Week 13 Discussion Solutions

CAS CS 132: Geometric Algorithms

November 27, 2023

During discussion sections, we will go over three problems.

- The first will be a warm-up question, to help you verify your understanding of the material.
- The second will be a solution to a problem on the assignment of the previous week. **But not this week, there was no assignment last week.**
- The third will be a problem similar to one on the assignment of the following week.

The remainder of the time will be dedicated to open Q&A.

1 Inner Products, Norms, Orthogonality

The first two problems come from *Linear Algebra and its Applications.*

A. Given vectors

$$\mathbf{u} = \begin{bmatrix} 2 \\ -5 \\ -1 \end{bmatrix} \qquad \mathbf{v} = \begin{bmatrix} -7 \\ -4 \\ 6 \end{bmatrix}$$

compute (by hand) $\mathbf{u} \cdot \mathbf{v}$, $\|\mathbf{u}\|^2$, $\|\mathbf{v}\|^2$, and $\|\mathbf{v} + \mathbf{u}\|^2$

B. Show that if $\mathbf{y}$ is orthogonal to $\mathbf{u}$ and $\mathbf{v}$, then it is orthogonal to $\mathbf{u} + \mathbf{v}$.

C. Find a linearly independent set of three nonzero vectors in $\mathbb{R}^4$, all of which are orthogonal to

$$\begin{bmatrix} 1 \\ 3 \\ -1 \\ 3 \end{bmatrix}$$

Solution.

A.

$$\begin{aligned}
\mathbf{u} \cdot \mathbf{v} &= 2(-7) + (-5)(-4) + (-1)(6) = -14 + 20 + -6 = 0 \\
\|\mathbf{u}\|^2 &= 2(2) + (-5)(-5) + (-1)(-1) = 4 + 25 + 1 = 30 \\
\|\mathbf{v}\|^2 &= (-7)(-7) + (-4)(-4) + 6(6) = 49 + 16 + 36 = 101 \\
\|\mathbf{u} + \mathbf{v}\|^2 &= (2 + (-7))^2 + ((-5) + (-4))^2 + (-1 + 6)^2 = 25 + 81 + 25 = 131
\end{aligned}$$

It may be worthwhile to think a bit about what is going on geometrically here.

B. We need to verify that $\mathbf{y}^T(\mathbf{u} + \mathbf{v}) = 0$. By distributing the multiplication over the sum, we see that this is the same as $\mathbf{y}^T\mathbf{u} + \mathbf{y}^T\mathbf{v}$. Each summand is equal to 0 by the assumption that $\mathbf{y}$ is orthogonal to $\mathbf{v}$ and $\mathbf{u}$ separately.

C. a vector $[v_1 \;\; v_2 \;\; v_3 \;\; v_4]^T$ is orthogonal to the given vector if

$$v_1 + 3v_2 - v_3 + 3v_4 = 0$$

We can think of this as a system of linear equations with a single nontrivial equation whose augmented matrix has the reduced echelon form

$$\begin{bmatrix} 1 & 3 & -1 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

In essence, this question asks to find a basis for the null space of this matrix, and we can use any of the techniques we've learned so far to do this. Writing the general form solution as a linear combination of vectors with free variables as weights gives us the vectors

$$\begin{bmatrix} -3 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} -3 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

It is straightforward to verify that all of these vectors are orthogonal to the given vector.

2 Damping without a Matrix

Suppose that A is a $n \times n$ stochastic matrix. According to what we talked about in lecture, the matrix we use to perform PageRank is

$$(1-\alpha)A + \frac{\alpha \mathbf{1}^{n\times n}}{n}$$

where $\mathbf{1}^{n\times n}$ here is the $n \times n$ all-ones matrix. However, for the assignment this week, you will not be able to build this matrix because it is too dense. Given a vector $\mathbf{v}$ from $\mathbb{R}^n$, write down an expression for

$$\left((1-\alpha)A + \frac{\alpha \mathbf{1}^{n\times n}}{n}\right)\mathbf{v}$$

which does not require building the matrix $\mathbf{1}^{n\times n}$. *Hint.* The expression should be of the form

$$(1-\alpha)A\mathbf{v} + \mathbf{u}$$

where $\mathbf{u}$ is a vector depending on $\mathbf{v}$ and the all-ones *vector* $\mathbf{1}^n \in \mathbb{R}^n$.

Solution. Clearly the vector $\mathbf{u}$ has to be the same as $\frac{\alpha \mathbf{1}^{n\times n}\mathbf{v}}{n}$. The trick is to recognize that this vector is simple in that all its entries are identical: every entry is $\mathbf{1}^n \cdot \mathbf{v}$. Therefore

$$\mathbf{u} = \frac{\alpha}{n}(\mathbf{1}^n \cdot \mathbf{v})\mathbf{1}^n$$

which does not require building the $n \times n$ all-ones matrix.

When you go to implement this with NumPy, its worth noting you can simplify it further. Think about how you might use `np.sum` instead of `np.dot`, for example.