Scientific Computing Language
Homework 3

Problem 1 (Least squares, 20 pts)**.** Define the following data in Matlab:

```
t=(0:.5:10)';
y = tanh(t);
```

1. Fit the data to a cubic polynomial (using least squares) and plot the data together with the polynomial fit.

2. Fit the data to the function $c_1 + c_2 z + c_3 z^2 + c_4 z^3$, where $z = t^2/(1+t^2)$. Plot the data together with the fit. What feature of z makes this fit much better than the original cubic?

Problem 2 (Householder, 10 pts)**.** Let P be a Householder reflection matrix.

1. Find a vector u such that $Pu = -u$.

2. What algebraic condition is necessary and sufficient for a vector w to satisfy $Pw = w$? In n dimensions, how many such linear independent vectors are there?

Problem 3 (QR, 10 pts)**.** Let $A = QR$ be the thin QR factorization where $Q \in \mathbb{R}^{m\times n}$ and $R \in \mathbb{R}^{n\times n}$. The matrix $P = QQ^T$ has some interesting and important properties:

1. Show that $P = AA^+$.

2. Prove that $P^2 = P$. Moreover, any vector x can be written as $x = u + v$ where $u = Px$ and $v = (I-P)x$. Prove that u and v are orthogonal.

Problem 4 (Condition number, 15 pts)**.** Suppose $A \in \mathbb{R}^{m\times n}$ has full column rank with $m > n$. Condition number is defined similar to that of square matrix by replacing the inverse with pseudo inverse:

$$\kappa_2(A) = \|A\|_2 \cdot \|A^+\|_2.$$

Show that $\kappa_2(A) = \frac{\sigma_1}{\sigma_n}$ where σ_1 and σ_n are the largest and smallest singular values of A respectively. Use the definition to show that $\kappa_2(A) = \kappa_2(R)$. Moreover, show that $\kappa_2(A^TA) = \kappa_2(A)^2$.

Problem 5 (Power method, 15 pts)**.** Suppose $A \in \mathbb{R}^{m\times n}$ is nonsingular and that $Q \in \mathbb{R}^{n\times p}$ has orthonormal columns. The following iteration is referred to as *inverse orthogonal iteration.*

for $k = 1, 2, \ldots$ **do**
  Solve $AZ_k = Q_{k-1}$ for $Z_k \in \mathbb{R}^{n\times p}$;
  Apply QR factorization: $Z_k = Q_k R_k$;
end

Explain why this iteration can usually be used to compute the p smallest eigenvalues of A in absolute value. Note that to implement this iteration it is necessary to be able to solve linear systems that involve A. If $p = 1$, the method is referred to as the *inverse power method.*

Problem 6 (SVD, 30pts)**.** 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 $\boldsymbol{u}_1, \dots, \boldsymbol{u}_r$, 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 $\left[\sigma_1, \dots, \sigma_k\right]$.

One use of the eigenworms is to create a compact representation of the data. A column $\boldsymbol{t}_j$ of the original data can be expressed in terms of its closest approximation as a linear combination of the eigenworms:

$$\boldsymbol{t}_j \approx c_1\boldsymbol{u}_1 + \cdots + c_r\boldsymbol{u}_r = \boldsymbol{U}_r\boldsymbol{c},$$

where $\boldsymbol{U}_r$ is $100 \times r$. This is a least squares problem, but by orthogonality its solution is $\boldsymbol{c} = \boldsymbol{U}_r^T\boldsymbol{t}_j$. The r values in this vector give the components of $\boldsymbol{t}_j$ in the principal eigenworm directions and could be used as a low-dimensional representation for further analysis.

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.

[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 `T`.
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. Using `svd`, compute the three matrices in the thin SVD (1).
4. Let `s` be the vector of singular values. Using it, plot $1 - \tau_r$ versus r on a semi-log scale, for $r = 1, \ldots, 100$. From this plot it should be clear that $r = 4$ is a compelling choice. Compute and print out the value of τ_4.
5. In a 2-by-2 subplot grid, plot the first 4 eigenworms.
6. For the first three columns of $\boldsymbol{\{}T\boldsymbol{\}}$, plot the best approximation of each column by the leading 4 eigenworms. (The results will be much smoother curves than in step 1.)