Givens Transform (GT) Approach for Efficient Probabilistic Principle Component Analysis (PPCA) for Bayesian Dimensionality Reduction (GT-PPCA)

Anonymous Author(s)
Affiliation
Address
email

Abstract

We develop scalable and flexible Probabilistic Principal Component Analysis (PPCA) methods for determining posterior distributions of spanning frames based on a Givens Representation of the PCA which we term (GT-PPCA). This addresses significant challenges that arise with latent variable in a traditional formulation of PPCA. For sampling posterior distributions we develop Hamiltonian Monte-Carlo Methods (HMC) for sampling on the Stiefel Manifold the PCA orthogonal frame sets. We demonstrate our approach on several challenging example problems including tests problems XYZ and problems arising in our recent work on understanding medical patient data associated with coagulopathy (factors influencing blood clotting). We show our methods provides ways to identify when data sets contain a mixture of low dimensional structures that would not be resolved with traditional PCA approaches. We further show how our approach can be used to develop heirarchical models in terms of low dimensional structures learned from the data sets or to develop prior distributions useful in generalizing low dimensional structures to new settings. To facilitate use of our GT-PPCA method we provide a package with the widely-used Stan statistics package.

1 Introduction

Principal Component Analysis (PCA) is a widely used dimensionality-reduction tool for exploratory analysis and modeling in both the natural and social sciences. By factorizing an empirical covariance matrix into a product of low rank matrices, PCA effectively finds a low dimensional subspace that describes the dataset in terms of the latent factors. These factors are given by the columns of the low rank factorization. Geometrically, traditional PCA can be interpreted as providing a point estimate of a low dimensional hyper-plane that is closest to a cloud of data points. For binary or integer valued matrices, such as graph adjacency matrices arising in network science, Matrix Factorization (NMF) [7] or Exponential Family PCA (EPCA) [4] are often used to find such low dimensional latent factors.

Probabilistic PCA (PPCA) [13] and Bayesian Exponential Family PCA (BXPCA) [10], posit probabilistic generative models that are equivalent to PCA in the limit of decreasing noise [12, chapt. 12.2]. This probabilistic approach is attractive because it allows a straightforward way, via Bayesian inference, to quantify the uncertainty in our estimates (to prevent overfitting) and conduct hypothesis testing, as is often used in the sciences. Furthermore, probabilistic models are amenable to expansion and can serve as modules within larger probabilistic graphical models. This is important in a real-world setting where true generative models are often complex and where they do not neccessarily follow the simple generative process set forth by PPCA. We illustrate both of these properties in our examples Section (ref).

In practice, conducting full Bayesian inference on PPCA and related models involves one or more unknown orthonormal matrices. This is difficult because orthonormal matrices form a rather particular subset out of all possible realizable matrices yielding a probability measure on a manifold within the full space with a non-trivial geometry. More specifically, the prior and posterior distributions of an orthornormal matrix W must have support over the set of $n \times p$ orthonormal matrices. This is known as the Stiefel manifold and denoted $V_{n,p}$ (cite). When first considering this problem, it may seem that this geometry may rule out a straight-forward use of two of the most prominent techniques for posterior inference posterior inference, Markov Chain Monte Carlo (MCMC) and Variational Inference (VI). Intuitively, for MCMC, we have no way of guaranteeing that a chain over W will explore only valid regions of parameter space that satisfy the orthonormality constraints. Similarly for Variational Inference, positing a common variational posterior distribution such as a Gaussian over the elements of W are sure to lead to posteriors that assign mass to invalid regions of parameter space.

To address these issues, there has been some recent work on posterior inference of orthonormal matrices based on modified versions of Hamiltonian Monte Carlo (HMC) with sampling for constrained parameters (cite). These methods use modified numerical integrators to ensure that exploration only occurs in valid areas of parameter space where the necessary constraints are satisfied (cite). Because these methods use modified integrators for constrained parameters, they require approaches in practice for keeping track of the type of each variable and for which type of integrator to use on each variable. This adds an extra layer of algorithmic complexity, especially for large complex probabilistic graphical models. This further makes it difficult to implement these methods in a scalable fashion and within widely used probabilistic programming languages and packages, such as Stan [3]. Furthermore, we have found while these methods represent some progress on this challenging problems the numerical methods are sometimes not rigorous or robust and at times can suffer from errors and instabilities that jeopardize the integrity of samples. In addition, we have found these methods can at times induce unnecessary multi-modality that are an artifact of the chosen respresentation and not intrinsic to the model under study. This arises in part from the choice of representations directly in terms of unconstrained matrices in these methods (cite). As a consequence, the matrix in the PPCA model has equal liklihood when making an arbitrary change of sign of one of its columns. Given this equivalent relation this can result in significant multi-modality that must be handled with some care in the final analysis.

In the case of positively constrained parameters, posterior distributions are in practice rather routinely inferred using both MCMC and VI. This can be accomplished by transforming the constrained variables to an unconstrained space using the one-to-one mapping $f : \mathrm{supp}(z_{\mathrm{constr}}) \to \mathbb{R}^D$. The $\mathrm{supp}(z_{\mathrm{constr}})$ is the support of the constrained random variable z_{constr}. To our knowledge, no such transformation has been proposed in the past to map orthonormal matrices to a comperable unconstrained space.

In this paper, we introduce just such a transform based on the Givens Reductions of a matrix. This expresses the matrix in terms of a sequence of funamental rotations through given angles. We call our approach the Givens Transform (GT) PCA or (GT-PCA) for short. The transform allows for a straight-forward implementation of PPCA and related models that involve orthonormal matrices. This allows for use in stand-alone HMC implementations or in the context of a probabilistic programming languages like Stan (cite). We first discuss the details of our approach and then illustrate how this can be done using Stan. A particular advantage of using Stan is the ability to use the built-in NUTS and VI implementations. When we consider more complicated models in our examples Section (ref), we will show how the Givens Transform also provides through this alternative representation of orthonormal matrices useful ways to work with and interpret our models. For instance, by limiting the range of the Givens angle representation we can avoid naturally the issues involved with other representations that have unneccessary multi-modal posteriors. Furthermore, our alternative representation allows us new and creative ways to generate and use prior distributions on orthonormal matrices. In the setting of using the matrix directly this task has previously been rather complicated and rather intractable for even small problem sizes. This is linked to the difficulty of evaluating densities of orthonormal matrix distributions in other representations. As we shall discuss in more detail, our GT representation provides a rather natural way to specify prior distributions comperable to the Langevin prior (cite).

We give a brief overview of probabilistic dimensionality reduction in Section 2. We discuss how orthonormal matrices arise naturally for performing inference and introduce a precise description of the Stiefel manifold in Section 3. We also briefly review prior work on Bayesian dimensionality reduction using orthonormal matrices in Section 3. We introduce the Givens Transform (GT) and use for reductions to obtain our representations in Section 4. We also discuss practical consideration for GT in computing and using the representation in Section 4. We then show how our methods can be used in practice for Bayesian inference on a few probabilistic graphical models in Section (ref). This includes illustration of the basic aspexcts of the technique on XYZ and presenting a more advanced real-world application to patient data for factors affecting coagulapathy in Section (ref). We conclude discussing how our GT-PPCA methods can be utilized on other problems and by discussing our implimentation in Stan that can be utlized by others interested in our approach.

2 Probabilistic dimensionality reduction

In probabilistic principle component analysis (PPCA) one starts by considering a collection of data points in a typically high-dimensional vector space and seeks to find a posterior distribution over a reduced representations of the data using a much lower dimensional subspace. The central postulate is that for a data vector $\mathbf{x} \in \mathbb{R}^n$ there exists an unknown low-dimensional latent representation $\mathbf{z} \in \mathbb{R}^p$ where $p < n$, (ideally with $p \ll n$). The two representations are related to each other by a single unknown linear transformation $\mathbf{x} \to \mathbf{z}$. Mathematically, we consider a finite collection of sampled data vectors $\mathbf{x}^i \in \mathbb{R}^n$, $i = 1, \cdots, N$ and see and estimate of the subspace. Formally, in PPCA this consists of using the following generative process

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

The W is an $n \times p$ orthonormal matrix and Λ is a $p \times p$ diagonal matrix with positive elements. For simplicity in our presentation of PPCA, we have assumed here that the data has only zero mean but the more general case can also readily be considered (cite). We also remark that generative model could also be expanded to the case of non-Gaussian data by replacing equation 1 with an exponential family member whose natural parameters are given by $\mathrm{Expon}(W\Lambda\mathbf{z}^i)$ where $\mathrm{Expon}(\cdot)$ is an appropriate link function [10].

3 Bayesian inference of orthonormal matrices

3.1 Unidentifiability

If we assume momentarily that the matrix W need not be orthonormal, then as pointed out in Murphy [12, chapt. 12.1.3], the likelihood will assign identical probability density values to different values of W. In other words, without assuming orthonormality, the PPCA model is unidentifiable. While one can proceed with inference using HMC as in [10], as pointed out by Holbrook et al. [6] the unidentifiability in the model leads to high curvature areas of the log-posterior which itself leads to numerical issues for any sampler as well as slow mixing times, making inference with non-orthonormal matrices impractical.

3.2 Inference on the Stiefel manifold

To alleviate issues of unidentifiability, Murphy [12, chapt. 12.1.3] suggests forcing W to be orthonormal. However, as noted earlier, conducting full Bayesian inference on the space of orthonormal matrices is difficult because of the complicated constraints. We denote the set of $n \times p$ orthonormal matrices $V_{n,p}$. Formally we define

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

$V_{n,p}$ is known as the Stiefel manifold in differential geometry [11]. While elements of the Stiefel manifold can be described as $n \times p$ matrices with np elements, due to the constraint of the columns of the matrix being orthonormal, there are actually only $np - p(p+1)/2$ degrees of freedom on the Stiefel manifold. One can see this by observing, that the first column of $Y \in V_{n,p}$ must be of length one and hence has one constraint placed on it. The second column must also be of length one, but must also be orthogonal to the first column, and hence has two constraints placed on it, etc.

3.3 Related works using HMC

Recently, two approaches to posterior inference of orthonormal matrices and other constrained parameters have been proposed. These methods work by modifying the leap-frog integrator used to simulate Hamiltonian dynamics in HMC to ensure that any trajectory in parameter space remains on the Stiefel manifold. Brubaker et al. [1] uses the SHAKE integrator [8] to ensure trajectories through parameter space continuously satisfy the required constraints. 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. This projection is done via Newton iterations, which may not converge in practice, possibly jeopardizing the ergodicity of a Markov chain.

Byrne and Girolami [2] took a different approach, exploiting the fact that closed form solutions are known for the geodesic equations over the Stiefel manifold in the embedded coordinates, W. 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.

Both methods rely on applying different integrators to constrained parameters. This adds implementation difficulty in practice, as one must keep track of the type of each variable (unconstrained or constrained, and type of constraint), and apply the appropriate integrator. Additionally, we note that even using orthonormal matrices, the PPCA likelihood is equivalent for a matrix W and any permutation of the columns of W being negative. As such, these methods lead to multi-modal posteriors, that can be avoided in a straight-forward way, as we show in the next section, using the Givens transform.

4 Givens Transform

We provide a brief exposition on Givens Reductions, which motivate the Givens Transform, then describe the Givens Transform along with relevant practical considerations.

4.1 Givens Reductions

Define R_{ij} to be the $n \times n$ rotation matrix that performs a counter-clockwise rotation in the (i, j)-plane of $\mathbb{R}^n$, where $j > i$. In $\mathbb{R}^3$, there are three such matrices, R_{12}, R_{13}, and R_{23}. They perform counter-clockwise rotation of angle θ_{ij} in the (x, y), (x, z) and (y, z) planes respectively. Rotation matrices have the following key properties:

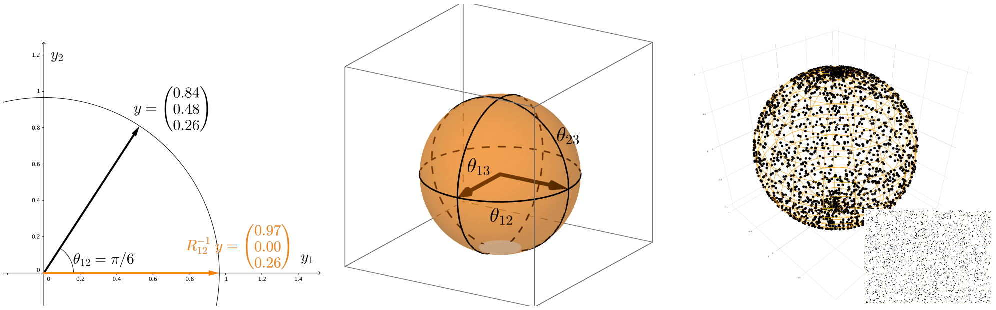
1. They preserve length and angles between vectors, i.e. for two vectors $u, v \in \mathbb{R}^n$, $R_{ij}u, R_{ij}v$ are the same length as u and v respectively, and if u and v are orthogonal then so are $R_{ij}u$ and $R_{ij}v$.
2. They are invertible and their inverse is their transpose $R_{ij}^{-1} = R_{ij}^T$. Their inverse corresponds to a clockwise rotation in the (i, j)-plane.

Now we consider an $n \times p$ matrix Y, with orthonormal columns. In general, the first column is a vector in $\mathbb{R}^n$ with a non-zero second element. However, we can apply an invertible clockwise rotation in the $(1, 2)$-plane, R_{12}^{-1}, to “zero out" the second element of the first column. Figure 1a depicts this visually for a 3D vector projected on to the $(1, 2)$-plane.

Similarly, we can apply consecutive rotations $R_{13}^{-1}, R_{14}^{-1}, \cdots, R_{1n}^{-1}$ to this result so that all entries of the first column besides the first element are zero. The first column of the resulting matrix, $R_{1n}^{-1} \cdots R_{13}^{-1} R_{12}^{-1} Y$, will be the length one vector $(1\, 0\, \cdots\, 0)^T$ which lies entirely along the first axis. If one takes the perspective that these rotations are applied to the columns of Y, then with the two properties of rotation matrices we mentioned earlier, it is evident that columns 2 through n must have zero in their first element because they will be orthogonal to the first column, $(1\, 0\, \cdots\, 0)^T$.

To “zero out" the elements of the second column, one can similarly apply rotations $R_{23}^{-1}, \cdots, R_{2n}^{-1}$. These rotations will leave the first column and first row unaffected, as the first column no lies entirely on the 1st axis, which these rotations do not involve. Continuing in this fashion yields

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


y_2
1.2
1
0.8
0.6
0.4
0.2
0
$y = \begin{pmatrix}0.84\\0.48\\0.26\end{pmatrix}$
$\theta_{12} = \pi/6$
$R_{12}^{-1}y = \begin{pmatrix}0.97\\0.00\\0.26\end{pmatrix}$
y_1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
θ_{23}
θ_{13}
θ_{12}

(a) (b) (c)

Figure 1: Visualizing the Givens Transform. (a) How the Givens Reduction "zeros out" a column vector. (b) A geometric view of the Stiefel manifold, two-frame in three dimensions. (c) Sampling without a proper measure adjustment.

where $I_{n,p}$ consists of the first p columns of the $n \times n$ identity matrix. This process of applying consecutive rotation matrices to a matrix is known in numerical analysis as the Givens reduction [9], and is applied more generally to square matrices for matrix-vector solves. In total we will have applied $(n-1)+(n-2)+\cdots+(n-p) = np - p(p+1)/2$ rotations matrices. We note that because rotation matrices are very sparse in high dimensions, multiplication by a rotation matrix is computationally much less intensive than matrix multiplication otherwise would be.

4.2 Givens Transform

Because rotation matrices are invertible, equation 3 implies that we can rewrite the $n \times p$ orthonormal matrix Y as the product of counter-clockwise rotation matrices and $I_{n,p}$:

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

Recall that 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. In this way we have reparameterized all $n \times p$ orthonormal matrices [1], a constrained space, in terms of unconstrained angles [2], using a transform $\Theta : V_{n,p} \to \mathbb{R}^{np-p(p+1)/2}$. We refer to 4 as the Givens representation or Givens transform.

4.2.1 A geometric perspective

The Givens transform can be visualized in the simple case when $n = 3$ and $p = 2$. This is the case where data is three-dimensional and we seek a two-dimensional subspace to represent the data. In this case, we can imagine a two dimensional bases, visualized as two perpendicular vectors that rotate rigidly together along three angles of rotations. This is referred to as a 2-frame in differential geometry [11] and is depicted in Figure 1b. Recall θ_{12} controls the amount of rotation in the $(1,2)$-plane, θ_{13} in the$(1,3)$-plane, and θ_{23} controls a rotation of the second basis vector about the first. All rotations will preserve the length and orthogonality of the basis vectors. Given a flat, pancake-like cloud of points in 3D, the two dimensional bases will move according to the appropriate angles to be aligned with the cloud of points, if one were conducting a (Maximum A-Posteriori) MAP estimation over the Stiefel manifold.

4.2.2 Practical considerations of topology and angles

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

[1] other than a set of measure zero we explain in the next subsection

[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

speaking, the Givens transform can map angles to all of $V_{n,p}$ except for a subset $S \subset V_{n,p}$, that in the $n = 3, p = 2$ case corresponds to a sliver when $\theta_{12} \in (-\pi, \pi)$, $\theta_{13} \in (-\pi/2, \pi/2)$, and $\theta_{23} \in (-\pi/2, \pi/2)$. Luckily this set is of measure zero (under the proper measure for the Stiefel manifold, see section 4.2.4), and thus, with probability one, the orthonormal matrix that describes the true subspace our data lie in will not be in that set.

In practice, we actually limit the angle θ_{12} to an interval of length π rather than an integral of length 2π, that traverses the entire Stiefel manifold. Examining the angles of the Givens transform makes it evident that in the latter case, two equivalent bases that are the negation of each other can be reached, resulting in a multi-modal posterior that makes sampling and VI more difficult and harder to 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 bases 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 close to 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 chart so that $\theta_{ij} \in (0, \pi)$, creating a uni-modal posterior in the new coordinate system, and alleviating numerical issues. In Stan this is straight-forward, as one simply has to change the lower and upper bound of the angle parameter.

4.2.3 Jacobian under a change of variables

The Givens transform 4 allows us to represent orthonormal matrices as angles $Y(\Theta)$. This in turn allows us to write probability densities $p_Y(Y)$ in terms of angles, so that we can conduct inference in an unconstrained space. It is well known in probability theory that the transform of a random variable is in general not the density of the transform, i.e. $p_\Theta(\Theta) \neq p_Y(Y(\Theta))$[12, chapt. 2.6]. To be more precise, densities are measured against volumes and integrated to get actual probabilities (otherwise known as probability mass). Under a transformation, densities are unaffected, but volumes (or rather the way in which volumes are measured) may change. This is important in the context of posterior inference, as not including the Jacobian adjustment would result in different priors than we intend.

Figure 1c depicts how samples that are uniform in the angle space are not uniform on the sphere. Samples congregate at the poles of the sphere because a patch of area in the angle space that is near the top corresponds to a very tiny patch of the sphere near the pole. In practice this would lead to posteriors that bias towards the poles, when what we really intend is a prior that is uniform on the Stiefel manifold.

For a K-dimensional random vector Y and a transformation $f : \mathrm{supp}(Y) \to \mathbb{R}^K$ the proper way to measure probability under a transform is by multiplying by the determinant of the Jacobian of the the inverse transform:

$$\int p_\Theta(\Theta)\, d\Theta = \int p_Y(Y(\Theta)) |\det J_{f^{-1}}(\Theta)|\, d\Theta. \tag{5}$$

In our case this poses a problem however, because an $n \times p$ orthonormal matrix is np-dimensional, but the Givens transform, $\Theta(Y)$, maps this set to a $np - p(p+1)2$-dimensional set. 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 even square and hence the determinant is undefined. To obtain the correct factor one must appeal to the calculus of differential forms.

4.2.4 Differential forms

We offer an intuitive high-level overview of differential forms. For a thorough account we recommend Muirhead [11], Edelman [5], or any standard text in differential geometry. The simplest non-trivial example of differential-forms between spaces of different sizes arises when trying to measure probablity using spherical coordinates. Spherical coordinates give us a map from $\mathbb{R}^2$ to the sphere, $(\theta, \varphi) \mapsto (x(\theta, \varphi), y(\theta, \varphi), z(\theta, \varphi))$. Although a sphere lies in 3-D space, if we have a density $p_{\mathrm{Euc}}(x, y, z)$ on the sphere, the "natural" way to measure the probability of a spherical random variable falling within some area on the surface of that sphere is by first covering that area with tiny rectangles tangent to the sphere, taking the average density of each rectangle, multiplying that density by the area of that rectangle, and summing over the resulting products. The probability of the random variable falling within that area is defined to be the limit of that result as the size of the rectangles go to zero. Area forms, or two-forms can be thought of as these infitessimal rectangles. They are written as the wedge product of two differentials, e.g. $dx \wedge dy$, and they are objects that are inserted into integrals to obtain measures. Differential forms form (no pun intended) a vector space useful for measuring areas and high-dimensional volumes. For example, the area forms on a sphere can be written as a linear combination of the standard-basis two forms

$$a(x, y, z)\, dx \wedge dy + b(x, y, z)\, dx \wedge dz + c(x, y, z)\, dy \wedge dz. \tag{6}$$

One can then integrate this over a sub-area of the sphere to obtain the measure of that sub-area. The beauty of differential forms is that a different coordinate system such as polar coordinates, simply correspond to using a different basis for representing forms, which again are vectors. In fact writing a form in a different bases simply involves taking partial derivatives, e.g. $dx = (\partial x/\partial\theta)\, d\theta + (\partial x/\partial\varphi)\, d\varphi$. If we substitute the forms in the angle bases in to 6 and simplify using the well-defined anti-symmetric and distributive properties of wedge products and differential forms (see [11]), we obtain a proper area-form 6 in spherical coordinates, $S(x(\theta, \varphi), y(\theta, \varphi), z(\theta, \varphi))\, d\theta \wedge d\varphi$, that can then be integrated using a double integral in polar coordinates. The absolute value of this area-form evaluated at some point specified in angle coordinates, (θ_0, φ_0), intuitively measures how much the area of tiny square in angle space gets shrunk or stretched when the area is mapped to an area on the sphere.

Analogously, we can 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 in fact 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 comes from the Givens Reduction 3. Muirhead [11] shows that the correct form comes from wedging the elements of the $n \times p$ matrix $G^T dY$ that lie below the diagonal i.e.

$$\bigwedge_{i=1}^{p} \bigwedge_{j=i+1}^{n} G_j^T\, dY_i \tag{7}$$

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 simply obtain dY in angle coordinates. dY_i can be obtained 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 7 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 7 in to the log-probability of all of our Stan examples.

5 Examples

5.1 Synthetic Data

Show how point estimation of subspace and dimensionality can be misleading. Show multi-modality of Symone's method.

5.2 FMRI

Illustrate hypothesis testing.

5.3 School Network

Mainly here to show we can do exponential families easy too. Possibly make this a change point detection?

5.4 Coagulopathy using hierarchical subspace models

Don't forget to mention sparse PCA, via Cauchy priors.

6 Discussion

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] 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.

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

[3] 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.

[4] Michael Collins, Sanjoy Dasgupta, and Robert E Schapire. A generalization of principal component analysis to the exponential family. In *Nips*, volume 13, page 23, 2001.

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

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

[7] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

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

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

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

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

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

[13] 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.

Appendix

$$R_{ij} := \begin{array}{cc}
\begin{array}{ccccccccccc} & & & i & & & & j & & & \end{array} & \\
\begin{pmatrix}
1 & & & & & & & & & & \\
 & \ddots & & & & & & & & & \\
 & & 1 & & & & & & & & \\
 & & & \cos\theta_{ij} & & & & -\sin\theta_{ij} & & & \\
 & & & & 1 & & & & & & \\
 & & & & & \ddots & & & & & \\
 & & & & & & 1 & & & & \\
 & & & \sin\theta_{ij} & & & & \cos\theta_{ij} & & & \\
 & & & & & & & & 1 & & \\
 & & & & & & & & & \ddots & \\
 & & & & & & & & & & 1
\end{pmatrix} &
\begin{array}{c} \\ \\ \\ i \\ \\ \\ \\ j \\ \\ \\ \\ \end{array}
\end{array}$$

$$R_{12}^{-1}Y = R_{12}^{-1}\begin{pmatrix}
y_{11} & y_{12} & \cdots & y_{1p} \\
y_{21} & y_{22} & \cdots & y_{2p} \\
y_{31} & y_{32} & \cdots & y_{3p} \\
\vdots & \vdots & \vdots & \vdots \\
y_{p1} & y_{p2} & \cdots & y_{pp} \\
y_{p+1,1} & y_{p+1,2} & \cdots & y_{p+1,p} \\
\vdots & \vdots & \vdots & \vdots \\
y_{n1} & y_{n2} & \cdots & y_{np}
\end{pmatrix} = \begin{pmatrix}
* & * & \cdots & * \\
0 & * & \cdots & * \\
y_{31} & y_{32} & \cdots & y_{3p} \\
\vdots & \vdots & \vdots & \vdots \\
y_{p1} & y_{p2} & \cdots & y_{pp} \\
y_{p+1,1} & y_{p+1,2} & \cdots & y_{p+1,p} \\
\vdots & \vdots & \vdots & \vdots \\
y_{n1} & y_{n2} & \cdots & y_{np}
\end{pmatrix}. \qquad (8)$$

$$(R_{1n}^{-1}\cdots R_{13}^{-1}R_{12}^{-1})Y = \begin{pmatrix}
1 & 0 & \cdots & 0 \\
0 & * & \cdots & * \\
0 & * & \cdots & * \\
\vdots & \vdots & \vdots & \vdots \\
0 & * & \cdots & * \\
0 & * & \cdots & * \\
\vdots & \vdots & \vdots & \vdots \\
0 & * & \cdots & *
\end{pmatrix}. \qquad (9)$$

$$(R_{2n}^{-1}\cdots R_{23}^{-1})(R_{1n}^{-1}\cdots R_{12}^{-1})Y = \begin{pmatrix}
1 & 0 & \cdots & 0 \\
0 & 1 & \cdots & * \\
0 & 0 & \cdots & * \\
\vdots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & * \\
0 & 0 & \cdots & * \\
\vdots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & *
\end{pmatrix}. \qquad (10)$$

Continuing in this fashion yields

$$(R_{pn}^{-1} \cdots R_{p,p+1}^{-1}) \cdots (R_{2n}^{-1} \cdots R_{23}^{-1})(R_{1n}^{-1} \cdots R_{12}^{-1})Y = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} = I_{np}. \qquad (11)$$