{23} \cdots R_{2n})(R_{p+1,n} \cdots R_{pn}) I_{n,p}. \tag{4}$$

which we refer to as the Givens Representation of an orthonormal matrix Y. Since each of the $np - p(p+1)/2$ rotation matrices have an associated angle $(\theta_{12} \cdots \theta_{1n}) \cdots (\theta_{23} \cdots \theta_{2n})(\theta_{p+1,n} \cdots \theta_{pn})$, that we collectively refer to as Θ, we can use these angles to represent any $n \times p$ orthonormal matrix. *In this way we have reparameterized all $n \times p$ orthonormal matrices [1], a constrained space, in terms of unconstrained angles [2], using the Givens Transform* $\Theta : V_{n,p} \to \mathbb{R}^{np - p(p+1)/2}$. In a probabilistic programming framework like Stan we can treat Θ as an unknown parameter and $Y(\Theta)$ as a transformed variable, allowing us to use an orthonormal matrix in arbitrary likelihoods, such as the likelihood from (1). We also mention that multiplication by rotation matrices are inexpensive to compute as they are highly sparse (especially in large dimensions) and when applied to a matrix, they only modify two rows of that matrix at a time.

4.2 Geometry of the Givens Transform

We address implementation details of the Givens Transform relevant in practical use. Topologically, $V_{n,p}$ is locally equivalent to Euclidean space, but not globally equivalent, meaning it is impossible to find a one-to-one map between the Stiefel manifold and Euclidean space. Technically speaking, the Givens transform can map angles to all of $V_{n,p}$ except for a subset $S \subset V_{n,p}$ of measure zero. In the $n = 3, p = 1$ case (the sphere), this corresponds to a sliver where $\theta_{12} = \pi$ and $\theta_{13} \in (-\pi/2, \pi/2)$. Since the set is of measure zero, with probability one, the orthonormal matrix that describes the true subspace our data lies in will not be in that set.

In practice, we limit the angle θ_{12} to an interval of length π rather than an interval of length 2π to avoid superfluous modes in our posterior. This is useful both for interpretation and in higher dimensional problems where the number of modes grows exponentially. Examining the angles of the Givens transform reveals how geometrically, two matrices whose columns may be the negation of each other can both be reached. Because these matrices will have equivalent likelihoods under Equation 1, infering over the full range of angles results in a multi-modal posterior that makes sampling and VI more difficult to perform and interpret. To avoid this multi-modality using the modified integrator methods would require a mechanism to avoid boundaries, which are not as intuitively defined in the default embedded coordinates as in the Givens transform.

Lastly, we note that if the true basis lies near a pole, i.e. θ_{ij} is close to $-\pi/2$ or $\pi/2$, then posteriors will tend to be multi-modal as the region in parameter space close to the boundaries will be nearly equally valid, while the region near zero, will not be valid and thus contain little probability mass. In these cases, one can simply change the coordinate bounds (chart) so that $\theta_{ij} \in (0, \pi)$ will have a uni-modal posterior in the new coordinate system, alleviating numerical issues. In Stan this is straightforward, as one simply has to change the lower and upper bound of the angle parameter.

An analogous Jacobian term using differential forms As stated earlier, performing inference on a transform space requires a Jacobian term accounting for how volumes are warped by the transform,

[1] other than a set of measure zero, that is thus negligible

[2] the angles are themselves constrained to lie in certain intervals e.g. $[0, \pi)$ but these sorts of constraints are routine to deal with using a one-to-one diffeomorphism between intervals and the real line e.g. the sigmoid transform

but in the case of the Givens Transform this poses a problem because an $n \times p$ orthonormal matrix is np-dimensional and the Givens transform, $\Theta(Y)$, maps this set to an $np - p(p+1)2$-dimensional set of angles Θ. In this more general scenario, one can not simply take the determinant of the Jacobian as the volume morphing factor, because the Jacobian is not square and hence the determinant is undefined. To obtain the correct factor one must appeal to the calculus of differential forms. For accessibility, we provide psuedo-code in the supplementary materials, as well as actual Stan code to illustrate how one actually computes a differential form on a computer.

Differential forms measure how a transform warps an infinitesimal volume from one space to another, and they can be applied irrespective of the coordinates we use to describe either space. For example, spherical coordinates $(\theta, \varphi) \mapsto (x(\theta, \varphi), y(\theta, \varphi), z(\theta, \varphi))$ map points in the flat plane, $\mathbb{R}^2$, to points in $\mathbb{R}^3$ that lie on the sphere. $d\theta \wedge d\varphi$ represents a small area in the plane that can be rewritten as a small patch in $\mathbb{R}^3$ by finding $d\theta$ and $d\varphi$ in terms of dx, dy, and dz and applying the well defined rules of a wedge product. For a thorough account we recommend Muirhead [13], Edelman [5], or any standard text in differential geometry.

We can analogously use differential forms and wedge products to measure volumes on the Stiefel manifold. For $n \times p$ orthonormal matrices, there are only $np - p(p+1)/2$ free parameters and so the proper form to measure sets of orthonormal matrices is a $np - p(p+1)/2$-form. For an orthonormal, $n \times p$ matrix, Y, we can find an orthonormal $n \times n$ matrix G such that $G^T Y = I_{n,p}$. In fact G just comes from the product of the appropriate rotation matrices that arises in the Givens Reduction (Equation 3). Muirhead [13] shows that the correct form for measuring volumes on the Stiefel manifold comes from wedging the elements of the $n \times p$ matrix $G^T dY$ that lie below the diagonal i.e.

$$\bigwedge_{i=1}^{p} \bigwedge_{j=i+1}^{n} G_j^T \, dY_i \tag{5}$$

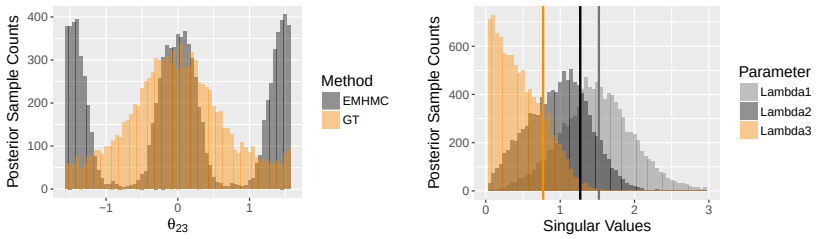
where G_j is the jth column of G and Y_i is the ith column of Y. To obtain the form in angle coordinates, we obtain dY_i in terms of the angle coordinates by the following relationship, $dY_i = J_{Y_i}(\Theta)\, d\Theta$, where J_{Y_i} is the Jacobian of Y_i with respect to the angle coordinates. Once we obtain the form (5) in terms of the angle coordinates, the result is a wedge product of $np - p(p+1)/2$ vectors that are $np - p(p+1)/2$ dimensional, which reduces to the determinant of these vectors aligned side by side as a $np - p(p+1)/2 \times np - p(p+1)/2$ matrix. This determinant is analogous to and serves the same purpose as Jacobian adjustment that comes from transforming random variables. We can insert it in to the log-probability of a model to avoid the sort of unintended sampling behavior depicted in Figure 1c. We incorporate the form (5) in to the log-probability of all of our Stan examples.

5 Empirical Studies

5.1 Synthetic Data

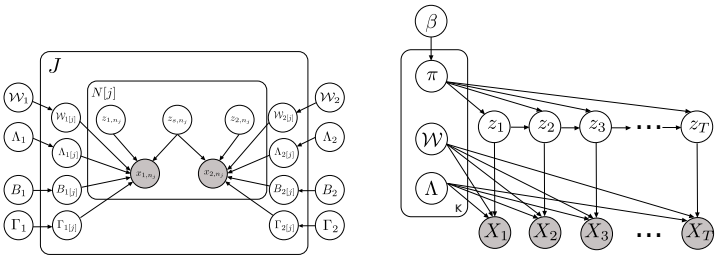
We apply GT-PPCA in Stan to a synthetic dataset to show how GT-PPCA naturally avoids multi-modal posteriors and provides plug-and-play access to Stan's powerful NUTS implementation. We generated a synthetic, three-dimensional dataset that lies on a two-dimensional plane with $N = 15$ observations according to 1 (data is pictured in Figure 1b). We chose $\mathrm{diag}(\Lambda) = \mathrm{diag}(1, 1)$, $\sigma^2 = 1$, and W to be $I_{3,2}$, which in the Givens representation corresponds to $\theta_{12} = \theta_{13} = \theta_{23} = 0$ i.e. the flat plane (Figure 1b). Our Stan code simultaneously provides posterior samples in both the W coordinates and the Givens angle coordinates. Examining posterior samples of the angle coordinates elucidates GT-PPCA's avoidance of superfluous modes, as compared to EMHMC (Figure 2a). Posterior samples of θ_{13}, which if we recall from Figure 1a is the Givens Transform angle that controls the upwards tilt of the plane, yields a median value of -0.24 and a 95% posterior interval of $(-1, 0.78)$, a wide range that is indicative of a lack of certainty given our data. Meanwhile, classical PCA yields the point estimate $\hat{\theta}_{13} = -0.15$, an estimate which in this case is overfit to the data (Figure 2a).

The fully Bayesian approach provided by GT-PPCA in Stan also allows us to examine posterior draws of Λ in Equation 1 to make probabilistic statements about the inherent dimensionality in our data. The posterior of Λ_3 for example places considerable mass close to zero (58% of samples were less that 0.5), providing strong evidence that our data is inherently two, not three, dimensional (Figure 2b). In comparison, the standard classical PCA analysis yields the singular values $\mathrm{diag}(\hat{\Lambda}) = (1.52,\ 1.27,\ 0.77)$, possibly, but ambiguously, suggesting that our data lie close to some



(a) (b)

Figure 2: Inferences for three-dimensional synthetic data. (a) By limiting the angle of rotation in GT-PPCA we can avoid the un-identifiability in our problem and eliminate multi-modal posteriors that show up in related methods. (b) Posterior draws from NUTS show that the third singular value, Λ_3 has a posterior that concentrates close to zero, indicating our model is most likely not three-dimensional given the data we have seen.



(a) (b)

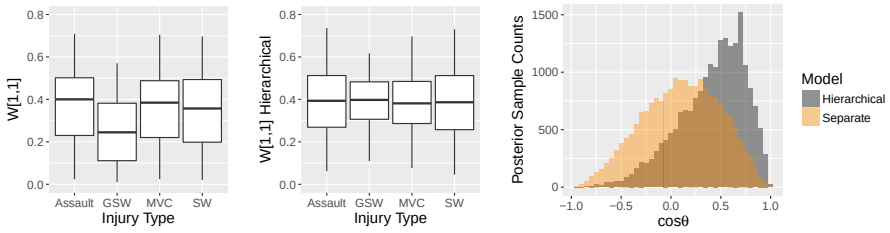
Figure 3: Probabilistic graphical models for (a) Hierarchical CCA Model (b) Network HMM.

two-dimensional plane, if one uses the heuristic that the third singular value has a larger drop off from the first two than the second has from the first.

5.2 Hierarchical subspace models for grouped multi-view medical data

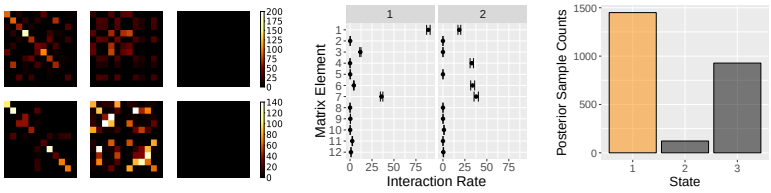
We modeled grouped multi-view hospital data for injured patients using a hierarchical CCA model [14, chapt. 15.2]. CCA can model two types (or views) of data as being a function of two respective latent low dimensional states, but also a common latent state that captures the common information contained in both view. In our case we compared blood protein measurements and clot strength measurements for injured patients belonging to one of four groups depending on the type of injury they had. While the four types of injuries were different enough so that we could not use a single CCA model to capture the characteristics of all models at once, the four groups were not so different as to warrant separate CCA models for each. To share information between the CCA models we placed a hierarchical prior over the orthonormal matrices belonging to the distinct CCA parameters of each group (Figure 3a).

While distributions on the Stiefel Manifold such as the Matrix Langevin distribution [13] exist, these distributions are difficult to use in practice, as computing their density requires evaluating an expensive matrix sum [7]. By appealing to the Givens Transform and placing a hierarchical prior over the angles of the different orthonormal matrices, we were able to build a hierarchical model over subspaces, a previously intractable task. The hierachical prior "shrinks" the posterior median of the orthonormal matrices towards a common mean in addition to reducing the variance of these estimates (Figure **??**). This is particularly helpful for groups with only a smaller number of observations such as the SW group, which only contains 16 patients in comparison of the GSW group with 86 patients. Comparing the angle between the first principal components for the SW and GSW group illustrates how using a hierarchical prior shrinks estimates of subspaces together towards a common hierarchical subspace (Figure 4).



(a) (b) (c)

Figure 4: Inferences for Hierarchical CCA model. (a) When estimated separately estimates of the matrix parameter W have high uncertainty. (b) Placing a hierarchical prior over these matrices with GT-PPCA shrinks these parameter to a common hierarchical mean and results in smaller posterior intervals. (c) Geometrically the respective first principal component of two different groups are shrunk closer together in a hierarchical model.



(a) (b) (c)

Figure 5: (a) Posterior modes of rate matrices for the three states (top) capture the pattern found in example count matrices belonging to each of these three states (bottom). (b) Posterior intervals from GT-PPCA with NUTS capture uncertainty in the orthonormal matrix estimates for the first two columns of the rate matrix for the first hidden state. (c) Posterior draws can tell us the posterior probability that the network was in a certain state given the data.

5.3 School Network

We built an HMM subspace for count data to model the hidden time-dependent structure of a network of school children. RF sensors were used to track the interactions between school children in 12 different classes (two classes for grades 1-6) for an entire school day so as to better understand how disease spreads throughout a network [15]. We collated the number of interactions between each pair of classes in to 11-minute contiguous time windows, giving us 177 symmetric matrices of counts representing the network structure between the different classes throughout the whole day (Figure 5a, lower row). We modeled the elements of these count matrices as each coming from a Poisson distribution whose rate comes from the element-wise exponential of a symmetric matrix $R = \exp(W \Lambda W^T)$, where the orthonormal matrix W captures the low-dimensional structure of the network. To model the time varying structure of the network, we posited that the network was always in one of three latent states, that evolve according to a Markov Chain (Figure 3b). The three states each have their own associated orthonormal matrix W_i that captures the low-dimensional latent network structure for that state.

Posterior modes capture the latent structure of the rate matrices of the three hidden state (Figure 5a top row), while NUTS sampling in Stan provides full Bayesian posteriors for each of the elements of these rate matrices (Figure 5b). Stan also allows us to generate samples from posterior samples of the Markov Chain, allowing us to provide a posterior over which of the hidden states the network is in at a given time (Figure 5c), a common inference task in disease networks as well as fMRI networks.

6 Discussion

We introduced the Givens Transform as a theoretical representation of orthonormal matrices as well as a practical tool for building complex Bayesian dimensionality-reduction models in a probabilistic framework like Stan. We showed using real-world examples how GT-PPCA allows out-of-the-box

usage of Stan's powerful suite of inference algorithms, as well as other practical benefits such as avoidance of multi-modal posteriors and access to new types of useful distributions using the angle representation. We provide Stan code for GT-PPCA and our other models allowing for fully Bayesian dimensionality reduction analysis without any implementation hassle.

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.

References

[1] Diagnosing biased inference with divergences. `http://mc-stan.org/documentation/case-studies/divergences_and_bias.html`. Accessed: 2017-05-11.

[2] Marcus Brubaker, Mathieu Salzmann, and Raquel Urtasun. A family of mcmc methods on implicitly defined manifolds. In *Artificial Intelligence and Statistics*, pages 161–172, 2012.

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

[4] Bob Carpenter, Andrew Gelman, Matt Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Michael A Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 20, 2016.

[5] Alan Edelman. 18.325: Finite random matrix theory volumes and integration. 2005.

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

[7] Peter D Hoff. Simulation of the matrix bingham–von mises–fisher distribution, with applications to multivariate and relational data. *Journal of Computational and Graphical Statistics*, 18(2): 438–456, 2009.

[8] Andrew Holbrook, Alexander Vandenberg-Rodes, and Babak Shahbaba. Bayesian inference on matrix manifolds for linear dimensionality reduction. *arXiv preprint arXiv:1606.04478*, 2016.

[9] Alp Kucukelbir, Rajesh Ranganath, Andrew Gelman, and David Blei. Fully automatic variational inference of differentiable probability models. In *NIPS Workshop on Probabilistic Programming*, 2014.

[10] Benedict Leimkuhler and Sebastian Reich. *Simulating hamiltonian dynamics*, volume 14. Cambridge University Press, 2004.

[11] Carl D Meyer. *Matrix analysis and applied linear algebra*, volume 2. Siam, 2000.

[12] Shakir Mohamed, Zoubin Ghahramani, and Katherine A Heller. Bayesian exponential family pca. In *Advances in neural information processing systems*, pages 1089–1096, 2009.

[13] Robb J Muirhead. *Aspects of multivariate statistical theory*, volume 197. John Wiley & Sons, 2009.

[14] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[15] Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-François Pinton, Marco Quaggiotto, Wouter Van den Broeck, Corinne Régis, Bruno Lina, et al. High-resolution measurements of face-to-face contact patterns in a primary school. *PloS one*, 6(8):e23176, 2011.

[16] 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.