COL 726 Homework 2

Due: Thursday, 31 January 2019

1. Suppose $\mathbf{A}$ is an $m \times n$ matrix with singular values $\sigma_1 \geq \cdots \geq \sigma_n$ and singular vectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$ and $\mathbf{v}_1, \ldots, \mathbf{v}_n$.

   (a) Let $S_k = \langle \mathbf{v}_1, \ldots, \mathbf{v}_k \rangle$ be the span of the first k right singular vectors. Prove that

   $$\inf_{\mathbf{x} \in S_k} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \sigma_k. \qquad [2 \text{ points}]$$

   (b) Prove that for <u>any</u> k-dimensional subspace $S \subseteq \mathbb{R}^n$, there exists a nonzero vector $\mathbf{x} \in S$ with $\|\mathbf{x}\|_2 = 1$ such that $\|\mathbf{A}\mathbf{x}\|_2 \leq \sigma_k$. [2 points]

2. Let S and T be two complementary subspaces in $\mathbb{R}^m$, and let $\{\mathbf{a}_1, \mathbf{a}_2, \ldots\}$ and $\{\mathbf{b}_1, \mathbf{b}_2, \ldots\}$ be bases for S and T respectively (but <u>not</u> necessarily orthonormal bases).

   (a) Prove that $\{\mathbf{a}_1, \mathbf{a}_2, \ldots\} \cup \{\mathbf{b}_1, \mathbf{b}_2, \ldots\}$ is a linearly independent set that spans $\mathbb{R}^m$, so it is a basis for the entire space. [2 points]

   (b) Find a formula for the matrix $\mathbf{P}$ that projects onto S along T. [1 point]

3. Suppose we define a modified inner product $f(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T\mathbf{C}\mathbf{v}$, where $\mathbf{C}$ is a diagonal matrix with positive diagonal entries. Then we can say that $\mathbf{u}$ and $\mathbf{v}$ are "C-orthogonal" if $f(\mathbf{u}, \mathbf{v}) = 0$.

   (a) Recall that the orthogonal projector in the direction of a nonzero vector $\mathbf{x}$ is the matrix $\frac{1}{\mathbf{x}^T\mathbf{x}}\mathbf{x}\mathbf{x}^T$. Find a formula for the <u>C-orthogonal</u> projector in the direction of $\mathbf{x}$, i.e. a matrix $\mathbf{P}$ such that, for any vector $\mathbf{v}$, $\mathbf{P}\mathbf{v}$ is a multiple of $\mathbf{x}$, and $\mathbf{v} - \mathbf{P}\mathbf{v}$ is $\mathbf{C}$-orthogonal to $\mathbf{x}$. [2 points]

   (b) Any $m \times n$ matrix $\mathbf{A}$ has a factorization of the form $\mathbf{A} = \hat{\mathbf{X}}\hat{\mathbf{R}}$, where $\hat{\mathbf{R}}$ is an $n \times n$ upper triangular matrix, and $\mathbf{X}$ is an $m \times n$ matrix whose columns are $\mathbf{C}$-orthonormal, i.e. $f(\mathbf{x}_i, \mathbf{x}_i) = 1$ and $f(\mathbf{x}_i, \mathbf{x}_j) = 0$ for all $i \neq j$. Give a method in the style of classical or modified Gram-Schmidt for computing such a factorization. (Even if you did not solve part (a), here you may assume you have solution for it.) [2 points]

4. Show that Householder reflectors work as advertised. That is, if $\mathbf{x}$ and $\mathbf{x}'$ are two vectors with $\|\mathbf{x}\|_2 = \|\mathbf{x}'\|_2$ and we define $\mathbf{F} = \mathbf{I} - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}$ where $\mathbf{v} = \mathbf{x} - \mathbf{x}'$, show algebraically that $\mathbf{F}^T\mathbf{F} = \mathbf{I}$ and $\mathbf{F}\mathbf{x} = \mathbf{x}'$. [2 points]

5. I have a set of n vectors $\mathbf{a}_1, \ldots, \mathbf{a}_n$ in $\mathbb{R}^m$ with $m > n$. I want to obtain n vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ in $\mathbb{R}^n$ such that the geometrical relationships between the vectors are exactly preserved, i.e. $\|\mathbf{b}_i\|_2 = \|\mathbf{a}_i\|_2$ and $\mathbf{b}_i^T\mathbf{b}_j = \mathbf{a}_i^T\mathbf{a}_j$ for all i, j. Describe a method to do so, and prove that it has the desired property. [3 points]

6. Low-rank approximations can be used for lossy data compression. Use this idea to compress an image by dividing it into blocks, and treating each one as a single row or column in a data matrix.

   (a) A colour image of width w and height h is essentially an $h \times w \times 3$ array of numbers. Construct a set of data points by dividing the image into $n \times n \times 3$ blocks and treating each block as an $3n^2$-dimensional vector (assume that w and h are multiples of n). Write a Python function `split(img, n)` that does this and returns a matrix **C**. Also write a function `join(C, n, w, h)` which undoes `split`, reconstructing the original image. [0.5 points]

   (b) Write a function `compress(C, r)` that returns an optimal rank-r approximation of a $p \times q$ matrix **C**, in the form of a $p \times r$ matrix **A** and an $r \times q$ matrix **B** such that $\mathbf{AB} \approx \mathbf{C}$. Then `join(A @ B, ...)` should reconstruct an approximation of the image from the compressed data. In your answer sheet, describe how `compress` works. [2 points]

   (c) Write a function `relError(img, img2)` which computes the relative error $\|$`img`$-$`img2`$\|/\|$`img`$\|$, where $\|\cdot\|$ denotes the sum of squares of all values. Separately, have `compress` return as a third output the relative error caused by compression, without actually computing it (or performing any subtractions at all!). In your answer sheet, explain how you do so. [1.5 points]

We will test your code as follows, so make sure this works:

```
# img is an array with shape (h,w,3), n and r are integers
import hw2
C = hw2.split(img, n)
A, B, e_rel = hw2.compress(C, r)
img2 = hw2.join(A @ B, n, w, h)
# e_rel should be nearly equal to hw2.relError(img, img2)
```

You are encouraged to use the functions

- `imread` and `imsave` from the `matplotlib.image` library to read and write images,
- `imshow` and `show` from `matplotlib.pyplot` to display them,
- `svd` from `scipy.linalg` to compute the singular value decomposition,

and any other NumPy or SciPy functions you find useful. Try your code on any of the test images from the Kodak image suite (`http://r0k.us/graphics/kodak/`) with $n = 16$, $r = 1, \ldots, 20$.