Early bird gets the eigenworm

The idea and data for this lab come from Stephens *et al.*, "Dimensionality and Dynamics in the Behavior of *C. elegans*," *PLoS Comput Biol*, 2008. In this work the authors captured video of worms moving as they were subjected to stimuli (from "standard" to "painful"). Using image processing, they found 100 points representing a path along the back of single worm within each frame of the video and computed a representation independent of rotation and translation using the tangents to the path. The result is a $100 \times n$ matrix of tangent angles for n frames of video. They used $n = 56200$.

The authors then noted that dimension reduction by the SVD is extraordinarily successful for this data set; they showed that the data are well characterized by a small number of "eigenworms".[1] This makes some sense, as the motions are constrained a great deal by anatomy and kinematics.

Suppose $\boldsymbol{T}$ is the matrix of angles; i.e., column $\boldsymbol{t}_j$ is the vector of angles in the jth video frame. Suppose we have the full SVD $\boldsymbol{T} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$. This would make $\boldsymbol{V}$ an $n \times n$ matrix, which would not fit in memory, so we have to use a thin SVD. In the text we only did this for an $m \times n$ matrix with $m > n$, which is not the case for $\boldsymbol{T}$. However, $\boldsymbol{T}^T = \boldsymbol{V}\boldsymbol{S}^T\boldsymbol{U}$ does have a thin form in which $\boldsymbol{T}^T = \widehat{\boldsymbol{V}}\widehat{\boldsymbol{S}}^T\boldsymbol{U}$, where $\widehat{\boldsymbol{V}}$ is only $n \times 100$ and $\widehat{\boldsymbol{S}}$ is 100×100. Finally, this gives us

$$\boldsymbol{T} = \boldsymbol{U}\widehat{\boldsymbol{S}}\widehat{\boldsymbol{V}}^T, \tag{1}$$

the thin SVD we need.

Observe that

$$\boldsymbol{T} = \sum_{k=1}^{100} \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^T. \tag{2}$$

Because the singular values are always in decreasing order, we may approximate the original matrix by

$$\boldsymbol{T}_r = \sum_{k=1}^{r} \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^T, \tag{3}$$

for some rank $r \ll 100$. The range of this matrix is spanned by $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r$, which are the eigenworms. A good way to express the proportion of $\boldsymbol{T}$ that is captured by $\boldsymbol{T}_r$ is the ratio

$$\tau_r = \frac{\|\mathbf{s}_r\|_2^2}{\|\mathbf{s}_{100}\|_2^2}, \tag{4}$$

where $\mathbf{s}_k$ is the vector $[\sigma_1, \ldots, \sigma_k]$.

Goals

Given the data matrix, you will perform the SVD analysis, find a reasonable value for the cutoff rank r using the coefficients τ_k, and extract the eigenworms.

Preparation

Read section 7.5.

[1] One often sees the "eigen-" prefix used in this context, but the SVD is a more fundamental description of the mathematics.

Procedure

1. Load the `shapes.mat` file from the assignment site. It has the matrix `theta`, which is what we call $\boldsymbol{T}$ above.
2. On one graph, plot the first three columns of $\boldsymbol{T}$. (These are tangent angles of the worm's body as a function of arc length.)
3. Compute the three matrices in the reduced SVD (1) (see item 2 in the Preparation section). **You must use a 0 (zero) as a second argument to `svd`, or you will run out of memory.**
4. Let `s` be the vector of singular values. Using it, plot $1-\tau_r$ versus r on a semi-log scale, for $r = 1,\dots,100$. From this plot it should be clear that $r=4$ is a compelling choice. Compute and print out the value of c_4.
5. In a 2-by-2 subplot grid, plot the first 4 eigenworms.

Discussion

Given any single worm tracing, you can find its least-squares projection onto the range of the four eigenworms $\boldsymbol{u}_1,\dots,\boldsymbol{u}_4$ using a single backslash operation. Specifically we have

$$\boldsymbol{t} \approx c_1\boldsymbol{u}_1 + \cdots + c_4\boldsymbol{u}_4 = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \boldsymbol{u}_3 & \boldsymbol{u}_4 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}. \tag{5}$$

Choose three columns of $\boldsymbol{T}$ and plot each one together with its least-squares approximation. You should see that the noisiness in the data is eliminated.

The values $c_1,\dots,c_4$ represent a low-dimensional projection of a single worm tracing. These coordinates might be used to train an algorithm to classify or simulate different types of motion, as was done in the source reference.

Extra

The data has been presented as $\theta(s)$, i.e. tangent angle as a function of arclength. The tangent and normal vectors to the curve are given by

$$\begin{aligned} \boldsymbol{t}'(s) &= \frac{d\theta}{ds}\mathbf{n}(s), \\ \mathbf{n}'(s) &= -\frac{d\theta}{ds}\mathbf{t}(s), \\ \boldsymbol{t}(0) &= \begin{bmatrix} 1,0 \end{bmatrix}, \quad \mathbf{n}(0) = \begin{bmatrix} 0,1 \end{bmatrix}. \end{aligned} \tag{6}$$

The position vector $\boldsymbol{r}(s) = \begin{bmatrix} x(s), y(s) \end{bmatrix}$ is then given by $\boldsymbol{r}'(s) = \boldsymbol{t}(s)$, $\boldsymbol{r}(0) = \begin{bmatrix} 0,0 \end{bmatrix}$. For each of the four eigenworms plotted in step 5, solve these equations and plot $\boldsymbol{r}(s)$. (Hint: You may either discretize the ODEs using Euler's method and then plug in the computed θ, or you may interpolate the computed θ and plug that into an IVP solver. High accuracy is not a concern here.)