www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 03 Issue 07 July, 2014 Page No. 6909-6913

Optimal Implementation of Graph Kernel Matching in Matrimonial Database Using Graph Mining Technique

S. A. Amala Nirmal Doss[1] , Dr.S.P.Victor[2],
[1]*Research Scholar,MS University,Tirunelvel*
amalnirmaldoss@#rediffmail.com

[2]*HOD-CS-St.Xaviers College*
drvictorsp@rediffmail.com

Abstract-The Graph mining provides a systematic way implement real-time data with different level of implications. Our conventional setup initially focuses with dataset and its entity. This paper perform a detailed study of graph kernel matching towards variant clusters in the field of graph mining which can be carried out with request to response matching strategies. We will implement our integrated graph mining techniques with real time implementation of Matrimonial database Domains. We will also perform algorithmic procedural strategies for the successful implementation of our proposed research technique in several sampling domains with a maximum level of improvements. In near future we will implement the cluster mining techniques for predicting the Graph sub structure behaviors.

Keywords- Graph Mining, ,kernel , prediction , matrimonial

1.Introduction

Given a graph G, a matching M is a set of edges such that no two edges in M are incident on the same vertex. Matching is a fundamental combinatorial problem that has applications in many contexts: high-performance computing, bioinformatics, network switch design, web technologies, etc. Examples in the first context include sparse linear systems, where matchings are used to place large matrix elements on or close to the diagonal, block triangular decomposition, computing a sparse basis for the null space or column space of under-determined matrices, and multi-level graph partitioning algorithms where matchings are used in the coarsening phase.
Label Propagation. A subset of nodes in a graph is labeled. The task is to learn a model from the labeled nodes and use the model to classify the unlabeled nodes.

Graph classification. A subset of graphs in a graph dataset is labeled. The task is to learn a model from the labeled graphs and use the model to classify the unlabeled graphs.

Based on the objective function, matching problems can be classified into:
1. Maximum (Cardinality) Matching: Maximize the number of edges in the matching.
2. Maximum (Edge) Weighted Matching: Maximize the sum of the weights of the matched edges.
3. Maximum (Vertex) Weighted Matching: Maximize the sum of the weights of the matched vertices.

Matchings in a bipartite graph are easier to compute than in general (or nonbipartite) 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 matchings 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 predistributed on the processors of a parallel computer.

2.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. .

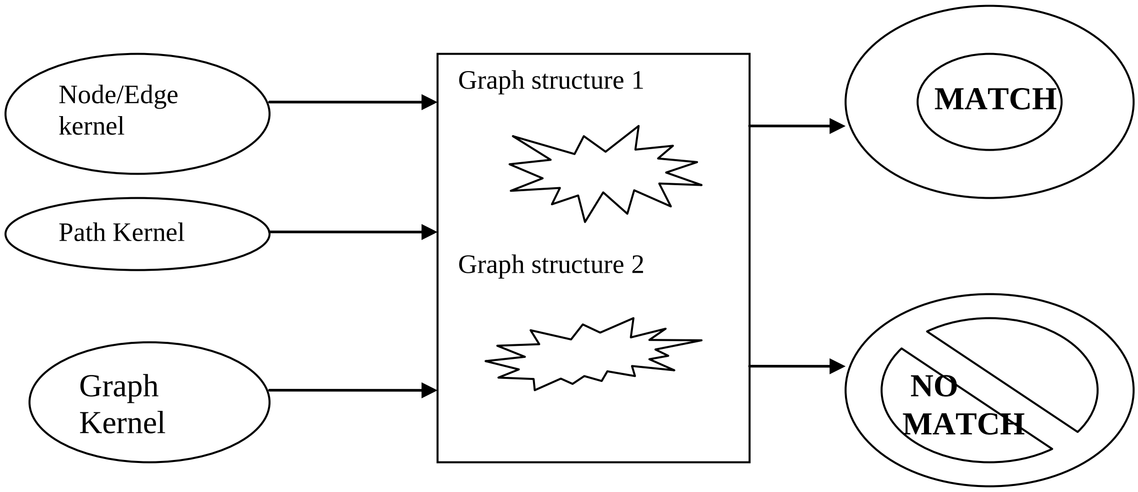


Figure 1: Proposed Graph matching structure

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) $\cap$PK(G1,G2) $\cap$GK(G1,G2)=1.

3.Experimental Methodology

Consider the sample matrimonial database of which p- denotes the parent and b,g denotes the boy and girl repectively.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.

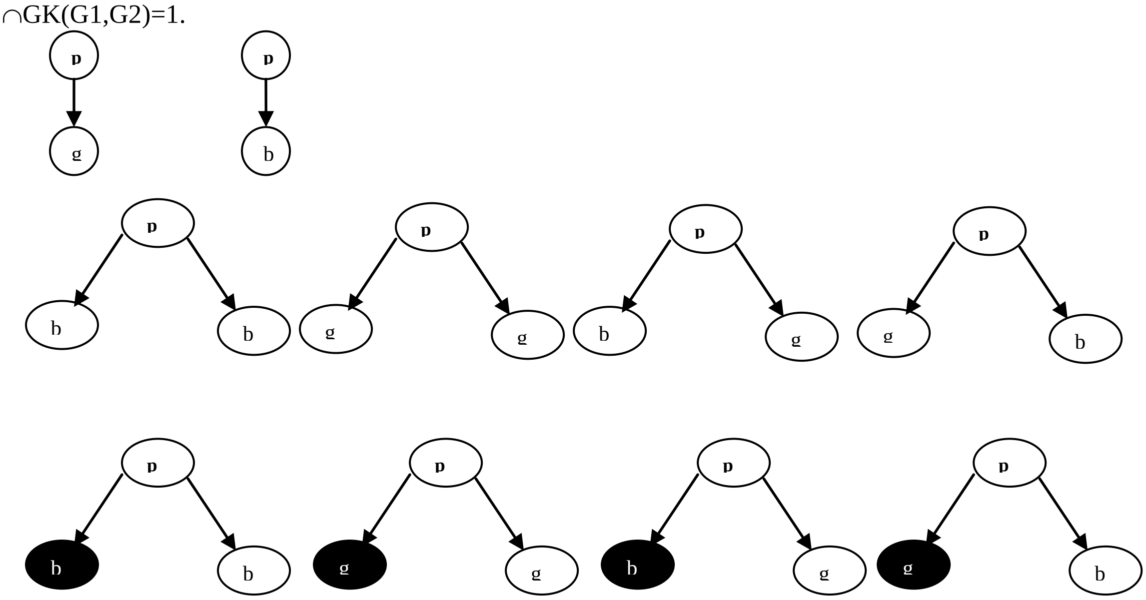
3.The matrix of Database elements are converted as G2.

4.Perform the computation NEK(G1,G2) , PK(G1,G2) and GK(G1,G2).

5.If NEK(G1,G2) $\cap$PK(G1,G2) $\cap$GK(G1,G2)=1 for any G2 then "MATCH FOUND" extract the Graph family else "NO MATCH FOUND"

6.STOP.

Now consider the sample database of Matrimonial information is as follows,



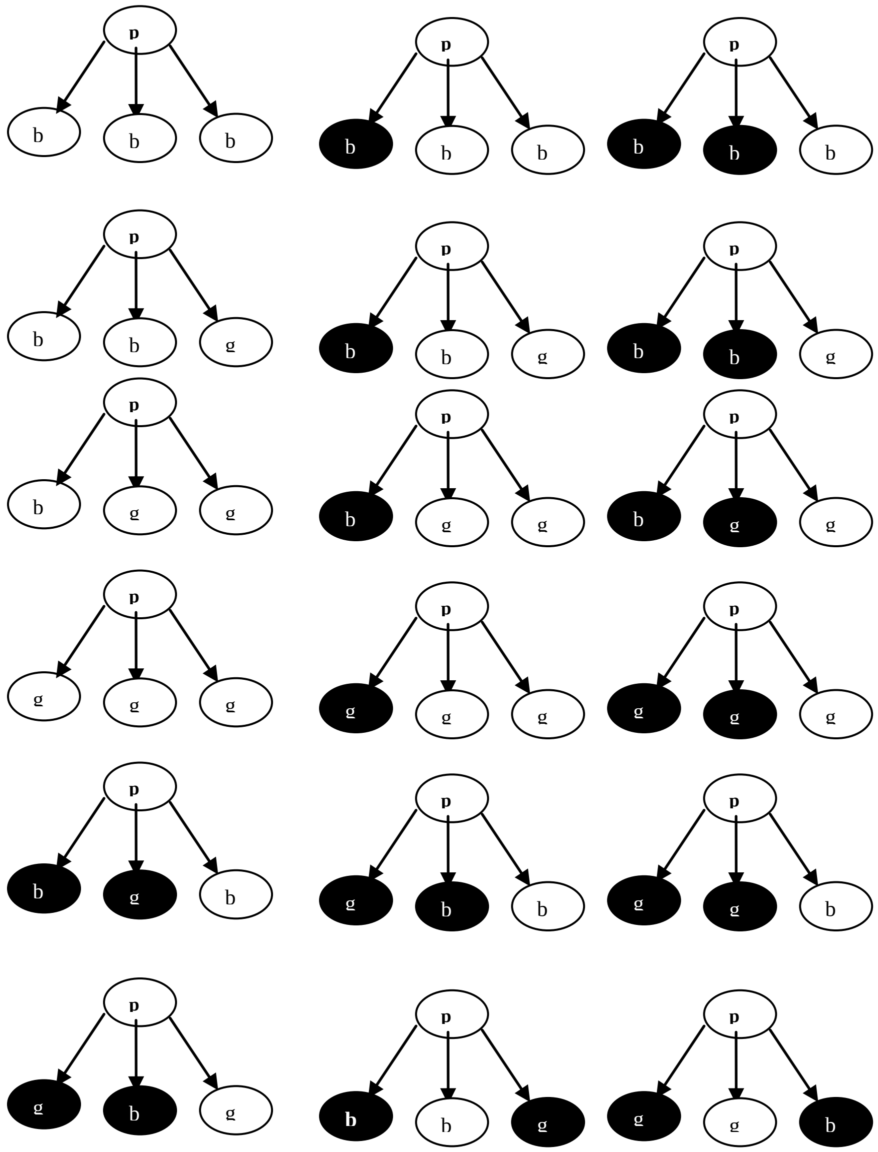
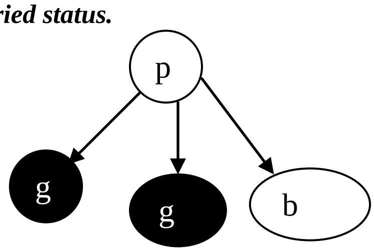


Figure 2: Sample Matrimonial database of 1x1,2x2,3x3 Family Graph. structure

4.Computation And Results

Consider the sample request as follows,

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



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

NEK(G1,G2) $\cap$PK(G1,G2) $\cap$GK(G1,G2)=0

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

NEK(G1,G2) $\cap$PK(G1,G2) $\cap$GK(G1,G2)=0

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

NEK(G1,G2) $\cap$PK(G1,G2) $\cap$GK(G1,G2)=0

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

NEK(G1,G2) $\cap$PK(G1,G2) $\cap$GK(G1,G2)=0

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

NEK(G1,G2) $\cap$PK(G1,G2) $\cap$GK(G1,G2)=0

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

NEK(G1,G2) $\cap$PK(G1,G2) $\cap$GK(G1,G2)=0

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

NEK(G1,G2) $\cap$PK(G1,G2) $\cap$GK(G1,G2)=0

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

NEK(G1,G2) $\cap$PK(G1,G2) $\cap$GK(G1,G2)=0

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

NEK(G1,G2) $\cap$PK(G1,G2) $\cap$GK(G1,G2)=0

G1=(4,1) element graph of 3 children graph,therefore

NEK(G1,G2) $\cap$PK(G1,G2) $\cap$GK(G1,G2)=0

G1=(4,2) element graph of 3 children graph,therefore

NEK(G1,G2) $\cap$PK(G1,G2) $\cap$GK(G1,G2)=0

G1=(4,3) element graph of 3 children graph,therefore

NEK(G1,G2) $\cap$PK(G1,G2) $\cap$GK(G1,G2)=0

G1=(5,1) element graph of 3 children graph,therefore

NEK(G1,G2) $\cap$PK(G1,G2) $\cap$GK(G1,G2)=0

G1=(5,2) element graph of 3 children graph,therefore

NEK(G1,G2) $\cap$PK(G1,G2) $\cap$GK(G1,G2)=0

G1=(5,3) element graph of 3 children graph,therefore

$$NEK(G1,G2) \cap PK(G1,G2) \cap GK(G1,G2)=1$$

Therfore MATCH FOUND with the family element (5,3).Hence the result graph extracted.

5.Results And Discussion

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,

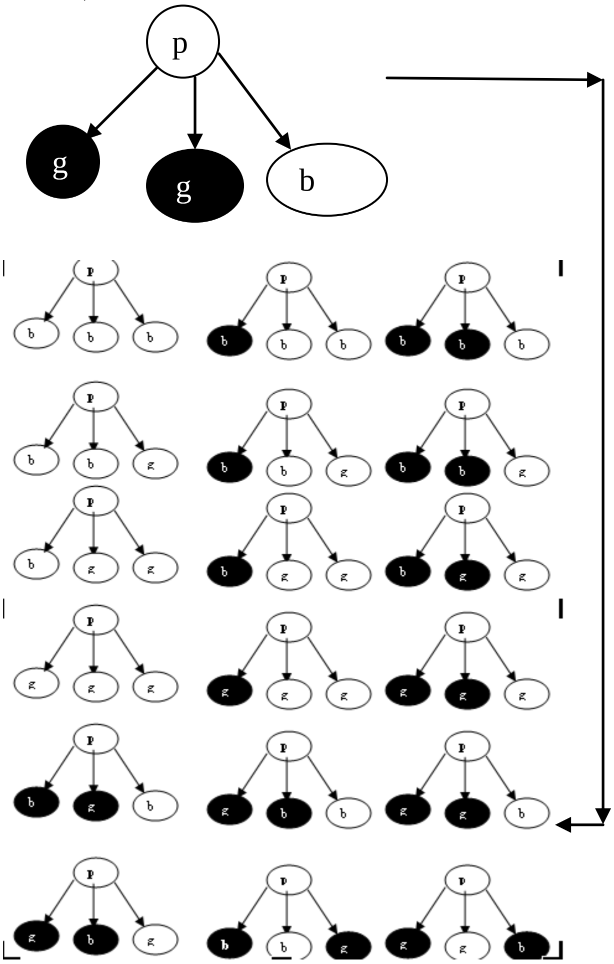


Figure 3: Sucessful implementation of Proposed Graph matching structure

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

6.Conclusion:

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 equilant 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.

References

[1] G. Di Fatta, S. Leue, E. Stegantova. “Discriminative Pattern Mining in Software Fault Detection.”Workshop on Software Quality Assurance, 2006.

[2] E. W. Dijkstra. “A note on two problems in connection with graphs. Numerische Mathematik” ,269-271.

[3] T. Falkowski, J. Bartelheimer, M. Spilopoulou. “Mining and Visualizing the Evolution of Subgroups in Social Networks”, ACM International Conferenceon Web Intelligence, 2006.

[4] M. Fiedler, C. Borgelt. “Support computation for mining frequent sub graphs in a single graph.” Workshop on Mining and Learning with Graphs(MLG’07), 2007.

[5] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Ma-honey. “Statistical properties of community structure in large social and information networks.” In WWW, pages 695-704, 2008.

[6] J. Feigenbaum, S. Kannan, A. McGregor, S. Suri, J. Zhang. “Graph Distances in the Data-Stream Model”. SIAM Journal on Computing, 38(5): pp.1709–1727, 2008.

[7] J. Ferlez, C. Faloutsos, J. Leskovec, D. Mladenic, M. Grobelnik. “Monitoring Network Evolution using MDL”. IEEE ICDE Conference, 2008.

[8] F. Eichinger, K. B-ohm, M. Huber. “Improved Software Fault Detection with Graph Mining”. Workshop on Mining and Learning with Graphs,2008.

[9] F. Eichinger, K. B-ohm, M. Huber. “Mining Edge-Weighted Call Graphs to Localize Software Bugs“. PKDD Conference, 2008.

[10] W. Fan, K. Zhang, H. Cheng, J. Gao. X. Yan, J. Han, P. S. Yu O. Verscheure.Direct “Mining of Discriminative and Essential Frequent Patterns via Model-based Search Tree“. ACM KDD Conference, 2008.

[11] C. Liu, F. Guo, and C. Faloutsos. Bbm: “Bayesian browsing model from petabyte-scale data. In KDD“, pages 537-546, 2009.

[12] Y. Low, J. Gonzalez, A. Kyrola, D. Bick son, C. Guestrin, and J. M. Heller stein. “Graph lab: A framework for parallel machine learning“. In UAI, pages 340-349, 2010.

[13]R. Gemulla, E. Nijkamp, P. Haas, and Y. Sisma-nis. “Large-scale matrix factorization with distributed stochastic gradient descent“. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 69-77. ACM, 2011.

[14] A. Ghoting, R. Krishnamurthy, E. P. D. Pednault,B. Reinwald, V. Sindhwani, S. Tatikonda, Y. Tian,and S. Vaithyanathan. “System: Declarative machine learning on map reduce“. In ICDE, pages 231-242, 2011

[15] U. Kang, H. Tong, J. Sun, C.-Y. Lin and C. Faloutsos. “Gbase: an ancient analysis platform for large graphs“.VLDB J., 21(5):637-650, 2012.