IE 512 Graphs, Networks, and Algorithms

Homework 4 Solution

Problem 4.1

(a) In the best case, where the range of B and the null space of A has no intersection, then $\text{rank(AB)} = \min(\text{rank(A)}, \text{rank(B)}) = 3$. In the worst case, where the range of B has the largest overlap with the null space of A, then $\text{rank(AB)} = \min(\text{rank(A)}, \text{rank(B)}) - \dim(\text{null(A)}) = 2$. Here we used the fact that $\dim(\text{null(A)}) = 5 - \text{rank(A)}$.

We have $\text{rank(AB)} = 3$ for

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \text{ and } B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

We have $\text{rank(AB)} = 2$ for

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \text{ and } B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

(b) We only need to verify additivity and scaling. For any $u, v \in V^\perp$ and $\lambda_1, \lambda_2 \in \mathbb{R}$, it is clear that $\lambda_1 u + \lambda_2 v \in V^\perp$. This follows form the fact that if $u^T x = 0$ and $v^T x = 0$ for all $x \in V$, then $(\lambda_1 u + \lambda_2 v)^T x = \lambda_1 u^T x + \lambda_2 v^T x = 0$.

(c) $P = UU^T$ for some orthonormal matrix U. Then, it follows that $P^2 = UU^T UU^T = UU^T = P$ and $P^T = (UU^T)^T = UU^T = P$.

(d) Consider the projection onto V and $V^\perp$ defined as the projection matrices $P = UU^T$ and $P^\perp = U^\perp (U^\perp)^T$. Here U and $U^\perp$ are orthonormal basis for V and $V^\perp$ respectively. Note that $[UU^\perp][UU^\perp]^T = \mathbb{I}$. Then,

$$x = [UU^\perp][UU^\perp]^T x = \underbrace{UU^T x}_{=v} + \underbrace{U^\perp (U^\perp)^T x}_{=v^\perp} .$$

We are left show that $v \in V$ and $v^\perp \in V^\perp$. This immediately follows from the fact that U span V and $U^\perp$ span $V^\perp$.

(e) Consider an orthonormal basis U of subspace V. Then, we know that $\text{rank}(U) + \dim(\text{null}(U)) = n$. The claim follows from the fact that $\text{rank}(U) = \dim(V)$ and $\dim(\text{null}(U)) = \dim(V^\perp)$.

Problem 4.4

Since $\lambda_1 = \max_x \frac{x^T A x}{x^T x}$, we have

$$\begin{aligned} d_{\text{ave}} &= \frac{\mathbb{1}^T A \mathbb{1}}{\mathbb{1}^T \mathbb{1}} \\ &\leq \lambda_1 \, . \end{aligned}$$

Let v be the eigenvector corresponding to the largest eigenvalue and i be the one with the maximum value in v such that $v_i \geq v_j$ for all $j \in [n]$. Then,

$$\begin{aligned} \lambda_1 &= \frac{(Av)_i}{v_i} \\ &= \frac{\sum_j A_{ij} v_j}{v_i} \\ &= \sum_{j \in N(i)} \frac{v_j}{v_i} \\ &\leq d_i \\ &\leq d_{\max} \, . \end{aligned}$$

Problem 4.5 We first prove the if part: if exists such a two partition then the graph is balanced. Notice that is such a partition exists, then any cycle in the graph must have even number of crossings between partitions: if it starts in A a cycle must end in A. This implies that there must be even number of negative edges in any cycle. This proves the if part.

Next, we prove that if the graph is balanced, then there exists such a two partition as described in the problem. We construct the partition as follows: choose any node i and let $A = \{i\}$ and $B = \emptyset$. Then we add all the positive edge neighbors of i in A and negative edge neighbors in B. We continue increasing the sets until all the nodes in the same connected component as i are included. We repeat for each connected component. The only time this algorithm creators conflict is when (a) a node already in A is connected to a node in B by a positive edge; or (b) a node already in A is connected to a node in A by a negative edge.

First, (a) never happens for a balanced graph, since the presence of a positive edge between a node a in A and node b in B implies that there is a cycle $i - \cdots - a - b - \cdots - i$ such that there are odd number of negative edges. This contradicts our assumption that the graph is balanced. Similarly, for (b) this never happens for a balanced graph. Hence, the above algorithm always finds a two partition for a balanced graph.