DM561 — Linear Algebra with Applications

Sheet 5, Fall 2020

Author: Robert Rasmussen

Exericse 1

As shown in the slides, pre-multiplying a permutation matrix P by another matrix A results in permuting the rows of A. Since P_2 also is a permutation matrix each row and column contains a single 1 and the remaining is 0s. Therefore, P_1P_2 has a single 1 in each row and the remaining is zeros. Permuting the rows doesn't change that no two rows share a 1 in the same column position. Thus, P_1P_2 is also a permutation matrix.

A bit more formal

Let P_1 and P_2 be permutation matrices for which P_1P_2 is defined. P_1P_2 is defined if P_1 is a $m \times n$ matrix and P_2 is a $n \times r$ matrix. However, permutation matrices are square matrices so both P_1 and P_2 are $n \times n$ matrices.

Observe that the rows of P_1 are the standard (orthonormal) basis for $\mathbb{R}^n$ and the columns of P_2 are too the standard (orthonormal) basis for $\mathbb{R}^n$. P_1 consists of row vectors r_i and P_2 of column vectors c_i for $i = 1, 2, \ldots, n$.

$$P_1 = \begin{bmatrix} \mathbf{r_1} \\ \mathbf{r_2} \\ \vdots \\ \mathbf{r_n} \end{bmatrix} \qquad P_2 = \begin{bmatrix} \mathbf{c_1} & \mathbf{c_2} & \ldots & \mathbf{c_n} \end{bmatrix}$$

The product P_1P_2 is computed as

$$P_1P_2 = \begin{bmatrix} \mathbf{r_1} \cdot \mathbf{c_1} & \mathbf{r_1} \cdot \mathbf{c_2} & \ldots & \mathbf{r_1} \cdot \mathbf{c_n} \\ \mathbf{r_2} \cdot \mathbf{c_1} & \mathbf{r_2} \cdot \mathbf{c_2} & \ldots & \mathbf{r_2} \cdot \mathbf{c_n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{r_n} \cdot \mathbf{c_1} & \mathbf{r_n} \cdot \mathbf{c_2} & \ldots & \mathbf{r_n} \cdot \mathbf{c_n} \end{bmatrix}$$

Observe the ith row consists of n scalars arising from multiplying r_i by c_j for $j = 1, 2, \ldots, n$. Since r_i is orthogonal to all but one of c_j (follows from the row vectors of P_1 and the column vectors of P_2 being the same orthonormal basis for $\mathbb{R}^n$), the scalars are all 0 except the one which arises from $r_i \cdot c_j$ where $r_i = c_j$ which evaluates to 1. The same argument can be made for the columns. Therefore, P_1P_2 is also a permutation matrix.

Exericse 2

(1) The graphs can be drawn like this:

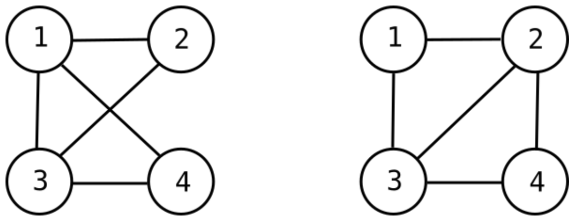


Figure 1: Left graph is G_A and right graph is G_B.

(2) Yes. The following structure preserving bijection $f : G_A.V \to G_B.V$ exists:

$$\begin{aligned} 1 &\to 2 \\ 2 &\to 1 \\ 3 &\to 3 \\ 4 &\to 4 \end{aligned}$$

(3) There are 6 different representations (adjacency matrices) for the mathematical object they represent. Why?

G_A has 4 vertices so the permutation matrices are of size 4×4. There are $4! = 24$ different permutation matrices of size $4! = 24$. However, some permutation matrices result in the same adjacency matrix when applied to G_A. So how many unique adjacency matrices are there?

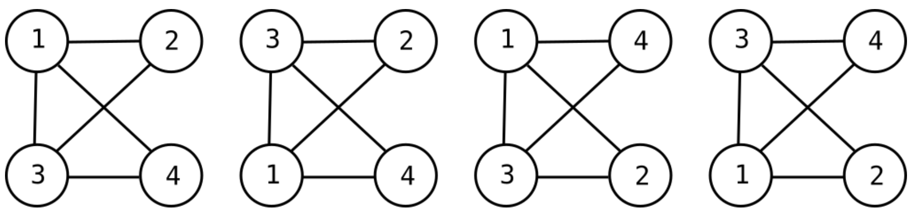
Observe that we can perform the following permutations without changing the adjacency matrices.

- Keep 1, 2, 3, and 4 as they are (identity).
- Switch 1 and 3 while keeping 2 and 4.
- Switch 2 and 4 while keeping 1 and 3.
- Switch 1 and 3 while also switching 2 and 4.

Ie. the graph of the adjacency matrix

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

can we drawn in the following ways



The same can be done with the other adjacency matrices (like the one for G_B). Thus, there are $\frac{4!}{4} = \frac{24}{4} = 6$ different adjacency matrices.

(4) Same as in (3) due to isomorphism between G_A and G_B cf. (1).

(5) Yes. Since they are isomorphic there must exist a permutation matrix P s.t. $A = P(PB)^T$. (see theorem in [2, p. 24]).

(6) We could use what [2] calls *Brute Force Graph Isomorphism Check*, ie. er construct all possible permutation matrices (here of size 4×4) and include as our answer, the permutation matrices s.t. $A = P(PB)^T$.

Lets be smart and use what [2] calls *Improved Graph Isomorphism Check*. The procedure is similar to the brute force approach from before, however, we perform some initial pruning before generating all permutation matrices. We make the following observations w.r.t. which vertices it makes sense a vertex in G_B is mapped to:

Vertex 1: To vertex 2 or 4 in G_A.

Vertex 2: To vertex 1 or 3 in G_A.

Vertex 3: To vertex 1 or 3 in G_A.

Vertex 4: To vertex 2 or 4 in G_A.

Consider Vertex 1 in G_B. Its degree is 2, so mapping to Vertex 1 or 3 in G_A doesn't make sense since their degree is 3.

We can now generate the permutation matrices from

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

which we create from the observations above (e.g. first column was 1 in position 2 and 4 to allow for permutation matrices which maps Vertex 1 of G_B to either Vertex 2 or 4 of G_A). Following the procedure from [2, p. 27] we obtain the permutation matrices

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

All of these permutation matrices satisfy $A = P(PB)^T$.

Exericse 3

(1) An adjacency matrix is

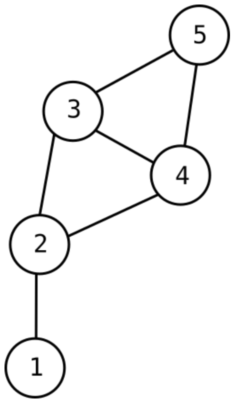
$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

How many are there? Like *The One Ring*...there is only one! Why? No matter how we re-label the vertices each vertex is still connected to the two other vertices.

(2) There are $3! = 6$ permutation matrices s.t. $A = P(PA)^T$. As noted in (1), no matter how you re-label (permute the graph) the vertices of the graph will end up being connected to the two other vertices. Therefore, all $3! = 6$ permutation matrices satisfy $A = P(PA)^T$.

Exericse 4

(1) Lets label the vertices like this



An adjacency matrix is

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

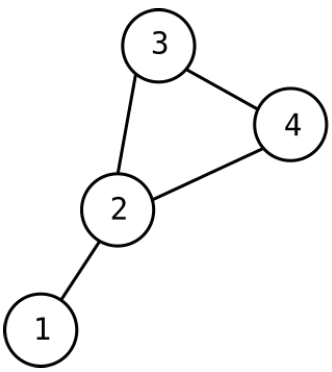
(2) Lets make the following obvervations:

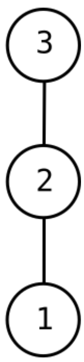
- The vertices 3 and 4 can be switched and they will still be adjacent to the same vertices, ie. it wouldn't change the adjacency matrix.
- Not changing anything (just keeping the graph labels as is) will not change the adjacency matrix.
- Otherwise, nothing can be done which wouldn't change the adjacency matrix. The reason is different degrees and neighbors among 1, 2, and 5. (the degree of 1 is 1, the degree of 5 is 2, and the degree of 2 is 3).

Thus, there are 2 different permutation matrices s.t. $A = P(PA)^T$ for a fixed A.

Exericse 5

(1) Lets label the vertices like this





Adjacency matrices for the G_A and G_B are

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

(2) Yes! See (3) and (4).

(3) The brute-force way would be to take A and B from (1) and create M^0 as

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

as described in [1, p. 8] and from it create the permutation matrices M^d systematically. Thereafter, we simply compute $C = M(MA)^T$ using the created permutation matrices and check $B_{ij} = 1 \Rightarrow C_{ij} = 1 \forall i, j$ - the result is the number of permutaiton matrices which satisfy the above steps. (we want to find G_B as subgraph of G_A).

Another way: Make the following observations

- By removing vertex 4 from G_A, we can find G_B as a subgraph in 2 ways.
- By removing vertex 3 from G_A, we can find G_B as a subgraph in 2 ways.
- By removing vertex 1 from G_A, we can find G_B as a subgraph in 6 ways.

Thus, in total G_B can be found as a subgraph of G_A in 10 ways. This is illustrated in Figure 2 and Figure 3.

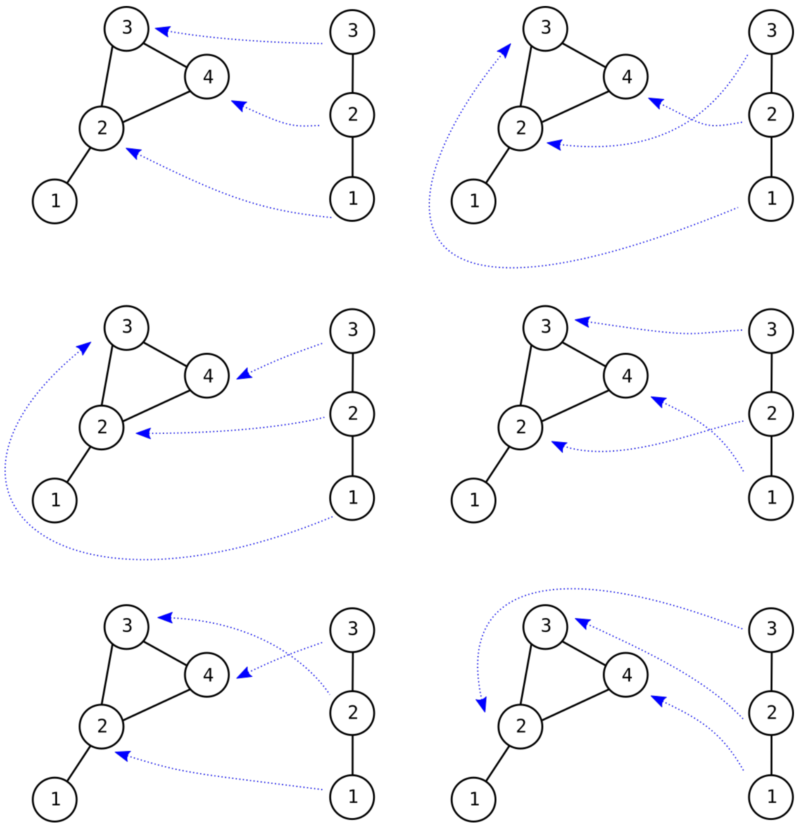


Figure 2: Ways to find G_B as a subgraph G_A (excluding the ways to find G_B as an induced subgraph of G_A)

(4) There are 4, which are the ways we found G_B as a subgraph of G_A when removing vertices 3 and 4. (see figure below) The other six ways to find G_B in G_A doesn't satisfy the *induced subgraphs* requirements. This is illustrated in Figure 3.

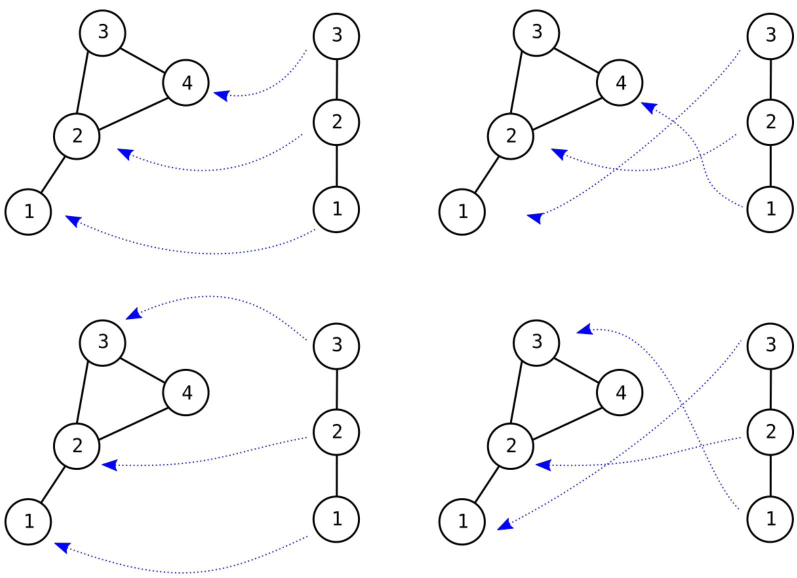
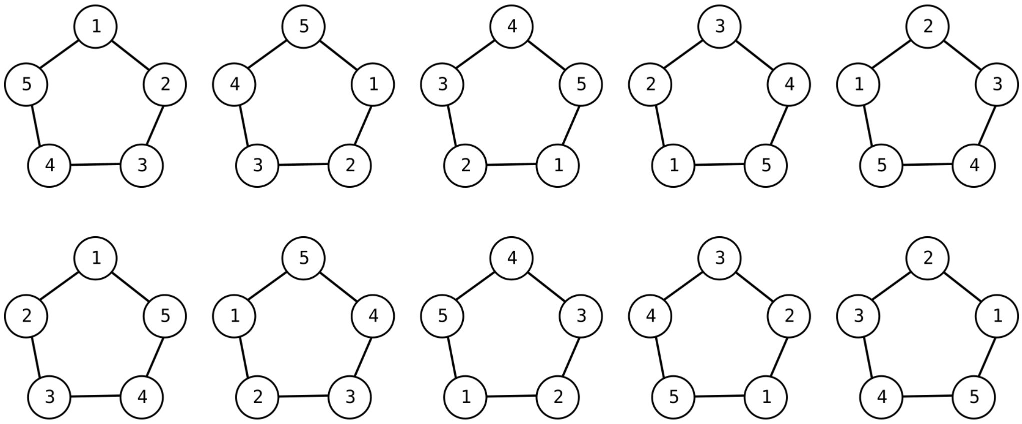


Figure 3: Ways to find G_B as an induced subgraph of G_A

Exericse 6

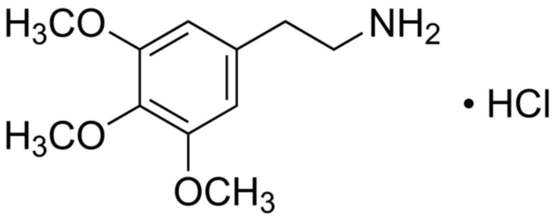
The `numIsomorphisms` function returns the number of permutation P matrices which satisfy $A = P(PB)^T$ (ie. number of different isomorphisms). In this case, $A = A$ so we are trying to find the number of different automorphisms (isomorphism between G_A and G_A). We can easily see that the following rotations and mirrorings (permutations) don't change the adjacency matrix:



Since we can find 10 such permutations, the result of the call `numIsomorphisms(A, A)` is as expected.

Exericse 7

I could find 5 structures with the specified substructure. For those I tried I could find a price. For example, consider the following structure



named "Mescaline hydrochloride solution" which would cost me 263 DKK for 1 mL.

Why this exercise? Finding structures with the given structure as a substructure can be solved by viewing the structures as graphs and then checking which structures/graphs the given structure/graph is a subgraph of.

References

[1] Daniel Merkle. Subgraph isomorphism. URL `https://dm561.github.io/assets/ullmann.pdf`, 2020.

[2] Daniel Merkle. The Graph-(and the Subgraph)-Isomorphism Problem, the Ullmann Algorithm,and Applications in Chemistry. URL `https://dm561.github.io/assets/DM561-DM562-Graphs-small.pdf`, 2020.