

## Comments from Committee Members and Updates to the Thesis

### Comments from Dr. Kalayanaraman

#### 1. Can any form of transitive relationship between subgraphs reveal information about the correlative relationship of the subgraphs, which may result in addition pruning opportunities?

*This possibility has been included in Section 3.4.*

*[...There is an inherent relationship between the pairwise correlations of subgraph features. For example given three different subgraph features  $g_i, g_j$  and  $g_k$  we have three different pairwise correlations  $\rho_{g_i, g_j}$ ,  $\rho_{g_j, g_k}$  and  $\rho_{g_i, g_k}$ . If we can prove that these pairwise correlations are transitive it will not be necessary to check the  $\beta$  constraint for the new subgraph feature under consideration against every subgraph feature already included  $H$  (step 12 in the expand procedure). As the size of  $H$  grows checking the  $\beta$  constraint can become quite expensive and the transitivity between pairwise correlations (if true) can be exploited to address this issue....]*

#### 2. Another possible measure of performance is the number of features per unit time that can be considered by gSpan (or any graph miner) with and without pruning. Such a measure is related more to how well pruning is improving the regression result per unit time spent looking for good sets of features. This is similar to recent alternatives to traditional high-performance computing metrics like FLOPS (e.g., QUIPS Quality Improvements Per Second).

*This has been included in Section 3.4.*

*[... While comparing the performance improvement due to the pruning mechanisms we compared gSpan with the pruning mechanisms against gSpan. The experimentation focused on maximizing the predictive accuracy achieved by features generated by each system and then comparing the computational resources consumed by each system (runtime). The intuition behind this experimentation is to compare the computational cost of the best possible predictive model produced by each system. An alternative evaluation mechanism would be to compare the number of subgraph features produced by each system per unit computational resource. This would be similar to the notion of measuring floating point operations per second as in the case of high performance computing. Such an evaluation mechanism based on subgraph features produced per unit computational time does give an objective measure of the speedup caused by the pruning mechanisms but raises the question as to how good the produced subgraph features are with respect to predictive accuracy of the model. Although this is a weakness, this could be a valid evaluation mechanism...]*

#### 3. Explain why gSpan+pruning actually took longer than just gSpan on the Huuskonen dataset.

*This has been discussed in Section 3.4.*

*[... The testing for the  $\beta$  constraint (line 12 in the Expand procedure) which involves computing and checking pairwise correlations for the subgraph features under consideration against all subgraph features included in  $H$ , can be a significant overhead. This is quite apparent in the Huuskonen dataset. While the number of subgraphs considered both by gSpan and gSpan with pruning mechanisms is the same, gSpan with pruning mechanisms consumes more time than gSpan. This discrepancy can be attributed to the overhead of computing the  $\beta$  constraint. In certain cases, as in the Huuskonen dataset, this can lead to a worse performance as compared to running gSpan without the pruning mechanisms. ...]*

**4. Add a discussion about constrained graph classes and the possibility that they may not benefit from your pruning mechanism. Ideally, you can say something about a few specific classes.**

*This has been included in Section 3.4.*

*[... Certain application domains may not require unconstrained graphs to fully capture the relevant data. Constrained cases such as trees or planar graphs might suffice. This raises the important question on the relevance of our pruning mechanisms in such cases. It must be pointed out that when the subgraph isomorphism test is not expensive (as in the case of trees or planar graphs) the pruning mechanisms may not give a speedup. It is also possible that the extra overhead of computing the  $\beta$  constraint might make the pruning mechanisms expensive. ...]*

**5. Add a discussion about how the inclusion of structural features in the regression process can help a domain expert (e.g., biologist).**

*This has been included in Section 4.2*

*[ ...In addition to improving the predictive accuracy of the regression model the incorporation of structural features can further the understanding of the underlying problem. For example, correlations between attribute valued features and structural features can lead to a better understanding of the attribute valued feature or the correlations between structural features can raise important questions about the properties of the domain. In general, structural information represents a relevant body of features and incorporating them into model building can lead to better models as well as a better understanding of the domain. ...]*

## Comments from Dr. Cook

1. In looking at your dissertation, it appears to be organized as a series of incremental contributions and individual experiment ideas. I don't see a central hypothesis that you are putting forward and validating. What is this hypothesis? Can you clearly state this in the Introduction and show throughout the dissertation how you are validating the hypothesis?

*I have updated the introduction to include a clear statement of the thesis, which is as follows.*

***“The state of the art in graph classification and regression algorithms can be improved both in terms of runtime and predictive accuracy by understanding and addressing their weaknesses.”***

*I also discuss in Section 1.4 how each chapter contributes to validating the thesis. The following table (also in Section 1.4) presents an overview of how each chapter (except the introduction and related work) contributes to this validation.*

Chapter	Empirical Observations	Theoretical Results	Contribution towards validating the thesis
Chapter 3	Massive redundancy in search for subgraph features.	Developed pruning mechanisms in the search for subgraph features.	Improvement in runtime.
Chapter 4	1) Need for combination of linear models in certain domains. 2) Including structure in models can improve predictive accuracy.	Developed gRegress, an algorithm that induces a tree based combination of linear models.	Improvement in predictive accuracy.
Chapter 5	1) Differences in the behavior of graph classification algorithms despite similar performance. 2) Walk-based Graph Kernels cannot capture structure.		Motivation behind thesis.
Chapter 6		Language of Walks in a graph is a Regular Language.	Important development in the direction of improving runtime.

**2. DT-GBI is another system which combines substructure discovery with decision trees. How does your approach compare? Why did you choose a decision tree and not another classifier with binary attributes? The equation for step 1 of step 4.1, need to define the terms here.**

*I have included this in section 4.1.*

*[...It is important to distinguish gRegress from the previously discussed DT-CLGBI. While both of these approaches are based on decision trees, there are some important differences. First, DT-CLGBI produces a pure decision tree where each node represents a subgraph presence/absence test while gRegress is a combination of linear models combined with a decision tree. Leaf nodes in DT-CLGBI are pure nodes (belonging to a single class) while leaf nodes in gRegress are linear models. Second, the splitting tests in DT-CLGBI are based on entropy measure with respect to binary classification while those in gRegress are based on an error measure based on standard deviation with respect to regression. In the case where gRegress is run with  $L = 1$  (number of examples at leaf) no linear model at the leaf is necessary (or possible) and induced tree is quite similar although there might be a difference due to the difference in the splitting condition. The key difference between DT-CLGBI and gRegress is that DT-CLGBI produces a decision tree of subgraph presence/absence tests while gRegress produces a combination of linear models organized as a tree. ...]*

**3. Your statement "An example of this would be the following hypothetical problem from chemistry. Note that this hypothetical problem is for the purpose of describing our intuition only and may be completely inaccurate as far as chemistry is concerned" is fairly weak for a PhD dissertation. Use an example that you know to be accurate here.**

*I did some reading on organic chemistry, the boiling point is in fact largely dependent on functional groups. So, this is a valid example. I have cited a reference.*