Chapter - V
Graph Kernel Matching in Matrimonial Database System
CHAPTER 5

GRAPH KERNEL MATCHING IN MATRIMONIAL DATABASE SYSTEM

This chapter describes the Implementation of Graph kernel matching Technique in Matrimonial Data Base System.

Graph mining in Data Base Management System has become an important topic of research recently because of numerous applications to a wide variety of identification problems in current educational system. Nowadays Graphs play a vital role everywhere, occupying the social networks and mobile networks to biological networks and the World Wide Web. Mining big graphs leads too many interesting applications including marketing, news groups, community mining, and many more [12]. This research describes a technique for the implementation of encoding schema problem for confidentiality management to a Graph Mining pattern.

These work findings include designs to survey different aspects of graph mining and encoding-decoding environment [13]. The results are revealed for selecting the optimized encoding and decoding schema for the cricket player identification based implementation towards selection strategies. This will lead to propose a Graph-Analysis Implementer for any real-time complex entities [11].
5.1 Experimental Schema Methodology

Based on the objective function, matching problems can be classified as follows:

1. **Maximum (Cardinality) Matching:** Maximize the number of edges in the matching.

2. **Maximum Edge Weight Matching:** Maximize the sum of weights of Matched edges.

3. **Maximum Vertex Weight Matching:** Maximize the sum of weights of matched vertices.

Matching in a bipartite graph is easier to compute than in general (or no bipartite) graphs. Similarly, the unweighted versions are easier than the weighted versions of the matching problem. The weighted versions may also have additional restrictions on the cardinality of the matching, e.g., a maximum weight matching among all matching of maximum cardinality.

Matching algorithms compute optimal solutions in polynomial time with the help of techniques like augmentation, blossoms and primal-dual formulations. However, these polynomial time algorithms can still be slow for many scientific computing applications. Approximation algorithms become important when matching needs to be computed a large number of times for a given application (for example, multi-level algorithms), for massive graphs, or in applications with resource limitations (for example, high-speed network switches that implement matching algorithms in hardware with severe restrictions on available memory and high performance requirements).
The need for parallel algorithms arise when matching needs to be computed on massive graphs, such as the ones arising from web applications, or when the graph is redistributed on the processors of a parallel computer.

**Proposed Methodology**

This proposed methodology focuses on the implementation of a graph matching algorithmic strategy to predict the unknown node behaviors by implementing the kernel computations.

![Proposed Graph matching structure](image)

**Figure 5.1 Proposed Graph matching structure**

**Implementation of Algorithmic strategies**

Consider the possible cluster graphs with unknown node behaviors as follows: The cluster contains 28 nodes with 4 levels (0, 1, 2 and 3). Each node works well and earns their clients as child based on their promotional credit (P). But some nodes are not function well due to its Non promotional credit (NP) also with exceptions. In order to measure the similarity between two graphs, we need to measure the similarity between nodes, edges, and paths.
Node / Edge kernel: An example of a node/edge kernel is the identity kernel. If two nodes/edges have the same label, then the kernel returns 1 otherwise 0. It is denoted by NEK(G).

Path kernel: A path is a sequence of node and edge labels. If two paths are of the same length, the path kernel can be constructed as the product of node and edge kernels. If two paths are of different lengths, the path kernel simply returns 0. It is denoted by PK(G).

Graph kernel: As each path is associated with a probability, we can define the graph kernel as the expectation of the path kernel over all possible paths in the two graphs. It is denoted by GK(G). A graph g1 is matched with g2 iff NEK(G1,G2) PK(G1,G2) GK(G1,G2)=1.

5.2 Experimental Schema Implementation

Consider the sample matrimonial database of which p denotes the parent and b, g denotes the boy and girl respectively. The shaded node represents the particular boy or girl in the family got married. The proposed algorithmic procedure is as follows:

1. START
2. Convert the request for bride or bride groom into a graph G1.
3. The matrix of Database elements are converted as G2.
4. Perform the computation NEK(G1,G2), PK(G1,G2) and GK(G1,G2).
5. If NEK(G1,G2) PK(G1,G2) GK(G1,G2)=1 for any G2 then “MATCH FOUND” extract the Graph family
6. Else “NO MATCH FOUND”
7. STOP.
Now consider the sample database of Matrimonial information as follows,

\[ \text{Figure 5.2 Graph matching sample space} \]
Consider the sample request as follows:

A family of 3 children containing the least age only boy with unmarried status.

The implementation graph matching is as follows, moving towards the matrix of 3 children graphs as G2 and now G1 becomes,

Converting the request to a graph G1.

\( G1=(1,1) \) element graph of 3 children graph, therefore

\[ NEK(G1,G2) \ PK(G1,G2) \ GK(G1,G2)=0 \]

\( G1=(1,2) \) element graph of 3 children graph, therefore

\[ NEK(G1,G2) \ PK(G1,G2) \ GK(G1,G2)=0 \]

\( G1=(1,3) \) element graph of 3 children graph, therefore

\[ NEK(G1,G2) \ PK(G1,G2) \ GK(G1,G2)=0 \]

\( G1=(2,1) \) element graph of 3 children graph, therefore

\[ NEK(G1,G2) \ PK(G1,G2) \ GK(G1,G2)=0 \]

\( G1=(2,2) \) element graph of 3 children graph, therefore

\[ NEK(G1,G2) \ PK(G1,G2) \ GK(G1,G2)=0 \]

\( G1=(2,3) \) element graph of 3 children graph, therefore

\[ NEK(G1,G2) \ PK(G1,G2) \ GK(G1,G2)=0 \]

\( G1=(3,1) \) element graph of 3 children graph, therefore

\[ NEK(G1,G2) \ PK(G1,G2) \ GK(G1,G2)=0 \]
$G_1 = (3,2)$ element graph of 3 children graph, therefore

$\text{NEK}(G_1, G_2) \text{ PK}(G_1, G_2) \text{ GK}(G_1, G_2) = 0$

$G_1 = (3,3)$ element graph of 3 children graph, therefore

$\text{NEK}(G_1, G_2) \text{ PK}(G_1, G_2) \text{ GK}(G_1, G_2) = 0$

$G_1 = (4,1)$ element graph of 3 children graph, therefore

$\text{NEK}(G_1, G_2) \text{ PK}(G_1, G_2) \text{ GK}(G_1, G_2) = 0$

$G_1 = (4,2)$ element graph of 3 children graph, therefore

$\text{NEK}(G_1, G_2) \text{ PK}(G_1, G_2) \text{ GK}(G_1, G_2) = 0$

$G_1 = (4,3)$ element graph of 3 children graph, therefore

$\text{NEK}(G_1, G_2) \text{ PK}(G_1, G_2) \text{ GK}(G_1, G_2) = 0$

$G_1 = (5,1)$ element graph of 3 children graph, therefore

$\text{NEK}(G_1, G_2) \text{ PK}(G_1, G_2) \text{ GK}(G_1, G_2) = 0$

$G_1 = (5,2)$ element graph of 3 children graph, therefore

$\text{NEK}(G_1, G_2) \text{ PK}(G_1, G_2) \text{ GK}(G_1, G_2) = 0$

$G_1 = (5,3)$ element graph of 3 children graph, therefore

$\text{NEK}(G_1, G_2) \text{ PK}(G_1, G_2) \text{ GK}(G_1, G_2) = 1$

Therefore MATCH FOUND with the family element (5, 3). Hence the result graph extracted.
5.3 Experimental Schema Result

The implementation of our proposed methodology computes the expectation of node behaviors in a predictable way. The final requested family may obtain the following desired structures if implemented in an optimistic approach as follows:

![Diagram](image-url)

**Figure 5.3** Successful implementation of Proposed Graph matching

The identification of the requested family executed successfully through the proposed graph matching algorithmic strategies.


5.4 CONCLUSION

This chapter concludes the overall method proves to be highly efficient compared to mining significant and open trees, dramatically reducing running time and number of features mined. In this paper, we implemented the graph mining technique of graph matching with our proposed algorithmic strategy.

This graph mining techniques is based on the node, edge, path and graph kernel approaches, which are the graph mining fundamentals. In addition, the strategies are supporting the optimistic way of stimulus response feature. We also have highlighted the research contributions and found out some limitations in different research works. Consequently, this work also depicts the critical evaluation in which requisition have been taken out to show the similarities and differences among different node responsibilities equivalent to Matrimonial Database clients. The importance of this work is that it reveals the literature review of different graph mining techniques and provides a vast amount of information under a single paper.

In our future work, we have planned to propose a cluster mining method based on graph mining technique, provide its implementation and compare its results with the different existing classification based graph mining algorithms.