CHAPTER 2

Graph Matching using Spectral Properties and Average Shortest Distance

One of the most fundamental problems in graph theory is the problem of graph isomorphism also referred to as exact graph matching problem in literature. The problem of determining whether a pair of graphs is isomorphic has found many applications in various domains and has been extensively studied. The question, whether graphs are isomorphic plays an important role in many applications of graph theory. For example, chemists use molecular graphs to model chemical compounds and isomorphism between molecular graphs indicates chemical isomers. Vertices of the molecular graphs represent atoms and edges represent chemical bonds. When a new compound is synthesized, a database of molecular graphs is checked to determine whether the graph representing the new compound is isomorphic to the graph of any of the existing compounds and this will result in identifying new isomers. Electronic circuits are modeled as graphs in which the vertices represent components and the edges represent connections between them. Graph isomorphism is the basis for the verification that a particular layout of a circuit corresponds to the design’s original schematics. It can also be used in determining whether a chip from one vendor includes the intellectual property of another vendor; these are a few of the very large number of applications where graph matching is employed. In the section 2.1 a brief introduction to graph matching and a new methodology for ascertaining graph matching using spectral properties of the adjacency matrix and average shortest distance to other vertices from a vertex is provided.

• ¹ Parts of this chapter appear in the paper “Eigen Vector and Shortest Distance Based Approach to find Graph Isomorphism” published in International Journal of Combinatorial Graph Theory and Applications (IJCGTA), Vol.4, No. 1, (2011), pp. 67-76.
2.1 Introduction

Finding whether two graphs are isomorphic is a very important problem in graph theory because of its applications. Two graphs can been described as isomorphic or similar if their properties are identical. The properties of graphs are preserved by graph isomorphism. Properties that do not describe the structure of the graph, but perhaps specify the representation, need not be preserved under isomorphism. A common example of this is: if the nodes of the graphs are represented as numbers from 1 to n, graph isomorphism does not require the quantity $\sum_{i=1}^{n} i \cdot \deg(v_i)$ to be preserved.

To show that two graphs are not isomorphic, the following conditions are individually sufficient.

1. The two graphs have different numbers of vertices.
2. The two graphs have different numbers of edges.
3. One graph has parallel edges and the other does not.
4. One graph has a self loop and the other does not.
5. One graph has vertices of degree k (for example) and the other does not.
6. One graph is connected and the other is not.
7. One graph has a cycle and the other does not, etc.

Though showing, two graphs are not isomorphic is easy; finding whether two connected graphs are isomorphic/similar has been known to be a difficult problem. A few necessary conditions/ invariance’s do exist for two graphs to be isomorphic; some of them are the degree invariance, distance multiplicity invariance, sub-graph invariance etc. Before a list of invariants under isomorphism is presented, a formal description of isomorphism will be provided.
Given a bijective function \( f: V \rightarrow V' \), the following two properties are equivalent

1. \( f \) is an isomorphic function, i.e. \( f \) is such that,
   \[ \forall (u, v) \in V \times V, (u, v) \in E \iff (f(u), f(v)) \in E', \]
   where \( V, E \) are vertex and edge set of one graph and \( V', E' \) are vertex and edge set of another graph.
2. \( \forall (u, v) \in V \times V, \delta(u, v) = \delta(f(u), f(v)), \]
   where \( \delta \) is an operation on the edge.

Graph isomorphism is a phenomenon that describes the similarity of two graphs. A formal definition of graph isomorphism is given in the following.

Two graphs \( G_1 \) and \( G_2 \) represented by adjacency matrices \( A \) and \( B \) are isomorphic if and only if there exist a permutation matrix \( P \) such that \( A = PBP^{-1} \) or \( A = PBP^T \).

OR

Two graphs \( G_1 = (V_1, E_1) \) and \( G_2 = (V_2, E_2) \) are isomorphic if there exists a one-to-one mapping \( \Phi \) from \( V_1 \) onto \( V_2 \) such that \( (x, y) \in E_1 \) if and only if \( (\Phi(x), \Phi(y)) \in E_2 \).

OR

Two graphs \( G_1 \) and \( G_2 \) are said to be isomorphic if there exists a one-to-one correspondence between their vertices and edges such that the incidence relationship is preserved. That is, suppose that edge \( e_1 \) is incident on vertices \( v_1 \) and \( v_2 \) in \( G_1 \); then the corresponding edge \( e_1' \) in \( G_2 \) must be incident on the vertices \( v_1' \) and \( v_2' \) that correspond to \( v_1 \) and \( v_2 \), respectively. Hence, graph isomorphism finds a bijection between the vertices of \( G_1 \) and \( G_2 \) such that any two vertices in \( G_1 \) are adjacent if the corresponding vertices are adjacent in \( G_2 \).

OR

More formally, two graphs \( G_1 = (V_1, E_1) \) and \( G_2 = (V_2, E_2) \), such that \( |V_1| = |V_2| \) are isomorphic if there exists a permutation \( P \) such that
\[(a, b) \in E_1 \iff (P(a), P(b)) \in E_2\]

Many properties of graphs remain invariant under isomorphism and are used to check for isomorphism. Some of the properties/characteristics of the graphs that remain invariant under isomorphism include:

- **Order**.
- **Size**.
- The number of **components**.
- The **degree** sequence.
- For any **subgraph**, the **number of distinct copies of that subgraph**.
- **Radius**.
- **Eccentricity of a vertex**.
- **Centers**.
- **Diameter**.
- **Girth** - The length of the shortest cycle contained in the graph.
- **Clustering coefficient** - A clustering coefficient is a measure of degree to which nodes in a graph tend to cluster together or the clustering coefficient of an undirected graph is a measure of the number of triangles in a graph.
- **Vertex covering number** - The minimal number of vertices needed to cover all edges.
- **Vertex connectivity** - The smallest number of vertices whose removal disconnects the graph.
- **Vertex chromatic number** - The minimum number of colors needed to color all vertices so that adjacent vertices have a different color.
- **Edge connectivity** - The smallest number of edges whose removal disconnects the graph.
- **Edge chromatic number** - The minimum number of colors needed to color all edges so that adjacent edges have a different color.
- **Independence number** - The largest size of an independent set of vertices.
- **Clique number** - The largest order of a complete subgraph.
• **Algebraic connectivity**: The algebraic connectivity of a graph $G$ is the second-smallest eigenvalue of the laplacian matrix of $G$. This eigenvalue is greater than zero if and only if $G$ is a connected graph.

• **Edge Covering Number** - The minimal number of edges needed to cover all vertices.

• **Graph Genus** – The genus of the graph is the minimal integer $n$, such that the graph can be drawn without crossings on a sphere using $n$ handles. The planar graph has a genus of zero.

• **Wiener Index**: The sum of the lengths of the shortest paths between all pairs of vertices.

• **Strength**: The strength of an undirected graph corresponds to the minimum ratio of edges removed/components created during a decomposition of the graph in question. It is a method to compute partitions of the set of vertices and detect zones of high concentration of edges.

The invariants listed above can be used for proving the isomorphism amongst the graphs. But finding a sufficient set of invariant characteristics for ascertaining isomorphism continues to be an open problem, also computing the above mentioned features is an involved task. The spectral properties of the matrices representing the graph generally characterize the graphs and summarize many graph properties; hence they have been a popular choice for investigating graph isomorphism. Spectral graph theory is a field of mathematics that aims to characterize the global structural properties of graphs using the eigenvalues and eigenvectors of matrix representation of the graphs. In this work invariance property of graphs is established by checking for number of vertices, correspondence of degrees and average shortest distance of each of the vertex with other vertices. If two graphs to be verified have same invariants (number of vertices, degrees and average shortest distance) then the spectral properties of the graphs are evaluated and used for establishing vertex correspondences between graphs.

In this chapter we prove that the equivalence of the average shortest distance from a vertex to others, and correspondence of such vertices between graphs leads to similarity
between graphs. Then spectral properties namely rank ordering of the coefficients of leading/ principal eigenvector are further employed to establish correspondence between vertices and edges. The ensuing section presents a new theorem, and a relevant corollary for considering the average shortest distance along with other graph parameters as necessary and sufficient condition for graph isomorphism.

2.2 Graph Invariance using Average Shortest Distance Correspondence

In this section a theoretical formulation for graph matching is provided, using average shortest distance from a vertex to all other vertices (ASD), vertex degree and vertex eccentricity as invariant parameters and correspondence of such vertices, between the two graphs. The formulation has been shown for a special class of graphs namely the path graphs. It is assumed that every graph can be split into edge disjoint subgraphs each of which are path graphs between different vertices. Hence the methodology is proved for the path graphs which can be further extended to all the graphs. Further it has been found experimentally that the invariants hold for other simple graphs also. The ensuing subsection 2.2.1 states and proves the theorem and a corollary for graph isomorphism of path graphs.

2.2.1 Isomorphism in Path Graphs

The path graphs are graphs containing a simple path, with all the vertices other than the end vertices having a degree two and the end vertices have a degree of one. The theorem 2.1 provides a basis for finding similarity of two path graphs/ isomorphism between two path graphs

Theorem 2.1

The invariance of degree, eccentricity and shortest distance sum between corresponding vertices of two path graphs is a necessary and sufficient condition for path graph isomorphism.
**Proof:** A path graph is a graph of $n$ vertices and $e$ edges such that two vertices are of degree one and $n-2$ vertices are of degree two. The necessary condition can be proved by showing that the equality / invariance of the three parameters give a similar structure to the two path graphs. Further the sufficiency condition can been proved by showing if two path graphs have invariance of the three parameters then path graphs derived from it (extended path graph) also have the equivalence and invariance of these parameters and the extended path graphs will continue to be isomorphic and hence the condition is sufficient and by mathematical induction will hold for all the extended path graphs.

**Proof of necessary condition:** The inequality/ variance of degree between corresponding vertices of two path graphs, indicates that the two graphs are dissimilar in structure. The inequality/ variance of eccentricities of corresponding vertices of two path graphs again indicates dissimilarity in structure of the path graphs as eccentricity gives the largest of the shortest distance to other vertices.

The shortest distance sum of the corresponding vertices should be the same as all the other vertices are equidistant from a specified vertex (for similarly structured path graphs). The inequality of these values shows that the graphs are dissimilar. Hence the graphs are similar/ isomorphic only if the three parameters are simultaneously equal, between corresponding vertices. Hence the conditions of theorem 2.1 are necessary for two path graphs to be isomorphic.

**Proof for sufficiency condition:** To prove the sufficiency condition it is shown that equality of graph parameters, gives the same structure which persists while extending path graphs.

Let $\sigma_i, \ \sigma_i'$: Represent the shortest path sum of $i^{th}$ vertex with every other vertex in the path graph and extended path graph

$\Delta_i, \ \Delta_i'$: Represent the degree of the $i^{th}$ vertex in the path graph and extended path graph
$e_i$, $e_i'$: Represent the eccentricity of the $i$th vertex in the path graph and extended path graph

The extended path graphs are obtained by adding an edge and a vertex to the end vertex of the path graph

The parameters $\sigma_i', \Delta_i', e_i'$ will be obtained from the parameters of the previous path graphs, using the logically obtained formulas. 2.1, 2.2 and 2.3 which are devised by logical reasoning.

$\Delta_i' = 1$, for added vertex

$\Delta_i' = \Delta_i + 1$, for the vertex to which an edge gets added

$\Delta_i' = \Delta_i$, for all other vertices

$e_i' = Max(e_i) + 1$, for the added vertex

$e_i' = e_i$, for the vertex to which an edge gets added

$e_i' = e_i$, if $d_1 = d_2$, where $d_1$ is the distance of the $i$th vertex from added vertex, $d_2$ is the distance of $i$th vertex from the other end vertex

$e_i' = e_i + 1$, for other vertices

$\sigma_i' = \sigma_i + 1$, for the vertex to which an edge gets added

$\sigma_i' = Max(\sigma_i) + e_i'$, for added vertex

$\sigma_i' = \sigma_i + e_i' - (d_2 - d_1)$, for vertices whose degree does not change

if $d_1 \geq 2$ & $d_2 \geq n/2$

$\sigma_i' = \sigma_i + e_i'$, for vertices whose degree does not change and $d_2 \geq n/2$
So the parameters which are essential for a particular structure of an extended path graph can be computed from parameters of the previous path graph and hence if two path graphs are isomorphic then extended paths graphs are also isomorphic. This is further established in the following.

Consider the graphs $G_1$, $G_2$, $G_3$ and $G_4$ given in Figure 2.1. The graph $G_2$ is an extended path graph of $G_1$, $G_3$ is an extended path graph of $G_2$, and $G_4$ is the extended path graph of $G_3$. The Table 2.1 lists the parameters, shortest path sum and eccentricity for each of the graphs. The parameter degree ($\Delta$) is not explicitly listed as the continued equality for extended path graphs is obvious. It can be seen from the Table 2.1 that the parameters of extended path graphs can be obtained using the formulas 2.2.1 to 2.2.4 and 2.3.1 to 2.3.4 as listed in the Table.

![Graph G1, Graph G2, Graph G3, Graph G4](image)

**Figure 2.1:** Graphs for finding Shortest Path Sums

**Table 2.1:** Graph Parameters of Path Graphs/ Extended Path Graphs

<table>
<thead>
<tr>
<th>Sl No</th>
<th>Graph</th>
<th>Vertex</th>
<th>Shortest Path Sum (Observed)</th>
<th>Eccentricity</th>
<th>Exended Graph</th>
<th>Vertex</th>
<th>Shortest Path Sum (Observed)</th>
<th>Eccentricity</th>
<th>Formula applicable</th>
<th>Shortest Path Computed using the formula</th>
<th>Formula applicable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$G_1$</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>$G_2$</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>(2.2.2)</td>
<td>4</td>
<td>2.3.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td>1</td>
<td></td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>(2.2.4)</td>
<td>4</td>
<td>2.3.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
<td>2</td>
<td></td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>(2.2.4)</td>
<td>6</td>
<td>2.3.4</td>
</tr>
</tbody>
</table>
Now consider two path graphs, $H_1$ and $H_2$ of $n=3$ vertices, the parameters, vertex degrees, vertex eccentricity and shortest distance sum are equivalent for corresponding vertices of $H_1$ and $H_2$, due to theorem 3.1 these are isomorphic. Now let us extend the path graphs to $H_1'$ and $H_2'$. As per the discussions above, these two graphs are also isomorphic. These arguments can be extended to path graphs of any number of vertices $N$. The application of the above listed formulas indicates that extended path graphs of $N+1$ vertex are also isomorphic. Hence, the proof.

The results of theorem 2.1 can be extended by replacing the parameter, sum of shortest path distance to all other distances by average shortest distance of each vertex to every other vertex. This results in the corollary 2.1.1.
**Corollary 2.1.1:** The invariance of degree, eccentricity and average shortest distance between corresponding vertices of the two path graphs is necessary and sufficient conditions for graph isomorphism.

**Proof:**

This statement is derived from theorem 2.1 and the only change is in the parameter is average shortest distance which is derived from shortest distance sum. The average shortest distance $\bar{d}_i = \frac{\sigma_i}{n-1}$ hence as theorem 2.1 is proved to be correct the corollary 2.1.1 is also true.

The theorem 2.1 and the corollary 2.1.1 are stated and proved for path graphs, but the invariance of the parameters indicated in these are also demonstrated to be true for all the simple graphs by experimentation and forms the basis of all the methodologies proposed in this work. In the work described in this chapter, along with these characteristics the spectral properties are employed for establishing vertex correspondence and are explained in the ensuing section.

**2.3 Spectra of Graph for Vertex Correspondence**

Algebraic graph theory is a branch of mathematics that studies graphs by using algebraic properties of the matrices that represent them. Spectral graph theory studies the relation between graph properties and the spectrum of the adjacency matrix or laplacian matrix (or any other matrices) that are used to represent them. Many spectral properties and their correspondence to graph properties have been stated and enlisted in Chapter 3 of this thesis. Spectral graph theory looks at the connection between the eigenvalues and eigenvectors of a matrix associated with a graph and the corresponding structures of a graph.

Many properties of the graphs can be deduced by using the eigenvalues of the matrices such as, the number of edges of the graphs from the eigenvalues, the number of connected
components etc. Whether the graphs are bipartite can be determined using the spectral values of adjacency matrix. For many people this is the extent to which they see the eigenvalues of matrices applied to graphs, and while these are interesting graph parameters to measure they do not show the strength of spectral graph theory. In this work the spectral property correspondence has been used to prove the similarity of graphs and establish vertex correspondence.

The fact that eigenvalue multiplicities must be integral provides strong restrictions on graphs and their correspondences [Brouwer, 2011]. The maximum eigenvector coefficients have a proportional value for ranking of vertices in a graph [Robles-Kelly and Hancock, 2001] and can be employed for finding vertex correspondence. This spectral property of the adjacency matrix is brought out in the observation 2.2 and is employed in this work. The ensuing sections describes the methodology employed for finding whether two graphs are same/ similar/ isomorphic and further for obtaining vertex/edge correspondences.

2.4 The Graph Matching Methodology

The proposed methodology makes use of the adjacency matrix representation of the graphs. The spectral properties of the adjacency matrix namely the eigenvalues/eigenvectors are extracted and used for systematically checking whether the graphs are isomorphic, along with vertex degrees and average shortest distance to other vertices.

Observation 2.2

The non-increasing order of the coefficients of the principal eigenvectors of the two isomorphic graphs provides a ranking to vertices. And further, the similarly ranked vertices provide a possible vertex correspondence.
The vertices of the two graphs are separately ordered by using the rank of the coefficients of largest/ principal eigenvector. Then rank of vertices is used for ordering and further checking for similarity of degrees and average shortest distance to the other vertices (ASD). If the corresponding degree’s and ASD’s, of the similarly ranked vertices match (for all vertices), then the two graphs are said to be similar/ isomorphic. The complete methodology is brought out in the flow chart of Figure 2.2. The different concepts and methodologies used are explained in the following subsections.

Figure 2.2: Flowchart describing the algorithm for finding Graph Isomorphism

Contd...
Figure 2.2: Flowchart describing the algorithm for finding Graph Isomorphism

Legend:

ROG: Rank Order of vertices in Graphs
### 2.4.1 Adjacency Matrix and Eigenvectors.

Let \( G_1(V_1,E_1) \) and \( G_2(V_2,E_2) \) be graphs of \( n \) vertices and \( m \) edges, where,

\[ V_1(G) = \{v_1,v_2,\ldots,v_n\} \quad \text{and} \quad E_1(G) = \{e_1,e_2,\ldots,e_m\}. \]

The adjacency matrix of \( G_1 \) is the \( n \times n \) matrix, constructed as in equation (2.4)

\[
X = \begin{bmatrix} x_{ij} \end{bmatrix} \quad \text{... (2.4)}
\]

where, \( x_{ij} = 1 \), if \( \langle v_i,v_j \rangle \) belongs to \( E(G) \)

\[ = 0, \text{otherwise} \]

To find graph isomorphism it is proposed to assign correspondence between vertices of the graph \( G_1 \) to the vertices of graph \( G_2 \) by using largest eigenvector coefficients and average shortest distance along with vertex degrees. Firstly, the eigenvalues associated with the adjacency matrix of each graph are found. The eigenvalues of graph \( G_1 \) are found by solving the polynomial equation \( |G_1 - \lambda I| = 0 \). The eigenvector \( \Phi_i \) associated with the eigenvalue \( \lambda_i \), are found by solving the system of linear equations depicted by 2.5

\[
G_1\Phi_i = \lambda_i \Phi_i \quad \text{... (2.5)}
\]

The equation \( |G_1 - \lambda I| = 0 \) and \( G_2\Phi_i = \lambda_i \Phi_i \) and similar others are solved using algebraic techniques.

Let the leading eigenvector of adjacency matrix of graph \( G_1 \) be \( \Phi_i \) as denoted by (2.6)

\[
\Phi_i = (\phi(1),\phi(2),\ldots,\phi(n))^T \quad \text{... (2.6)}
\]

while that of \( G_2 \) be \( \Phi_j \) as denoted by (2.7)

\[
\Phi_j = (\phi'(1),\phi'(2),\ldots,\phi'(n))^T \quad \text{... (2.7)}
\]

The eigenvalues associated with the eigenvectors \( \Phi_i \) and \( \Phi_j \) are \( \lambda_i \) and \( \lambda_j \).
Further, these leading eigenvectors are used to assign rank order to vertices of the graphs $G_1$ and $G_2$ by employing the magnitudes of the coefficients in the eigenvector. The rank order of the vertices in the graphs is used for locating correspondence between vertices. The rank order of the nodes in $G_1$ is given by the sorted node-indices as depicted in equation (2.8).

$$O_i = (i_1, i_2, \ldots, i_n), \text{ where } \phi_i(i_1) > \phi_i(i_2) > \ldots > \phi_i(i_n) \quad \ldots (2.8)$$

Similarly the rank order of vertices in $G_2$ is given by equation (2.9)

$$O_j = (j_1, j_2, \ldots, j_n), \text{ where } \phi'_j(j_1) > \phi'_j(j_2) > \ldots > \phi'_j(j_n) \quad \ldots (2.9)$$

To find whether the two graphs match; first the numbers of the nodes are compared if they are different then the two graphs cannot be similar, otherwise, the corresponding nodes of the same rank order are compared for the equality of degrees and average shortest distance. If they are equal for all nodes then the graphs are isomorphic. The complete methodology is brought out by the flowchart given in Figure-2.2.

### 2.4.2 Matching the nodes

The matching/correspondence of nodes of the two graphs $G_1$ and $G_2$ for isomorphism is achieved by first ranking the vertices of the graphs based on the coefficients of the leading eigenvectors of the two adjacency matrices representing them. Further each vertex with the same rank is verified for equal degree and later for equality in average shortest distance (as per the arguments in section 2.2). The average shortest distance is found by computing the shortest distance to every other vertex and dividing their sum by $n-1$. If the above test is positive for all the vertices, it is concluded that the two graphs are similar/isomorphic, further the vertex correspondence and hence the edge correspondence is established. A typical example application of the proposed methodology is given in the following subsection.
2.4.3 Example

Consider the two graphs $G_1$ and $G_2$ shown in Figure 2.3.

The adjacency matrices of these graphs are shown in Figure 2.4.

$X(G_1)$

\[
\begin{pmatrix}
0 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0
\end{pmatrix}
\]

$X(G_2)$

\[
\begin{pmatrix}
0 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0
\end{pmatrix}
\]
The eigenvalues and leading eigenvectors of the two graphs are shown in Figure 2.5

**EigenValues of** $X(G_1) = \begin{cases} -1.8608 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.61803 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.25410 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.61803 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.1149 \end{cases}$

**EigenValues of** $X(G_2) = \begin{cases} -1.8608 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.61803 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.25410 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.61803 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.1149 \end{cases}$

**Leading EigenVector of** $X(G_1) = \begin{cases} -0.3851 \\ -0.4294 \\ -0.5230 \\ -0.4294 \\ -0.3851 \\ -0.2473 \end{cases}$

**Leading EigenVector of** $X(G_2) = \begin{cases} -0.5230 \\ -0.4294 \\ -0.3851 \\ -0.3851 \\ -0.2473 \\ -0.4294 \end{cases}$

**Figure 2.5:** Eigen Values and Leading Eigen Vectors
After sorting the absolute values of coefficients of the leading eigenvector, the vertices of graphs are arranged in non increasing order. One to one correspondence is assumed between the vertices of the same rank. This is further verified for equivalence of degree and average shortest distance with all the other vertices in a graph. The rank order of the vertices of graph $G_1$ and $G_2$, their degrees and average shortest distances are listed in Table-2.2.

**Table 2.2: Rank order, degrees and average shortest distances of vertices**

<table>
<thead>
<tr>
<th>Sl.No.</th>
<th>The vertex numbers in rank order in graph $G_1$</th>
<th>Average Shortest distance with other vertices of $G_1$</th>
<th>Degree of Vertices in $G_1$</th>
<th>The vertex numbers in rank order in graph $G_2$</th>
<th>Average shortest distance with other vertices of $G_2$</th>
<th>Degree of Vertices in $G_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>1.4</td>
<td>3</td>
<td>1</td>
<td>1.4</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1.6</td>
<td>2</td>
<td>2</td>
<td>1.6</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>1.6</td>
<td>2</td>
<td>6</td>
<td>1.6</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1.8</td>
<td>2</td>
<td>3</td>
<td>1.8</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1.8</td>
<td>2</td>
<td>4</td>
<td>1.8</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>2.2</td>
<td>1</td>
<td>5</td>
<td>2.2</td>
<td>1</td>
</tr>
</tbody>
</table>

As the values of the degree and average shortest distance of the corresponding vertices at the same rank order are same, the two graphs are isomorphic and the vertex correspondences are as given in the Table-2.3.

**Table 2.3: Vertex correspondence of graphs $G_1$ and $G_2$**

<table>
<thead>
<tr>
<th>Graph $G_1$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph $G_2$</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
The above methodology is implemented using MATLAB; synthetic graphs having different number of vertices are used for testing and more than hundred graph pairs have been tested and the results are brought out in section 2.5.

2.5 The Experimentation

The methodology for graph matching proposed in this chapter is implemented using MATLAB R12. The method has been applied to synthetic graphs having different number of vertices and edges and the results are enlisted in Table 2.4 below. About hundred and sixteen pairs of graphs have been tested and accurate results are achieved. Among these hundred and sixteen pairs, about sixty eight graph pairs where isomorphic and forty eight were not isomorphic. All the tests have been carried out very efficiently and the results are found to be correct for all the types of graphs. Amongst the graphs tested typical graphs such as those in Figure 2.6 which are counter examples for isomorphism testing are also included, and the results are as expected (they are not isomorphic). The two graphs seem to be isomorphic on the first look as they have same number of vertices, same number of vertices with degree k, ∀ k=1..3, but actually they are not isomorphic[Narasingh Deo, 2004].

![Figure 2.6: Typical non isomorphic graphs for testing](image)
Table 2.4: The Experimentation Results

<table>
<thead>
<tr>
<th>Sl.No.</th>
<th>Number of Vertices</th>
<th>Number of Graph Pairs</th>
<th>Number of Isomorphic Graph Pairs</th>
<th>Number of Non Isomorphic Graph Pairs</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>7</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>7</td>
<td>5</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>12</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>13</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>14</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>15</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>16</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>17</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>18</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>19</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>20</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>21</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>22</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>23</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>24</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>25</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>30</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

All the tests have been carried out on synthetic graph pairs and the results are correct for all the types of graphs. Amongst the graphs tested typical graphs for isomorphism testing are also included.

The results are also brought on the bar chart given in Figure 2.7.

![Bar Chart showing results of Isomorphism Testing using the proposed Methodology](image)

Figure 2.7: Bar Chart showing results of Isomorphism Testing using the proposed Methodology
The average time taken by the methodology for isomorphism testing is listed in Table 2.5 for a few synthetic graph pairs and the same information is plotted in Figure 2.8 for the graphs which are isomorphic/similar.

**Table 2.5: Average Time for Graph Matching**

<table>
<thead>
<tr>
<th>Sl No</th>
<th>No. of Vertices</th>
<th>No. of Edges</th>
<th>Isomorphic (Y/N)</th>
<th>Correct Mapping of Vertices (Y/N)</th>
<th>Time for Computation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>6</td>
<td>Y</td>
<td>Y</td>
<td>0.0938</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>4</td>
<td>Y</td>
<td>Y</td>
<td>0.0313</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>6</td>
<td>N</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>7</td>
<td>Y</td>
<td>Y</td>
<td>0.0156</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>N</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>Y</td>
<td>Y</td>
<td>0.0156</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>9</td>
<td>Y</td>
<td>Y</td>
<td>0.0156</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>10</td>
<td>N</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>8</td>
<td>Y</td>
<td>Y</td>
<td>0.0313</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>8</td>
<td>Y</td>
<td>Y</td>
<td>0.0156</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>9</td>
<td>N</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>8</td>
<td>Y</td>
<td>Y</td>
<td>0.0156</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>7</td>
<td>N</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>14</td>
<td>9</td>
<td>11</td>
<td>Y</td>
<td>Y</td>
<td>0.0313</td>
</tr>
<tr>
<td>15</td>
<td>9</td>
<td>11</td>
<td>Y</td>
<td>Y</td>
<td>0.0156</td>
</tr>
<tr>
<td>16</td>
<td>9</td>
<td>11</td>
<td>Y</td>
<td>Y</td>
<td>0.0156</td>
</tr>
<tr>
<td>17</td>
<td>9</td>
<td>14</td>
<td>N</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>18</td>
<td>10</td>
<td>12</td>
<td>Y</td>
<td>Y</td>
<td>0.0469</td>
</tr>
<tr>
<td>19</td>
<td>10</td>
<td>12</td>
<td>Y</td>
<td>Y</td>
<td>0.0156</td>
</tr>
<tr>
<td>20</td>
<td>10</td>
<td>10</td>
<td>N</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>21</td>
<td>10</td>
<td>11</td>
<td>Y</td>
<td>Y</td>
<td>0.0156</td>
</tr>
<tr>
<td>22</td>
<td>11</td>
<td>13</td>
<td>N</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>23</td>
<td>11</td>
<td>13</td>
<td>Y</td>
<td>Y</td>
<td>0.0156</td>
</tr>
<tr>
<td>24</td>
<td>11</td>
<td>13</td>
<td>Y</td>
<td>Y</td>
<td>0.0156</td>
</tr>
<tr>
<td>25</td>
<td>11</td>
<td>13</td>
<td>Y</td>
<td>Y</td>
<td>0.0156</td>
</tr>
<tr>
<td>26</td>
<td>11</td>
<td>13</td>
<td>Y</td>
<td>Y</td>
<td>0.0156</td>
</tr>
<tr>
<td>27</td>
<td>11</td>
<td>13</td>
<td>N</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>28</td>
<td>14</td>
<td>13</td>
<td>Y</td>
<td>Y</td>
<td>0.0156</td>
</tr>
<tr>
<td>29</td>
<td>15</td>
<td>15</td>
<td>N</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>30</td>
<td>20</td>
<td>16</td>
<td>Y</td>
<td>Y</td>
<td>0.0313</td>
</tr>
<tr>
<td>31</td>
<td>22</td>
<td>24</td>
<td>Y</td>
<td>Y</td>
<td>0.0156</td>
</tr>
<tr>
<td>32</td>
<td>24</td>
<td>27</td>
<td>Y</td>
<td>Y</td>
<td>0.0156</td>
</tr>
<tr>
<td>33</td>
<td>26</td>
<td>32</td>
<td>Y</td>
<td>Y</td>
<td>0.0313</td>
</tr>
<tr>
<td>34</td>
<td>28</td>
<td>36</td>
<td>Y</td>
<td>Y</td>
<td>0.0469</td>
</tr>
<tr>
<td>35</td>
<td>30</td>
<td>40</td>
<td>Y</td>
<td>Y</td>
<td>0.0469</td>
</tr>
</tbody>
</table>
The plot of time for isomorphism testing v/s number of vertices shows that for small number of vertices, the time taken is almost the same. As the number of vertices increase, the time taken by the methodology increases, albeit slowly. The complete time analysis of the methodology is provided in Chapter 6.

2.6 Summary

A new technique is presented in this chapter to check for graph matching using the spectral properties of the adjacency matrix of graph and average shortest distance to other vertices in the graph. The proposed method, first finds eigenvalues and eigenvectors of adjacency matrix representation of the graphs, then the vertices of the graphs are arranged according to rank order (non-increasing) considering the coefficients of the largest eigenvector. Average shortest distance of a vertex to all the other vertices and degrees of the vertices are further employed for verifying graph isomorphism. The vertices of the same ranks are proposed to be the corresponding vertices and are verified by the equality of the average shortest distances and degree. The technique developed has been tested for various synthetic graphs of different vertices ranging from four to thirty and 100% accurate results are obtained. The proposed methodology is fast and can be employed for different applications that make use of the simple undirected graphs.