NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Assignment 2 : Final Report

CZ4071 – NETWORK SCIENCE
(Semester 2, AY 2019/2020)

Paper Reviewed in this report:
[1905.13732] End to end learning and optimization on graphs

Team 3:

WANG ZIYAN	U1822682G
WILSON THURMAN TENG	U1820540H
CAI LINGZHI	U1622184H
LUO JINQI	U1722733J
TEO BOON SHUAN	U1821709J
LIU FANGBING	U1622143H

Final Report:
End-to-End Learning and Optimization on Graphs

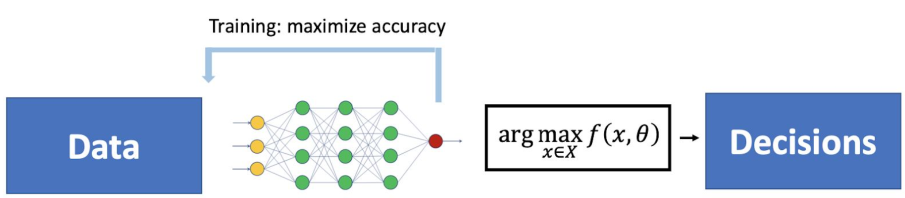
1. Introduction

Nowadays, the research on Graph Optimization (GOpt) problems has become increasingly popular. Researchers are expanding knowledge boundaries of this area under influence of multiple factors such as the abundance of open-source network datasets, the emergence of deep learning, and the obvious limitation of existing graph solutions. Empirically speaking, real practices of GOpt problems are not purely machine learning problems nor simply traditional optimization problems. It is becoming a complex subject, which combines statistical learning techniques and efficient algorithms for optimization processes. Since GOpt research has a variety of real-world applications like Cluster Classification, modern approaches to GOpt problems would like to treat predictive models training and graph optimization separately. In this paper, the author proposes an alternative method where the traditional training process of deep learning is integrated with a new differentiable layer, representing the ultimate goal of a given graph optimization problem.

2. Problem Description

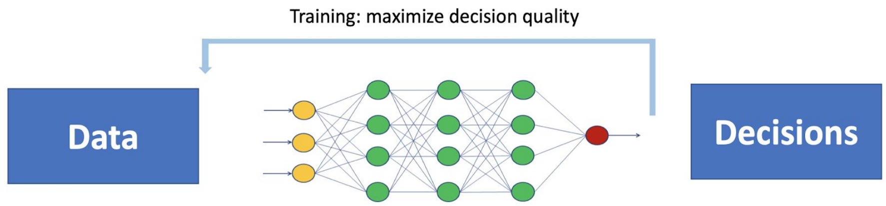
Current machine learning solutions of real-world GOpt problems are insufficient, since they fail to utilize node attributes or linked edges data which are partially-observed in nature. Towards solutions of better efficiency or accuracy, researchers propose solutions in two directions that are different in the number of processing stages.

The first approach implements two stages (Fig. 2.1). In the first stage, a predictive model setting with standard loss will be trained. In the second stage, the posterior generated will then be fed to finish the optimization process. However, the problem with this approach is an oversimplified standardized loss function which leads to poor performance when dealing with complex optimization problems in particular.



(Fig. 2.1) Two-stage Approach

A preferable alternative would be the end-to-end approach (Fig. 2.2). In this approach, the solution of the specific optimization problem is entirely dependent on the parameters obtained from training deep learning models to develop a smaller-scale representation of the problem (e.g. Policy model of Reinforcement-Learning). However, training this representation is time-intensive and disregards additional prior knowledge which may lead to unsatisfactory performance. This problem is further exacerbated if data is lacking.



(Fig. 2.2) End-to-end Approach

In this paper [1], the author investigated the shortcomings of the aforementioned approaches and proposed a new solution on ClusterNet. ClusterNet has a trainable (differentiable) layer for the solver of a simpler problem that is embedded into the neural network. By obtaining the parameters of this layer in the training process, we can use the model to generate the mapping function of the Problem of Interest (e.g. Link Prediction).

Specifically, the author formalized the optimization problem as:

$$\max_{x \in \mathcal{X}} f(x, A)$$

(Eqn. 1) Graph Optimization

where A denotes the adjacency matrix of the partially-observed graph *G=(V, E)* and *x* denotes the final decision of our targeted variable. Traditionally, to solve this optimization problem, the completed *A* is obtained by the predictive model learnt from the loss *L(A, A_train)*, where *A_train* is defined as the adjacency matrix with only the training edges (partial edges). Afterwhich, the optimal *x* can be calculated using *Eqn.1*. in the optimization process.

However in this paper, the author proposed an end-to-end model that directly maps *A_train* to a feasible decision *x*. That is, the trained model directly solves the optimal *x* from the optimization function *f(x,A_train)*. Intuitively, we observe that *f(x,A_train)* represents the quality of the decision made from evaluating the training data, while the original loss *L(A, A_train)* only measures the predictive accuracy.

3. Main Contributions

First, the author presents a general framework for integrating graph learning and optimization. The framework utilizes a simpler problem of optimization in continuous space as a substitute for the more complicated, discrete problem. Unlike the end-to-end approach, this framework is able to simplify the optimization component instead of updating the entire algorithm.

Second, the author introduced the idea of simplifying graph optimization problems into specific clustering problems, in which the clustering layer is differentiable. This allows the solution to be applied in deep learning systems. In this paper, the author demonstrated this approach to interpret the cluster assignments as a solution to the discrete problem. Specifically, this idea is helpful for two classes of optimization problems: Graph partitioning (e.g., community detection or maxcut) and Subset Selection Amongst K Nodes (e.g., facility location, influence maximization, immunization, etc).

Third, the author shows experimental improvements over both two-stage baselines as well as alternate end-to-end approaches over a range of open-source research datasets. The author compares ClusterNet with other methods from two aspects, namely Results on Single Graphs (RSG) and Generalizing Across Graphs (GAG). The RSG explores how the model performs on combined link prediction and corresponding optimization problems. (Fig 3.1) shows the objective value obtained by each method on the full graph for community detection, with (Fig 3.2) showing the results of facility location. The GAG experiment investigates the ability of ClusterNet to generalise its learning across similar optimization problems from different domains. (Fig 3.3) shows the results of ClusterNet's performance compared to other baseline models for generalized strategy learning. We can see from the author's experiment that ClusterNet outperforms most of the baseline methods on learning+optimization problems. (Tables below are obtained from the original paper)

	Learning + optimization				
	cora	cite.	prot.	adol	fb
ClusterNet	**0.54**	**0.55**	**0.29**	**0.49**	**0.30**
GCN-e2e	0.16	0.02	0.13	0.12	0.13
Train-CNM	0.20	0.42	0.09	0.01	0.14
Train-Newman	0.09	0.15	0.15	0.15	0.08
Train-SC	0.03	0.02	0.03	0.23	0.19
GCN-2stage-CNM	0.17	0.21	0.18	0.28	0.13
GCN-2stage-Newman	0.00	0.00	0.00	0.14	0.02
GCN-2stage-SC	0.14	0.16	0.04	0.31	0.25

	Learning + optimization				
	cora	cite.	prot.	adol	fb
ClusterNet	**10**	**14**	**6**	**6**	**4**
GCN-e2e	12	15	8	**6**	5
Train-greedy	14	16	8	8	6
Train-gonzalez	12	17	8	**6**	6
GCN-2Stage-greedy	14	17	8	7	6
GCN-2Stage-gonzalez	13	17	8	**6**	6

Fig 3.1 & Fig 3.2 Experiment Results from the Original Paper

Community detection					Facility location				
	synthetic		pubmed			synthetic		pubmed	
No finetune	Avg.	%	Avg.	%	No finetune	Avg.	%	Avg.	%
ClusterNet	**0.57**	**26/30**	**0.30**	**7/8**	ClusterNet	**7.90**	**25/30**	7.88	3/8
GCN-e2e	0.26	0/30	0.01	0/8	GCN-e2e	8.63	11/30	8.62	1/8
Train-CNM	0.14	0/30	0.16	1/8	Train-greedy	14.00	0/30	9.50	1/8
Train-Newman	0.24	0/30	0.17	0/8	Train-gonzalez	10.30	2/30	9.38	1/8
Train-SC	0.16	0/30	0.04	0/8	2Stage-greedy	9.60	3/30	10.00	0/8
2Stage-CNM	0.51	0/30	0.24	0/8	2Stage-gonz.	10.00	2/30	**6.88**	**5/8**
2Stage-Newman	0.01	0/30	0.01	0/8	ClstrNet-1train	7.93	12/30	7.88	2/8
2Stage-SC	0.52	4/30	0.15	0/8					
ClstrNet-1train	0.55	0/30	0.25	0/8					
Finetune					Finetune				
ClstrNet-ft	0.60	20/30	0.40	2/8	ClstrNet-ft	8.08	12/30	8.01	3/8
ClstrNet-ft-only	0.60	10/30	0.42	6/8	ClstrNet-ft-only	7.84	16/30	7.76	4/8

Fig 3.3 Comparison of Performance on Different Tasks from the Original Paper

4. Dataset Analysis

The dataset we will use in our re-implementation experiment is WebKB which contains 877 scientific publications and 1608 links in total. The publication information is gathered from webpages of four universities, which are Cornell University, University of Texas, University of Wisconsin and University of Washington.

There are two files corresponding to each university which describes the citation network. The first one *(.cites)* describes the links among the publications. If one paper is cited by another paper, then there will be a citation link between these two publications. Another file *(.content)* contains the descriptions of the webpages. After stemming and removing the stopwords in the webpages, a vocabulary size of 1703 unique words remained. Each publication in the file is described by binary values which indicate whether each word in the vocabulary is present (indicated by 1) or absent (indicated by 0) in the webpage.

In this experiment, we will train the clusterNet and the other three comparison models on this dataset and compare the performance of each model on the community detection, where modularity is used to reflect the performance.

5. Models Evaluation

In the experiment, we developed four models. ClusterNet model is the approach proposed in this paper and the other three are built up for comparison.

I. ClusterNet Model

<u>Graph embedding layer:</u>
This layer takes edges, Atrain(.cites) and node features (.content) as input and embed the nodes of the graph into matrix. In our experiment, we use GCN to implement this. By training this layer, the model could be tuned to fit specific problems. In this way, the clustering process in the next step could be more objective and efficient for the following optimization task.

<u>Differentiable optimization layer:</u>
This layer takes continuous-space embeddings as input, and outputs the soft K-means clustering assignment of each node.

II. GCN End to End

This method is also an end-to-end approach, but it lacks an explicit algorithm. A GCN-based model is trained to directly output final communities and facilities from the graph.

III. GCN Two Stage

This method firstly trains a model to predict links, where the GCN-based system [2] is applied here. Then run the optimization algorithm on the predicted graph.

IV. No Prediction Training

This method skips the link prediction step and directly runs an optimization algorithm, agglomerative clustering, on the partially observed graph.

6. Experiment Result

In the experiment, we use the same coefficients as the reference paper. We remain 40% of edges as training data and use K = 5 clusters.

I. Modularity

	Cornell	Texas	Washington	Wisconsin
ClusterNet	**0.37824**	**0.2408**	**0.2798**	**0.3838**
GCN-2Stage	0.2417	0.2219	0.2105	0.2960
GCN-e2e	0.0907	0.0152	0.0617	0.0334
No prediction	0.0726	0.0475	0.0938	0.0801

II. Time(s)

	Cornell	Texas	Washington	Wisconsin
ClusterNet	4.815	5.08	5.457	5.541
GCN-2Stage	11.840	13.042	17.234	18.406
GCN-e2e	**4.124**	**4.870**	**4.882**	**4.600**
No prediction	0.0190	0.025	0.034	0.037

When measured by modularity, the ClusterNet Model outperforms the other models to a large extent on the WebKB dataset. Although it does not guarantee less computing time than compared models, if we take both accuracy and running time into consideration, the ClusterNet Model is an efficient method which could achieve higher performance with a relatively acceptable processing time.

7. Connection with CZ/CE4071

The main Network Science problem this paper's algorithm serves to solve is the GOpt problem. GOpt problems can be interpreted as variations of the cluster assignment problem.

For example, in community detection we can interpret the cluster assignment problem as assigning the nodes to communities. In maxcut, we can use two clusters to assign nodes to either side of the cut. In maximum coverage and related problems, where we attempt to select a set of K nodes which covers as many neighbouring nodes as possible. This problem can be approximated by clustering the nodes into K components and choosing nodes whose embedding is close to the center of each cluster. However, please note that while these problems are not entirely reducible to K means, using K means as a layer in the network provides a useful inductive bias. We would also like to mention that the paper solves a graph partitioning problem. Graph embedding using the GCN allows for a flexible format to perform graph partitioning.

However, the technique discussed in the paper has some limitations. Firstly, it is not general enough to deal with graphs which possess node, edge or attribute uncertainties. With the rise in uncertain graph datasets, we are often interested in the probabilities that a certain graph property holds, therefore, thresholding or considering the probability as weights may not be appropriate. To solve this problem, we can implement a monte carlo sampling algorithm as the first layer to estimate the reliability of the links and only keep links with reliability above a certain threshold value. To improve this idea further and integrate this into the technique, we can make the monte carlo sampling layer differentiable with the threshold value as the loss function to minimize. Secondly, the technique relies solely on graph embedding which takes into account the distance between nodes, and disregards feature extraction. Although the technique shows promising results that rival state-of-art models, we argue that performance could be improved further if we were to incorporate results from feature extraction as well.

8. Potential Real-world Applications

Intuitively, many GOpt problems can be interpreted as a variation of the cluster assignment problem. There are plenty of real-world business applications for clustering a graph and finding the central nodes. For example, Pupu Mall, a start-up company in China, runs its online supermarket with a 30-minute delivery guarantee to anywhere in the city. The company arranges its warehouses around 5 to 10 kilometers apart, and hires delivery men at these warehouses to cover a particular geographical segment of the city. Before expanding its services to a new city, the company models the traffic network of the city into a graph. The nodes represent the buildings, while the distance between any two nodes represent the time required for the delivery men to travel between the buildings. Considering the limited funding that start-ups possess, it is crucial for Pupu mall to be able to experiment with layouts of where to place their initial batch of warehouses to minimise the facility's distance to all other nodes in the graph.

9. Conclusion

In our report, we have presented ClusterNet, an end-to-end neural network architecture that merges a differentiable graph embedding layer with an optimization layer (e.g. K-means clustering layer). The graph embedding layer simplifies the problem while the optimization layer solves the problem. This allows the quality of the solution to be the measure (loss) of how the graph embedding layer simplifies the problem and hence, the simplified representation of the graph would be optimal for the problem. With a tradeoff of incremental training time, this combined approach (2-stage + end-to-end) reinforces the robustness of the network to different optimisation problems, and also enhances its performance. While there are some limitations to this network, like its inability to work with uncertain graphs, the approach employed in this paper formalises a framework for future work. Our experimental results show that ClusterNet has state-of-art performance on modularity and has the potential power of generalising to different datasets. While there are many future directions of research for these approaches, we feel that this paper adds another important primitive to our ever-evolving problem-solving toolbox. We are excited for these possibilities.

10. Reference and Links

In this section, we list relevant documents and papers for this course project.

I. **Target Paper**

[1] B. Wilder, E. Ewing, B. Dilkina, and M. Tambe, End to End Learning and Optimization on Graphs. In NeurIPS, 2019. [ArXiv Link]

II. **Other Relevant Papers**

[2] Thomas N. Kipf and Max Welling. Semi-supervised Classification with Graph Convolutional networks. In *ICLR*, 2017. [ArXiv Link]

III. **Course Project Document Deliverables**

Presentation Slides: [Google Drive Link]
Mid-Progress Report: [Google Drive Link]
Github Repository: [Github Link]