Paper Title

AMATH 383 Term Paper

Saransh Kacharia, Abishek Hariharan, Alex Chkodrov

Introduction

Hello.

Section 2

1. Define the orthogonal projection of $\mathbf{u} \in \mathbb{R}^n$ onto a linear space S as

$$\mathrm{proj}_S(\mathbf{u}) = \sum_{i=1}^{k} \frac{\mathbf{v}_i \cdot \mathbf{u}}{\|\mathbf{v}_i\|_2^2} \mathbf{v}_i,$$

where $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ forms an orthogonal (but not necessarily orthonormal) basis for S.

(a) Show $\mathbf{u} - \mathrm{proj}_S(\mathbf{u})$ is orthogonal to $\mathrm{proj}_S(\mathbf{u})$.

To show this we have to show that the dot product between $\mathbf{u} - \mathrm{proj}_S(\mathbf{u})$ and $\mathrm{proj}_S(\mathbf{u})$ is zero.

$$\mathbf{u} - \mathrm{proj}_S(\mathbf{u}) = \sum_{i=1}^{k} \frac{\mathbf{v}_i \cdot \mathbf{u}}{\|\mathbf{v}_i\|_2^2} \mathbf{v}_i$$

$$|\mathbf{u} - \sum_{i=1}^{k} \frac{\mathbf{v}_i \cdot \mathbf{u}}{\|\mathbf{v}_i\|_2^2} \mathbf{v}_i| \cdot |\sum_{i=1}^{k} \frac{\mathbf{v}_i \cdot \mathbf{u}}{\|\mathbf{v}_i\|_2^2} \mathbf{v}_i| =$$
$$\mathbf{u} \cdot \sum_{i=1}^{k} \frac{\mathbf{v}_i \cdot \mathbf{u}}{\|\mathbf{v}_i\|_2^2} \mathbf{v}_i - \sum_{i=1}^{k} \left(\frac{\mathbf{v}_i \cdot \mathbf{u}}{\|\mathbf{v}_i\|_2^2} \mathbf{v}_i \right) \cdot \left(\frac{\mathbf{v}_i \cdot \mathbf{u}}{\|\mathbf{v}_i\|_2^2} \mathbf{v}_i \right)$$

$\frac{\mathbf{v}_i \cdot \mathbf{u}}{\|\mathbf{v}_i\|_2^2} \mathbf{v}_i$ can be split into a unit vector and a scalar.
$\frac{\mathbf{v}_i \cdot \mathbf{u}}{\|\mathbf{v}_i\|_2} = \beta$, a scalar and $\frac{\mathbf{v}_i}{\|\mathbf{v}_i\|_2}$ is a unit vector.

$$\mathbf{u} \cdot \sum_{i=1}^{k} \frac{\mathbf{v}_i \cdot \mathbf{u}}{\|\mathbf{v}_i\|_2^2} \mathbf{v}_i - \sum_{i=1}^{k} \left(\frac{\mathbf{v}_i \cdot \mathbf{u}}{\|\mathbf{v}_i\|_2^2} \mathbf{v}_i \right) \cdot \left(\frac{\mathbf{v}_i \cdot \mathbf{u}}{\|\mathbf{v}_i\|_2^2} \mathbf{v}_i \right) =$$
$$\mathbf{u} \cdot \sum_{i=1}^{k} \beta \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|_2} - \sum_{i=1}^{k} \beta \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|_2} \cdot \beta \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|_2} =$$
$$\mathbf{u} \cdot \sum_{i=1}^{k} \beta \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|_2} - \beta^2$$

Because $\sum_{i=1}^{k} \beta \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|_2}$ is a projection of $\mathbf{u}$.

$$\mathbf{u} \cdot \sum_{i=1}^{k} \beta \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|_2} = \beta^2$$
$$\mathbf{u} \cdot \sum_{i=1}^{k} \beta \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|_2} - \beta^2 =$$
$$\beta^2 - \beta^2 = 0$$

(b) Show that $\|\mathbf{u}\|_2^2 = \|\mathrm{proj}_S(\mathbf{u})\|_2^2 + \|\mathbf{u} - \mathrm{proj}_S(\mathbf{u})\|_2^2$

2. In this exercise, we prove the Cauchy-Schwarz inequality, which states that

$$|\mathbf{u} \cdot \mathbf{v}| \leq \|\mathbf{u}\|_2 \|\mathbf{v}\|_2$$

for any vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$.

(a) Prove that $\|\mathrm{proj}_{\mathbf{v}}(\mathbf{u})\|_2 \leq \|\mathbf{u}\|_2$.

(b) Use part (a) to prove the Cauchy-Schwarz inequality.

(c) Show that $|\mathbf{u} \cdot \mathbf{v}| = \|\mathbf{u}\|_2 \|\mathbf{v}\|_2$ if and only if $\mathbf{u} = \alpha \mathbf{v}$ for some $\alpha \in \mathbb{R}$. That is, equality of the Cauchy-Schwarz inequality holds if and only if $\mathbf{u}$ is a scalar multiple of $\mathbf{v}$.

QR Factorization

3. Compute (by hand) the reduced QR factorization of the given matrix $\mathbf{A}$, and use it to solve (by hand) the linear system of equations $\mathbf{Ax} = \mathbf{b}$, where

$$\mathbf{A} = \begin{bmatrix} 1 & 6 & 14 \\ 1 & -2 & -8 \\ 1 & 6 & 2 \\ 1 & -2 & 4 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} 7 \\ -1 \\ 7 \\ -1 \end{bmatrix}.$$

4. Compute (by hand) the full QR factorization for the following matrix:

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

Note that we allow the diagonal of R to have zeros if $\mathbf{A}$ is not full rank.

5. Consider $\mathcal{P}_2$, the space of polynomials of degree at most 2. For any functions $f, g \in \mathcal{P}_2$ define their inner product as

$$\langle f, g \rangle = \int_{-1}^{1} f(t)g(t)\, dt.$$

(You need not show that this is an inner product.) Starting from the basis $\{1, x, x^2\}$, use the Gram-Schmidt process (Algorithm 2 on page 94) to build an orthonormal basis for $\mathcal{P}_2$.

The resulting polynomials are scalar multiples of what are known as the Legendre polynomials, P_j, which are conventionally normalized so that $P_j(1) = 1$. Computations with such polynomials form the basis of spectral methods, one of the most powerful techniques for the numerical solution of partial differential equations.

Matlab

6. In this exercise, we will learn how to create functions in matlab to study the stability of different algorithms for QR factorization. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ \epsilon & \epsilon & 0 & 0 & 0 \\ \epsilon & 0 & \epsilon & 0 & 0 \\ \epsilon & 0 & 0 & \epsilon & 0 \\ \epsilon & 0 & 0 & 0 & \epsilon \end{bmatrix} \quad \text{with} \quad \epsilon = 10^{-6}.$$

   (a) Use the command `cond` to find the condition number of $\mathbf{A}$ in the 1-norm. Would you say that this matrix is ill-conditioned? Save this result as `A1.dat`.

   (b) Watch the video tutorial on creating functions in matlab found at `https://www.youtube.com/watch?v=qo3AtBoyBdM`.

   (c) Create the function `classicalGS` (inside a new m-file named `classicalGS.m`) to compute the QR factorization using the classical Gram-Schmidt algorithm (Algorithm 1, page 90). This function should take as its input a matrix $\mathbf{A}$ and return as its output the matrices $\mathbf{Q}$ and $\mathbf{R}$. Use this function to compute the QR factorization of our given matrix $\mathbf{A}$, then compute the 1-norm of the error $\|\mathbf{Q}^T\mathbf{Q} - \mathbf{I}\|_1$ and save it as `A2.dat`. Below is some code to get you started:

   (d) Create the function `modifiedGS` to compute the QR factorization using the modified Gram-Schmidt algorithm (Algorithm 4, page 106). Use this function to compute the QR factorization of our given matrix $\mathbf{A}$, then compute the 1-norm of the error and save it as `A3.dat`.

   (e) Create the function `twostepGS` to compute the QR factorization using two steps of the Gram-Schmidt algorithm (Algorithm 5, page 106). Use this function to compute the QR factorization of our given matrix $\mathbf{A}$, then compute the 1-norm of the error and save it as `A4.dat`.

   (f) Compute the QR factorization of our given matrix $\mathbf{A}$ using the matlab command `qr`. Compute the 1-norm of the error and save it as `A5.dat`.

   (g) Which algorithms appeared to be the most stable for computing the QR factorization of $\mathbf{A}$? In the table below, fill in the error values and rank the algorithms from most stable (1) to least stable(4). Include this table in your written homework.

Algorithm	Error	Rank
Classical Gram-Schmidt		
Modified Gram-Schmidt		
Two Steps Gram-Schmidt		
Matlab QR		

For this problem, you must submit four matlab files to Scorelator: the functions `classicalGS.m`, `modifiedGS.m`, and `twostepGS.m`, and your main script which generates your data files. Before clicking submit, **you must highlight your main script** by clicking on it. If you highlight one of the function files, Scorelator will give you a 0.