Analyzing the Mariana Trench

Due in Canvas April 8, 2020 at 11:59 PM

1 Introduction

1.1 The Mariana Trench

The Mariana Trench is the deepest trench in the world, lying in the Pacific Ocean between Japan and Papua New Guinea. Because it is so deep, and the sides so steep, it is a unique environment on earth and the subject of much scientific inquiry.

The United States National Oceanic and Atmospheric Administration (NOAA) has conducted a series of bathymetric investigations of the Mariana Trench, collecting high-resolution data of the region. This data is often difficult to work with, containing an extremely high number of data points. You've been asked to reduce the size of the data in a way that maintains the overall structure, and perform some scientific analysis.

1.2 The Singular Value Decomposition

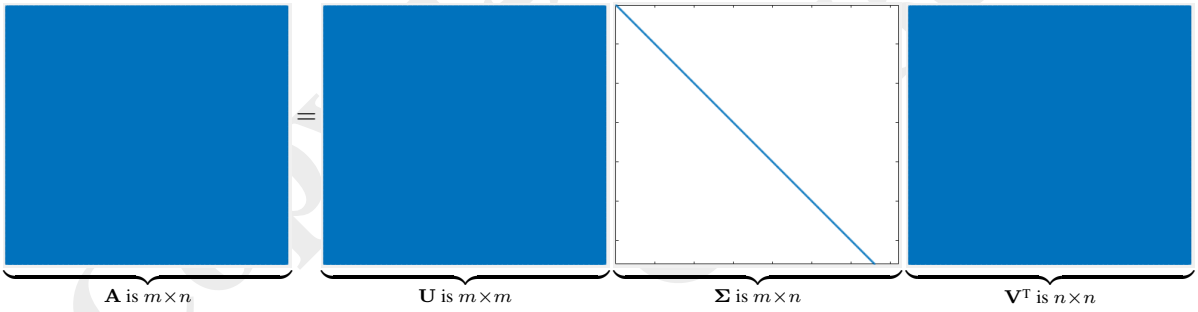
To perform this science, we will need to compute the Singular Value Decomposition (SVD) of the dataset. The SVD is a special factorization of a given matrix $\mathbf{A}$. Similarly to standard numbers, which can be factored several ways, for example,

$$12 = 3 \cdot 4 = 3 \cdot 2 \cdot 2 = 6 \cdot 4 \cdot \frac{1}{2},$$

a matrix can have many different factorizations, with many different purposes and benefits. The SVD factorization breaks a given matrix $\mathbf{A}$ into three parts,

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}},$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices (that is, $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}$, $\mathbf{U}\mathbf{U}^{\mathrm{T}} = \mathbf{I}$, $\mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}$ and $\mathbf{V}\mathbf{V}^{\mathrm{T}} = \mathbf{I}$) and $\mathbf{\Sigma}$ is a diagonal matrix.



Each of the three factor matrices tells you information about the matrix. $\mathbf{\Sigma}$ contains the "singular values," which are the square roots of the eigenvalues of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, and the columns of $\mathbf{V}$ are the associated eigenvectors. $\mathbf{U}$ holds the same information about the eigenvectors of $\mathbf{A}\mathbf{A}^{\mathrm{T}}$.

Physically speaking, this representation tells you how the matrix $\mathbf{A}$ acts on vectors. The columns of $\mathbf{V}$ represent a basis for $\mathbb{R}^n$, and the columns of $\mathbf{U}$ are a basis for $\mathbb{R}^m$. $\mathbf{A}$ is a matrix which maps the basis $\mathbf{V}$ to the basis $\mathbf{U}$, scaled by the singular values on the diagonal of $\mathbf{\Sigma}$. Written mathematically, that is

$$\mathbf{A}\vec{\mathbf{x}} = \mathbf{A}\left(c_1\vec{\mathbf{V}}_1 + c_2\vec{\mathbf{V}}_2 + \cdots + c_n\vec{\mathbf{V}}_n\right) = c_1\Sigma_{11}\vec{\mathbf{U}}_1 + c_2\Sigma_{22}\vec{\mathbf{U}}_2 + \cdots + c_m\Sigma_{mm}\vec{\mathbf{U}}_m.$$

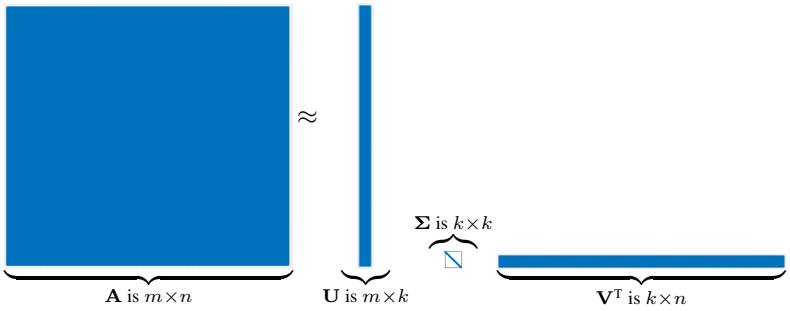
1.3 The Incomplete SVD

This representation is helpful for more than just a deeper understanding of the matrix $\mathbf{A}$, though. If we order the singular values and vectors so that $\Sigma_{11} \geq \Sigma_{22} \geq \ldots$, then we can make the approximation

$$c_1\Sigma_{11}\vec{\mathbf{U}}_1 + c_2\Sigma_{22}\vec{\mathbf{U}}_2 + \cdots + c_k\Sigma_{kk}\vec{\mathbf{U}}_k + \cdots + c_m\Sigma_{mm}\vec{\mathbf{U}}_m \approx c_1\Sigma_{11}\vec{\mathbf{U}}_1 + c_2\Sigma_{22}\vec{\mathbf{U}}_2 + \cdots + c_k\Sigma_{kk}\vec{\mathbf{U}}_k$$

for some $k < M$. That is, we're "cutting off" the action of the matrix $\mathbf{A}$ a little early, but we're confident that we've ordered the way the matrix $\mathbf{A}$ works so that we're losing as little information as possible.

This is called the "Incomplete Singular Value Decomposition." The core idea behind using the Incomplete SVD is that the columns of $\mathbf{U}$ and $\mathbf{V}$ corresponding to large diagonal elements of $\mathbf{\Sigma}$ essentially matter more, and we can save a lot of energy by using tall, skinny approximations to $\mathbf{U}$ and $\mathbf{V}$ which throw away the columns we don't care about.



1.4 Acquiring the data

NOAA hosts bathymetric datasets of the Mariana trench. A cleaned and lightly sampled version of that dataset has been posted to the Canvas page for this course. The original is $25\times$ the size, and is difficult to work with on personal computers. Download the three files

- `mariana_depth.csv`
- `mariana_latitude.csv`
- `mariana_longitude.csv`

from that location and put them in your working directory. CSV files are easy to import into most coding languages. For example, in MATLAB one would use the commands

```
A = importdata('mariana_depth.csv');
lon = importdata('mariana_longitude.csv');
lat = importdata('mariana_latitude.csv');
```

2 Questions

Be sure to type full sentences explaining the questions posed and your responses. Don't simply number your responses to individual questions.

2.1 Investigating the Trench

1. Import the data, and display it as an image in your report. Be sure to label your axes.
2. How deep is the deepest point sampled in the trench? What are its latitude and longitude?
3. The ocean floor can be nominally defined in this region at a depth of 6 kilometers. What is the average depth of the trench, that is, the mean depth among all points deeper than the ocean floor?

2.2 Computing Eigenvectors

1. Code your own algorithm to find the first eigenvector and eigenvalue of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, where $\mathbf{A}$ denotes the depth matrix. To do this,
   (a) First begin with a random guess vector of the correct size and magnitude one.
   (b) Apply $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ to that vector, and then divide the result by its magnitude to make sure it's still magnitude one. That is your updated guess.
   (c) Repeat the two steps above until the vector stops changing.

   Put another way, that is:

   (a) $\vec{\mathbf{u}}_1 =$ a random unit vector (magnitude 1).
   (b) $\vec{\mathbf{u}}_{n+1} := \mathbf{A}^{\mathrm{T}}\mathbf{A}\vec{\mathbf{u}}_n / \left\|\mathbf{A}^{\mathrm{T}}\mathbf{A}\vec{\mathbf{u}}_n\right\|$
   (c) Repeat until $\|\vec{\mathbf{u}}_{n+1} - \vec{\mathbf{u}}_n\|$ is small. It should only take around ten steps.

Note that the notation $\|\vec{\mathbf{u}}\|$ denotes the norm (magnitude) of the vector $\vec{\mathbf{u}}$ that has N components, and is defined as

$$\|\vec{\mathbf{u}}\| = \sqrt{\sum_{i=1}^{N} u_i^2}$$

Call the eigenvector you've found at the end $\vec{\mathbf{V}}_1$. Provide the eigenvalue and a plot of the eigenvector $\vec{\mathbf{V}}_1$ in your report. The $x-$axis of this plot should go from 1 to N, and the $y-$axis should be the corresponding value of the component of the eigenvector. Also provide a hypothesis for why this method of finding the eigenvector works. HINT: recall that the eigenvectors of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ constitute a basis of $\mathbb{R}^n$, and try expressing $\vec{\mathbf{u}}_n$ in terms of that basis.

2. We can also compute the k largest eigenvalues and associated eigenvalues of the system by using the fact that every eigenvector of a symmetric matrix (such as $\mathbf{A}^{\mathrm{T}}\mathbf{A}$) must be orthogonal to all the previous ones, with a process called *Gram-Schmidt Orthogonalization*. The basic idea is that we make sure that the current eigenvector we're building is orthogonal to every other eigenvector we've built so far. In other words, if we're currently working on constructing $\vec{\mathbf{V}}_i \approx \vec{\mathbf{u}}_n$,

$$\vec{\mathbf{u}}_n^{\mathrm{T}}\vec{\mathbf{V}}_1 = \vec{\mathbf{u}}_n^{\mathrm{T}}\vec{\mathbf{V}}_2 = \cdots = \vec{\mathbf{u}}_n^{\mathrm{T}}\vec{\mathbf{V}}_{i-1} = 0.$$

From a next-guess vector $\vec{\mathbf{u}}^*_{n+1} = \mathbf{A}^{\mathrm{T}}\mathbf{A}\vec{\mathbf{u}}_n$ that we haven't touched with this Gram-Schmidt method yet, we can "orthogonalize" it to the k previously computed eigenvectors $\vec{\mathbf{V}}_1$, $\vec{\mathbf{V}}_2$, ... $\vec{\mathbf{V}}_i$ by computing

$$\vec{\mathbf{u}}_{n+1} = \vec{\mathbf{u}}^*_{n+1} - \sum_{j=1}^{i} \left(\vec{\mathbf{u}}^{*\mathrm{T}}_{n+1}\vec{\mathbf{V}}_j\right)\vec{\mathbf{V}}_j.$$

These components we are subtracting from $\vec{\mathbf{u}}^*_{n+1}$ are called the "projection of $\vec{\mathbf{u}}^*_{n+1}$ onto $\vec{\mathbf{V}}_j$."

Using this idea, alter your code to compute the $k = 50$ largest eigenvalues and associated eigenvectors of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$. This orthogonalization step should be done *after* you multiply your previous guess vector by $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, but *before* you divide your new guess by its own magnitude.

(a) Initialize $\vec{\mathbf{V}}_1$ to $\vec{\mathbf{V}}_{50}$ as a matrix of zeros.

(b) For $i = 1$ to 50,

i. $\vec{\mathbf{u}}_1 =$ a random unit vector (magnitude 1)

ii. $\vec{\mathbf{u}}^*_{n+1} := \mathbf{A}^{\mathrm{T}}\mathbf{A}\vec{\mathbf{u}}_n$

iii. $\vec{\mathbf{u}}_{n+1} := \vec{\mathbf{u}}^*_{n+1} - \sum_{j=1}^{i-1}(\vec{\mathbf{u}}^{*\mathrm{T}}_{n+1}\vec{\mathbf{V}}_j)\vec{\mathbf{V}}_j$

iv. $\vec{\mathbf{u}}_{n+1} = \vec{\mathbf{u}}_{n+1}/\|\vec{\mathbf{u}}_{n+1}\|$

v. repeat ii-iv until $\|\vec{\mathbf{u}}_{n+1} - \vec{\mathbf{u}}_n\|$ is small (less than 10^{-2} or 10^{-3} should be fine for us), and store the final $\vec{\mathbf{u}}_{n+1}$ as $\vec{\mathbf{V}}_i$.

Store the eigenvectors you compute as the columns of a 1440×50 matrix and call it $\mathbf{V}$. Provide a semilog plot of the eigenvalues you find.

2.3 The Incomplete SVD Decomposition

1. Construct a 50×50 matrix with the square roots of the eigenvalues you found in the previous part on the diagonal. Call that matrix $\mathbf{\Sigma}$. Likewise, define a matrix $\mathbf{U}$ so that each column of $\mathbf{U}$ is equal to $\mathbf{A}$ times the associated column of V, divided by the associated element of $\mathbf{\Sigma}$.

2. The $\mathbf{V}$, $\mathbf{\Sigma}$ and $\mathbf{U}$ matrices you've constructed constitute the incomplete SVD decomposition of the matrix $\mathbf{A}$. The idea is that we've captured the relevant features of $\mathbf{A}$ without using so much storage. To test this, first calculate how many numbers we need to save in total between the $\mathbf{U}$, $\mathbf{\Sigma}$ and $\mathbf{V}$ matrices we've built, and compare that to the total number of elements of $\mathbf{A}$.

3. Now, provide a picture like you did of the matrix $\mathbf{A}$ in Question 1.1, but now of the matrix product $\mathbf{U\Sigma V}^{\mathrm{T}}$. Describe the relationship between these two images.

4. Try using fewer columns of $\mathbf{U}$ and $\mathbf{V}$ and a similarly smaller $\mathbf{\Sigma}$ and see how the image holds up. If your computer can handle it, try using more than 50. Describe how the number of singular values used affects the image.

3 Submission

Your group's report needs to be submitted to Canvas as a single pdf file. Any code that you include should be included within the single pdf as an appendix. Be sure to type up your final report.