The Jacobian of the Fréchet mean on SPD spaces: Implementation details

Mathis Birken

April 5, 2023

We use the same notation as in the other document.

The big picture

The Fréchet mean, also called Karcher mean, is about finding

$$g(x_1, ..., x_N) := \operatorname*{argmin}_{y \in \mathrm{SPD}_d} \sum_{i=1}^{N} \omega_i d(x_i, y),$$

where the ω_i are some weights. In the following discussion, we will ignore them because they can be introduced with only minor changes to the formulas.
The forward direction can be implemented using simple Riemannian gradient descent, which makes use of the computability of the Riemannian exponential map and its inverse. [1]
The backward direction can't be done with Pytorch autograd because we don't have a closed formula for the argmin. So the goal is to calculate the Jacobian of the Fréchet mean using the equation in terms of the second derivatives of the goal function f. In all of this, we use the coordinate versions of the matrix, that is, their vectorized upper half. Now recall that in order to make the Fréchet mean part of the autograd framework, we are actually interested in vector Jacobian products of the form wJ, where w is some vector in the output space. Thus the problem presents itself in the following form: given some vector (symmetric matrix) w, we need

$$w(Jg) = \left(-w \left(\sum_{n=1}^{N} \mathrm{Hess}^D (d^2_{x_n})_y \right)^{-1} \right) [B^T_{x_1,y}, ..., B^T_{x_N,y}].$$

(Small deviation: After evaluating the first product above, we have again a vector-Jacobian style product, because the matrix on the right is simply the Jacobian $J(x_1, ...x_N \mapsto \nabla_y f(x_1, ...x_N, y))$. That's why we could in theory use autodiff for this second half of the calculation – and that's what they do in the backward code for the Differentiating through the FM paper for the hyperbolic setting. Here, we will however also implement this second half explicitly using our theoretical results.)

[1] See `https://www.math.fsu.edu/~whuang2/pdf/KarcherMeanSPD_techrep.pdf` for more refined techniques.

In the following, we will use the symbols $(\partial_i)_{1\leq i\leq d(d+1)/2}$ for the basis matrices, the ones that correspond to unit vectors in the coordinate representations. (They contain either a one on the diagonal or two ones on opposite sides of the diagonal, and zeros everywhere else.)
Note that in principle, the Jacobian can be approximated (or tested) by using ε-pertubations in all the coordinate directions for all the input points. But as the forward direction is the real bottleneck here, that's clearly not feasible for use in practice.

The gradient

Let $X \in \mathrm{SPD}_d$ have orthogonal diagonalization $X = V\Lambda V^T$, where $\Lambda = \mathrm{diag}(\lambda_1, ..., \lambda_d)$. For the derivative of d_I^2 we notice the fact that we can write:

$$\mathrm{d}_X(d_I^2)(Z) = \sum_{i=1}^{d} 2\frac{\log\lambda_i}{\lambda_i}\langle Zv_i, v_i\rangle = 2\,\mathrm{tr}(V^T ZV \underbrace{\mathrm{diag}(..., \frac{\log\lambda_i}{\lambda_i}, ...)}_{=:\tilde{\Lambda}}),$$

because $\langle Zv_i, v_j\rangle = (V^T ZV)_{ij}$.
Now for d_Y^2 we use the chain rule. But because in general $Y^{-1}X$ won't be symmetric, writing $d_Y^2(X) = d_I^2(Y^{-1}X)$ doesn't make sense. Instead, we use conjugation: $d_Y^2(X) = d_I^2(\sqrt{Y^{-1}}X\sqrt{Y^{-1}})$. We realize that conjugation is linear in X and so the chain rule gives the very simple formula (this time V and $\tilde{\Lambda}$ are obtained from the conjugation of X.)

$$\begin{aligned}\mathrm{d}_X(d_Y^2)(Z) &= \mathrm{d}_{\sqrt{Y^{-1}}X\sqrt{Y^{-1}}}(d_I^2)(\sqrt{Y^{-1}}Z\sqrt{Y^{-1}})\\ &= 2\,\mathrm{tr}(V^T\sqrt{Y^{-1}}Z\sqrt{Y^{-1}}V\tilde{\Lambda})\\ &= 2\,\mathrm{tr}(Z\sqrt{Y^{-1}}V\tilde{\Lambda}V^T\sqrt{Y^{-1}}),\end{aligned}$$

where we used that cyclic permutation under the trace are permissible. Now how do we plug ∂_i in for Z? Notice that the trace of the product of two symmetric matrices is nothing else but the sum over their hadamard product. In this case, this sum is either exactly one diagonal element of the second factor or the sum of two opposite elements depending on the basis element ∂_i. (This is similar to how taking the regular inner product with a unit basis vector picks out a coordinate.) All in all, we get:

$$\nabla(d_Y^2)(X) = 2\delta(\sqrt{Y^{-1}}V\tilde{\Lambda}V^T\sqrt{Y^{-1}}),$$

where δ is the map doubling everything but the diagonal. The above equation should be thought of in vectorized form.

The Hessian

The formula given in the other document contains two small errors. After correcting them, we can write:

$$\begin{aligned}\mathrm{Hess}^D_X(d_I^2)(Z,W) &= 2\sum_{i=1}^{d}\frac{1-\log(\lambda_i)}{\lambda_i^2}\langle Zv_i,v_i\rangle\langle Wv_i,v_i\rangle \\ &\quad + 2\sum_{j=i+1}^{d}\left(\frac{\log\lambda_i}{\lambda_i}-\frac{\log\lambda_j}{\lambda_j}\right)\frac{1}{\lambda_i-\lambda_j}\langle Zv_i,v_j\rangle\langle Wv_i,v_j\rangle \\ &= 2\sum_{i,j=1}^{d}h(\lambda_i,\lambda_j)\langle Zv_i,v_j\rangle\langle Wv_i,v_j\rangle \\ &= 2\sum_{i,j=1}^{d}H_{ij}(V^TZV)_{ij}(V^TWV)_{ij} \qquad (H_{ij}:=h(\lambda_i,\lambda_j))\end{aligned}$$

where we define $h:\mathbb{R}^2_{>0}\to\mathbb{R}$ to be the continous function

$$h(x,y):=\begin{cases}\left(\frac{\log x}{x}-\frac{\log y}{y}\right)\frac{1}{x-y}, & x\neq y\\ \frac{1-\log x}{x^2}, & x=y.\end{cases}$$

Now the above sum can be interpreted as the sum over the Hadamard product of three matrices. Equivalently, if we represent those matrices by vectors in $\mathbb{R}^{d^2}$, it is simply the inner product of the Hadamard product $H\circ V^TZV$ and V^TWV.

Let's now turn to the more general case d_Y^2. By a similar argument as above involving the linearity of the conjugation in one argument we have:

$$\mathrm{Hess}_X(d_Y^2)(Z,W)=\mathrm{Hess}_{c(X)}(d_I^2)(c(Z),c(W)) \quad \text{where } c:A\mapsto\sqrt{Y^{-1}}A\sqrt{Y^{-1}}$$

In practice, we want to calculate the Hessian matrix in coordinates, so we plug in our matrix basis vectors $(\partial_i)_{1\leq i\leq d(d+1)/2}$ for Z and W. If we define the $(d(d+1)/2\times d^2)$-matrix M to have as rows the reshaped matrices $V^Tc(\partial_i)V$, we can perform the calculation above for all ∂_i,∂_j at once:

$$\mathrm{Hess}^D_X(d_Y^2)(\partial_i,\partial_j)=((H\circ M)M^T)_{ij}.$$

(The Hadamard product is meant to be applied in each row of M.)

The mixed second derivatives

Now we want to implement the $B_{x,y}$ matrices from the other document. In the notation of the paragraph in practice, in order to calculate the second partial derivatives we need to find a matrix Z such that $\bar{Z}_Y=\partial_j$, for each j. We simply set $Z:=\frac{1}{2}\partial_jY^{-1}$. Now the reasoning given appearently also works using the flat connection D. Using $(D_{\partial_i}\bar{Z})_X=Z\partial_i+\partial_iZ^T$ we obtain:

$$\begin{aligned}B_{X,Y}(\partial_i,\partial_j) &= -\,\mathrm{d}_X(d_Y^2)\left(\frac{1}{2}\partial_iY^{-1}\partial_j+\frac{1}{2}(\partial_iY^{-1}\partial_j)^T\right)\\ &\quad -\mathrm{Hess}^D_X(d_Y^2)\left(\partial_i,\frac{1}{2}\partial_jY^{-1}X+\frac{1}{2}(\partial_jY^{-1}X)^T\right)\end{aligned}$$

Recall that we are interested in calculating vector matrix products of the form $wB_{X,Y}$, with $w = \sum w_i \partial_i$. Coordinatewise:

$$\begin{aligned}
(wB_{X,Y})_j &= \sum_i w_i B_{X,Y}(\partial_i, \partial_j) \\
&= -\operatorname{d}_X(d_Y^2)\left(\frac{1}{2} w Y^{-1}\partial_j + \frac{1}{2}(wY^{-1}\partial_j)^T\right) \\
&\quad - \operatorname{Hess}_X^D(d_Y^2)\left(w, \frac{1}{2}\partial_j Y^{-1}X + \frac{1}{2}(\partial_j Y^{-1}X)^T\right) \\
&= -\frac{1}{2}\operatorname{d}_X(d_Y^2)(wY^{-1}\partial_j) + \frac{1}{2}\operatorname{d}_X(d_Y^2)(\partial_j Y^{-1} w) \\
&\quad - \sum_{i,k} H_{ik}(V^T\sqrt{Y^{-1}}w\sqrt{Y^{-1}}V)_{ik}(V^T\sqrt{Y^{-1}}(\partial_j Y^{-1}X + XY^{-1}\partial_j)\sqrt{Y^{-1}}V)_{ik} \\
&= -\operatorname{tr}(wY^{-1}\partial_j\sqrt{Y^{-1}}V\tilde{\Lambda}V^T\sqrt{Y^{-1}}) - \operatorname{tr}(\partial_j Y^{-1} w\sqrt{Y^{-1}}V\tilde{\Lambda}V^T\sqrt{Y^{-1}}) \\
&\quad - \operatorname{tr}\left((H \circ (V^T\sqrt{Y^{-1}}w\sqrt{Y^{-1}}V))^T V^T\sqrt{Y^{-1}}(\partial_j Y^{-1}X + XY^{-1}\partial_j)\sqrt{Y^{-1}}V\right) \\
&= -2\operatorname{tr}(\partial_j \operatorname{sym}(Y^{-1}w\sqrt{Y^{-1}}V\tilde{\Lambda}V^T\sqrt{Y^{-1}})) \\
&\quad - 2\operatorname{tr}\left(\partial_j \operatorname{sym}(Y^{-1}X\sqrt{Y^{-1}}V(H \circ (V^T\sqrt{Y^{-1}}w\sqrt{Y^{-1}}V))V^T\sqrt{Y^{-1}})\right)
\end{aligned}$$

(We set $\operatorname{sym}(M) := (M + M^T)/2$). Now using the same doubling-outside-the-diagional trick as above the above expression can be evaluated for all j at once. (Those calculations looks quite messy, I wonder if all of this can be written in a nicer way.)
Using similar techniques, it is also possible to compute the B terms in full matrix form.