rotations: A Package for $SO(3)$ Data

by Bryan Stanfill, Heike Hofmann, Ulrike Genschel

Abstract In this article we introduce the **rotations** package which provides users with the ability to simulate, analyze and visualize 3-dimensional rotation data. The **rotations** package includes four distributions from which to simulate data, four estimators of the central orientation and a novel approach to visualizing these data. All of the above features are available to two different parameterizations of rotations: 3-by-3 matrix form and quaternions.

Introduction

Data in form of three-dimensional rotations find application in several scientific areas, such as biomedical engineering, computer visioning, and geological and materials sciences. A common goal shared by these fields is to estimate the main or central orientation for a sample of rotations. That is, letting the rotation group $SO(3)$ denote the collection of all 3×3 rotation matrices, observations $\boldsymbol{R}_1, \ldots, \boldsymbol{R}_n \in SO(3)$ can be conceptualized as a random sample from a *location model*

$$\mathbf{R}_i = \boldsymbol{S}\boldsymbol{E}_i, \quad i = 1, \ldots, n, \tag{1}$$

where $\boldsymbol{S} \in SO(3)$ is the *fixed* parameter of interest indicating an orientation of central tendency, and $\boldsymbol{E}_1, \ldots, \boldsymbol{E}_n \in SO(3)$ denote i.i.d. *random* rotations which symmetrically perturb $\boldsymbol{S}$.

The **rotations** package provides users with the tools necessary to simulate data from four common choices of symmetric distributions, estimate $\boldsymbol{S}$ in (1) and visualize a sample of rotations. The remainder of this paper is organized as follows. We will begin with a discussion on how rotation data can be parameterized. Then we discuss how data generation is possible in this package. Next we discuss the different estimators used to estimate the central direction. Finally, visualizations of rotation data is presented.

Rotation Representations

The variety of applications for rotations is echoed by the number of equivalent ways to parameterize them. We consider three of the most popular: matrices in $SO(3)$ and unit quaternions.

Matrix Form

Three-dimensional rotations can be represented by 3×3 orthogonal matrices with determinant one. All matrices with these characteristics form a group called the *special orthogonal group*, or *rotation group*, denoted $SO(3)$. Every element in $SO(3)$ can be described by an angle, $r \in [0, \pi)$ and an axis, $\boldsymbol{u} \in \mathbb{R}^3$ with $||\boldsymbol{u}|| = 1$. Thus $\boldsymbol{R} \in SO(3)$ can be thought of as rotating the coordinate axis, $\boldsymbol{I}_{3\times 3}$, about the axis $\boldsymbol{u}$ by the angle r. We adopt the material scientist's terminology in calling r the misorientation angle and $\boldsymbol{u}$ the misorientation axis.

More specifically, given an angle, r, and axis, $\boldsymbol{u}$, a 3×3 rotation matrix can be formed by

$$\boldsymbol{R} = \boldsymbol{R}(r, \boldsymbol{u}) = \boldsymbol{u}\boldsymbol{u}^\top + (\boldsymbol{I}_{3\times 3} - \boldsymbol{u}\boldsymbol{u}^\top)\cos(r) + \Phi(\boldsymbol{u})\sin(r) \tag{2}$$

where

$$\Phi(\boldsymbol{u}) = \Phi(u_1, u_2, u_3) = \begin{pmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{pmatrix}.$$

Quaternion Form

A second parameterization of rotations is with a *quaternion* of unit length. Quaternions are a form of imaginary numbers with one real entry and a vector of three imaginary parts that can be expressed as

$$q = x_1 + x_2 i + x_3 j + x_4 k$$

where i, j, and k are square roots of -1, i.e. $i^2 = j^2 = k^2 = -1$. We can write $q = (s, \boldsymbol{v})$ as tuple of the scalar s for coefficient $\mathbf{1}$ and vector $\boldsymbol{v}$ for the imaginary coefficients, i.e. $s = x_1$ and $\boldsymbol{v} = (x_2, x_3, x_4)$.

A rotation around axis $\boldsymbol{u}$ by angle r translates to $q = (s, \boldsymbol{v})$ with

$$s = \cos(r/2), \;\; \boldsymbol{v} = \boldsymbol{u}\sin(r/2)$$

This makes q a unit quaternion. The following code forms a unit quaternion from an angle sampled uniformly from the interval $[-\pi, \pi]$ and an axis sampled uniformly from the unit circle.

```
r <- runif(1, -pi, pi)
theta <- acos(runif(1, -1, 1))
phi <- runif(1, -pi, pi)
U <- c(sin(theta) * cos(phi), sin(theta) * sin(phi), cos(
R <- SO3(U, r)
q <- Q4(R)
```

For the remainder of this article we will present our work using the matrix representation. All the functions to follow will work with any of the representations we have discussed above.

Data generation

From (1), one can simulate a matrix $\boldsymbol{R}_i \in SO(3)$ by picking an axis $\boldsymbol{u}$ uniformly on the sphere then drawing an angle r from a distribution symmetric about 0 and bounded between $-\pi$ and π then applying (2).

A matrix generated in this fashion is said to belong to the *uniform-axis random spin*, or UARS, class of distributions and has the density

$$f(\boldsymbol{E}_i|\kappa) = \frac{4\pi}{3-\text{tr}(\boldsymbol{E}_i)} C\left(\cos^{-1}\left\{\frac{1}{2}[\text{tr}(\boldsymbol{E}_i)-1]\right\}\middle|\kappa\right) \tag{3}$$

where $C(\cdot|\kappa)$ is distribution function connecting to the angle of rotation r (Bingham et al., 2009). Members of the UARS family of distributions are distinguished based on the angluar distribution.

The **rotate** package allows the user access to four members of the UARS class. Each member is differentiated by the distribution function for r: the uniform distribution on the circle, the matrix Fisher (**????**), the Cayley (Schaeben, 1997; León et al., 2006) and a circular-von Mises-based distribution (Bingham et al., 2009).

The uniform distribution on the circle is given by the following density function

$$C_{\text{H}}(r) = \frac{1-\cos(r)}{2\pi} \tag{4}$$

for $r \in (-\pi, \pi]$. This function also plays the part of a measure on the sphere, called the Haar measure on the sphere. The function `rhaar` with input `n` will draw a sample of size n from (4) and `dhaar` with evaluate the density at a given point.

The following three distributions are all symmetric around 0 on the range $[-\pi, \pi)$ and have one parameter, κ, which is the concentration parameter. As κ increases, the distribution becomes more peaked about 0 and less variable. If one would prefer to specify the variability instead, the circular variance denoted ν can also be set by the user. For $r \sim F$ where F is a distribution on the circle, the circular variance is defined as $\nu = 1 - E\cos(r)$, and $E\cos(r)$ is called the mean resultant length (**?**).

The symmetric matrix Fisher distribution is the oldest and also the most difficult to sample from. It takes on the following distributional form

$$C_{\text{F}}(r|\kappa) = \frac{1}{2\pi[\text{I}_0(2\kappa) - \text{I}_1(2\kappa)]} e^{2\kappa\cos(r)}[1-\cos(r)]$$

where $\text{I}_{\text{p}}(\cdot)$ denotes the Bessel function of order p defined as $\text{I}_{\text{p}}(\kappa) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\cos(pr)e^{\kappa\cos r}dr$.

For a given κ, the function `rfisher` generates a sample of size n from this distribution using a rejection algorithm and `dfisher` evaluates the density at a given angle r.

León et al. (2006) proposed the symmetric Cayley distribution, which is identical to the de la Vallée Poussin distribution and a favorite among material scientists (Schaeben, 1997). This distribution is closely related to the beta distribution and has the distributional form

$$C_{\text{C}}(r|\kappa) = \frac{1}{\sqrt{\pi}}\frac{\Gamma(\kappa+2)}{\Gamma(\kappa+1/2)} 2^{-(\kappa+1)}(1+\cos r)^{\kappa}(1-\cos r).$$

`rcayley` simulates from this distribution by taking a simple transformation of random deviates from a beta distribution and `dcayley` evaluates the Cayley density at a given angle, r.

Finally the circular-von Mises-based distribution is included because the distribution is non-regular and has been applied to EBSD data (Bingham et al., 2009). An angle following this distribution has the distribution form

$$C_{\text{M}}(r|\kappa) = \frac{1}{2\pi\text{I}_0(\kappa)} e^{\kappa\cos(r)}.$$

Simulation from this distribution was developed by **?** and the function `rvmises` follows this procedure closely. Also, `dvmises` evaluates the density at a given angle, r.

Once an angular distribution has been chosen and a vector of n angles of rotation have been generated, the `genR` function with option `space="SO3"` creates an $n \times 9$ matrix representing a sample from the appropriate UARS member as demonstrated in the following code. If quaternions are required the options `space` can be changed to `"Q4"`.

```
Rs <- ruars(20, rcayley, kappa = 1)
matrix(Rs[1, ], 3, 3)
##         [,1]    [,2]    [,3]
## [1,]  0.6025  0.2661 0.75247
## [2,] -0.7338 -0.1863 0.65337
## [3,]  0.3141 -0.9458 0.08303
```

Here `rcayley(20,1)` simulates $r_1, \ldots, r_{20}$ from $C_{\text{C}}(r|\kappa = 1)$ then `genR` generates the matrices. Each row of `Rs` is an element in $SO(3)$, as demonstrated by `is.SO3`, in vector form with central orientation $\boldsymbol{I}_{3\times 3}$. Any other central orientation in $SO(3)$ is possible by changing the `S` option. If a central orientation not in $SO(3)$ is proposed, however, an error is returned.

$SO(3)$ data analysis

Given a sample of n observations $\boldsymbol{R}_1, \ldots, \boldsymbol{R}_n$ generated according to (1) we offer four ways to estimate the matrix $\boldsymbol{S}$, i.e. the central orientation. These estimators are either Riemannian- or Euclidean-based in geometry and either mean- or median-based. First we discuss how the choice of geometry affects distance.

The choice of geometry results in two different metrics to measure the distance between rotation matrices $\boldsymbol{R}_1$ and $\boldsymbol{R}_2 \in SO(3)$. Under the embedding approach, the natural distance metric between two random matrices in the Euclidean distance, d_E, is induced by the Frobenius norm

$$d_E(\boldsymbol{R}_1, \boldsymbol{R}_2) = \|\boldsymbol{R}_1 - \boldsymbol{R}_2\|_F, \tag{5}$$

where $\|\boldsymbol{A}\|_F = \sqrt{\text{tr}(\boldsymbol{A}^\top \boldsymbol{A})}$ denotes the Frobenius norm of a matrix $\boldsymbol{A}$ and $\text{tr}(\cdot)$ denotes the trace of a matrix. The Euclidean distance between two rotation

matrices corresponds to the shortest cord in $\mathcal{M}(3)$ that connects them. If $r \in [-\pi, \pi)$ denotes the misorientation angle in the angle-axis representation (2) of $\boldsymbol{R}_1^\top \boldsymbol{R}_2 \equiv \boldsymbol{R}_1^\top \boldsymbol{R}_2(r, \boldsymbol{u})$ (so that $\mathbf{tr}(\boldsymbol{R}_1^\top \boldsymbol{R}_2) = 1 + 2\cos r$), then $d_E(\boldsymbol{R}_1, \boldsymbol{R}_2) = 2\sqrt{2}\sin(|r|/2)$ holds.

By staying in the Riemannian space $SO(3)$ under the intrinsic approach, the natural distance metric becomes the Riemannian (or geodesic) distance, d_R, by which the distance between two rotations $\boldsymbol{R}_1, \boldsymbol{R}_2 \in SO(3)$ is defined as

$$d_R(\boldsymbol{R}_1, \boldsymbol{R}_2) = \frac{1}{\sqrt{2}} ||\mathrm{Log}(\boldsymbol{R}_1^\top \boldsymbol{R}_2)||_F = |r|, \qquad (6)$$

where $\mathrm{Log}(\boldsymbol{R})$ denotes the principle logarithm of $\boldsymbol{R}$ (i.e., $\mathrm{Log}(\boldsymbol{R}) = \mathrm{Log}(\boldsymbol{R}(\boldsymbol{u}, r)) = \boldsymbol{\Phi}(r\boldsymbol{u})$ in (2)) and $r \in [-\pi, \pi)$ is the misorientation angle of $\boldsymbol{R}_1^\top \boldsymbol{R}_2$. The Riemannian distance corresponds to the length of the shortest path that connects $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ *within* the space $SO(3)$. For this reason, the Riemannian distance is often considered the more natural metric on $SO(3)$; see Moakher (2002) for this discussion along with more details on exponential/logarithmic operators related to $SO(3)$.

We first consider estimators based on the embedding approach, which we call the projected estimators. The median-based estimator in this class is

$$\widetilde{\boldsymbol{S}}_E = \mathrm{argmin}_{\boldsymbol{S} \in SO(3)} \sum_{i=1}^{n} d_E(\boldsymbol{R}_i, \boldsymbol{S}). \qquad (7)$$

The function `median` with option `type="projected"` approximates $\widetilde{\boldsymbol{S}}_E$ and uses an adaptation of the Weiszfeld algorithm (**?**). The mean-based estimator is

$$\begin{aligned} \widehat{\boldsymbol{S}}_E &= \mathrm{argmin}_{\boldsymbol{S} \in SO(3)} \sum_{i=1}^{n} d_E^2(\boldsymbol{R}_i, \boldsymbol{S}) \\ &= \mathrm{argmax}_{\boldsymbol{S} \in SO(3)} \mathrm{tr}(\boldsymbol{S}^\top \overline{\boldsymbol{R}}) \end{aligned} \qquad (8)$$

and is computed by the function `mean` with option `type="projected"`. For an in-depth discussion of the algorithm used to compute this value consult Moakher (2002).

The intrinsic estimators minimize the first and second order Riemannian distances. The *geometric median* is

$$\widetilde{\boldsymbol{S}}_R = \mathrm{argmin}_{\boldsymbol{S} \in SO(3)} \sum_{i=1}^{n} d_R(\boldsymbol{R}_i, \boldsymbol{S}). \qquad (9)$$

An algorithm proposed by Hartley et al. (2011) is employed by the function `median` with option `type="intrinsic"`. The *geometric mean* is the L_2 analog of $\widetilde{\boldsymbol{S}}_R$ given by

$$\widehat{\boldsymbol{S}}_R = \mathrm{argmin}_{\boldsymbol{S} \in SO(3)} \sum_{i=1}^{n} d_R^2(\boldsymbol{R}_i, \boldsymbol{S}). \qquad (10)$$

The function `mean` with option `type="intrinsic"` implements an algorithm first proposed by Manton (2004) in estimating $\widehat{\boldsymbol{S}}_R$.

Confidence Regions

For the projected mean ($\widehat{\boldsymbol{S}}_E$), confidence regions are available through the `region` function. The `method` option specifies which method to be used to

Visualizations

In this section we introduce a method to visualize $SO(3)$ data via the the **ggplot2** package (Wickham, 2009). The function `plot` takes as input a $n \times 9$ matrix of $SO(3)$ observations and returns a visualization of one of the three columns. The user can specify which column to use with the `col` option, the default is one. If the data are highly concentrated in one part of the sphere then the the `toRange` option can be set to `TRUE` then the range of the plot is set to zoom in on the populated area.

Any of the four estimates of the central direction can be plotted along with a sample of rotations. To show all estmiates at once add the option `estimates_show="all"`. If only a few estimates are of interest then any combination of `"proj.mean"`, `"proj.median"`, `"riem.mean"` or `"riem.median"` are valid inputs. The estimators are indicated by shape. One can also center the data about any observation in $SO(3)$ by setting `center=S`. Typically take `center=mean(Rs)`.

```
Rs <- ruars(50, rcayley, kappa = 1)
plot(Rs, center = mean(Rs), show_estimates = "all")
```

```
## Error:  type needs to be one of
'projected' or 'geometric'.
```

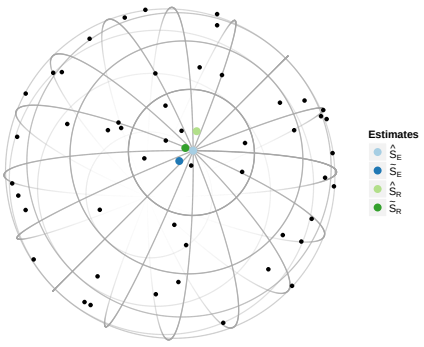

Figure 1: A plot of the a random sample from the von Mises-UARS distribution with $\kappa = 1$.

In Figure 1 a random sample of 50 matrices following the Cayley-UARS distribution with $\kappa = 1$ is plotted along with four estimates of the central orientation. The code to produce this plot is also given.

Summary

The **rotations** package is introduced and allows the user to create, simulate, analyze and visualize rotation data. There are three parameterizations possible in this package and four built-in distributions from which data can be simulated. The four estimators discussed in Stanfill et al. (2012) are each implemented and each can be visualized via `ggplot2`.

Bibliography

M. Bingham, D. Nordman, and S. Vardeman. Modeling and inference for measured crystal orientations and a tractable class of symmetric distributions for rotations in three dimensions. *Journal of the American Statistical Association*, 104(488):1385–1397, 2009.

R. Hartley, K. Aftab, and J. Trumpf. L1 rotation averaging using the weiszfeld algorithm. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3041–3048. IEEE, 2011.

C. León, J. M. é, and L. Rivest. A statistical model for random rotations. *Journal of Multivariate Analysis*, 97(2):412–430, 2006.

J. H. Manton. A globally convergent numerical algorithm for computing the centre of mass on compact lie groups. In *Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th*, volume 3, pages 2211–2216. IEEE, 2004.

M. Moakher. Means and averaging in the group of rotations. *SIAM Journal on Matrix Analysis and Applications*, 24(1):1–16, 2002.

H. Schaeben. A simple standard orientation density function: The hyperspherical de la vallée poussin kernel. *Phys. Stat. Sol. (B)*, 200:367–376, 1997.

H. Wickham. *ggplot2: elegant graphics for data analysis*. Springer New York, 2009. ISBN 978-0-387-98140-6. URL http://had.co.nz/ggplot2/book.

Bryan Stanfill
Department of Statistics
Iowa State University
Ames, IA 50011
stanfill@iastate.edu

Heike Hofmann
Department of Statistics
Iowa State University
Ames, IA 50011
hofmann@mail.iastate.edu

Ulrike Genschel
Department of Statistics
Iowa State University
Ames, IA 50011
ulrike@mail.iastate.edu