

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis addresses the problem of graph matching, which is the process of finding the similarity between the two graphs. Depending on the requirements of matching, graph matching is classified into two categories. When a strict matching between two graphs is needed, exact matching is used. Whereas inexact graph matching is used, when some flexibility and tolerance to error is allowed. Graph matching has a wide range of applications in structural pattern recognition, biometric identification, computer vision, biological and chemical applications, etc. The main advantage of using graphs as compared to vectors in object recognition is its representational power and flexibility to link to any number of objects. However, with the advantage of using graphs, it brings some limitations. For example, one of the key issues of graph matching techniques is that they are computationally much expensive. Due to the non-availability of an efficient solution, various approximation and suboptimal algorithms have been proposed. Another key issue in graph matching is the lack of standard similarity measures to find similarity or dissimilarity between two graphs. This thesis considers the above key issues and presents algorithms for exact, approximate and error-tolerant graph matching.

An extensive survey of various exact and inexact graph matching techniques was provided. Graph edit distance is one of the most flexible technique to perform graph matching. It is the minimum number of edit operations needed to transform one graph into another one. A

common set of edit operations includes insertion, deletion and substitution of nodes and edges. A variation to graph edit distance using the concept of graph homeomorphism is introduced. The homeomorphic graph edit distance between two graphs is equal to graph edit distance between two graphs after applying path contraction on both input graphs. The path contraction operation on a graph removes nodes of degree two of all simple paths of the graph such that all intermediate nodes except first and last is of degree two. It leads to a reduction in search space and substantial saving in computation time.

A class of graph matching algorithms is presented, which diminishes the graph size by deleting the less important nodes using some measure of relevance. Node contraction approach reduces the graph size by deleting the nodes with least degree provided these nodes are not cut vertex. The condition for not being cut vertex is put to avoid unrestricted deletion of nodes, which may lead to many disconnected components of the graph. The extended graph edit distance between two graphs is defined, which is equal to graph edit distance between two graphs after applying the appropriate node contraction operations on both input graphs. Computational results show that this technique can be used for a trade-off between execution time and classification accuracy.

One limitation of the above node contraction approach is that the number of nodes of a given degree in different graphs can differ from each other, so the number of nodes removed may not be uniform. An extension to the above approach is proposed, which reduces the graph size by removing a fraction of nodes from both graphs based on a given centrality measure. Four different centrality measures, namely, degree, betweenness, eigenvector and PageRank centrality measures are used to contract the graphs. Experiments show that different centrality criteria lead to a different saving in computation time and classification ratio. Depending on the application requirements, suitable centrality measure can be selected to achieve the best performance. One advantage of this proposed method is that it can be used to perform graph matching under given time constraints. For large graphs, an early estimate of graph matching can be obtained based on a fixed fraction of nodes of both graphs, which are most important according to a given centrality measure. One can note that this approach, along with the above approaches of node contraction and homeomorphism, can be used on top of any graph matching technique as an optimization criterion.

A novel metric for measuring the similarity between two geometric graphs is introduced. The proposed similarity measure is promising yet straightforward. Vertex distance between two geometric graphs is defined as a linear sum assignment problem formulation for the assignment of the vertex set of the first graph to the vertex set of the second graph so that their overall distance is minimized. Edge distance between two graphs is defined using three feature of edges, which are orientation, length and position of edges. Finally, the graph distance metric is defined by combining the vertex distance and the three components of edge distance. The proposed graph distance can be considered as a step towards introducing a standard similarity measure for finding similarity between two graphs embedded in a plane.

The proposed geometric graph similarity framework is applied to exact and error-tolerant graph matching. An algorithm for geometric graph isomorphism is presented, which checks whether two graphs are geometrically isomorphic. In case there is a chance of slight distortion in the coordinates of vertices, this algorithm can also check that the two graphs are t -tolerant isomorphic when the distance between two corresponding coordinates is within the distance t . Before checking for geometric graph isomorphism, graph alignment algorithm is used to ensure that the reference coordinates of two graphs as identical as possible. The proposed similarity scheme is also used for error-tolerant graph matching. A set of weight parameters are used to combine the vertex distance and the angular, length and position components of edge distance. This framework is particularly useful for application involving large graphs, where data can be altered by distortions. Depending on the application need, suitable weight factors can be selected to optimize the performance. Computational results show that this framework is promising to graph dataset in which each vertex has its associated coordinate point.

6.2 Future Directions

Graph matching using node contraction is based on the concept of ignoring nodes having less importance, so as reduce the total search space. One of the future work can be to explore the performance of removing edges rather than nodes during the preprocessing of the graph using some relevant criteria. The proposed work of error-tolerant graph matching using centrality information considers four centrality measures. Future work can be to

consider evaluating the performance of other centrality measures on the different dataset. The experimental results are performed mainly on letter and molecules graph dataset; future work can be to evaluate the performance of the algorithms on the other graph dataset, especially chemical and biological graph dataset. Another direction can be to implement the parallel version of the algorithm and compare its performance with the classical one. In the proposed work of error-tolerant geometric graph matching framework, before the computation of vertex distance, size of both graphs are made equal by inserting nodes in the smaller graph with coordinate equal to the average of all its existing coordinates. Other techniques can be explored to insert the nodes so that they are more representative concerning initial graphs. Similarly, other approaches can be investigated to add the edges in one graph to make the edge size of both graphs equal. Weight parameters are used to combine the vertex distance and the other three components of edge distance. Future work can be to perform experiments for the different combination of weight parameters and compare their performance. Future work can also be to apply the proposed geometric graph similarity framework for the design of graph kernel and embedding.