Semi-supervised Learning on Hypergraphs: Comparing and Exploring

Interim Report

Chen Jiali 3035085695

Zhang Ying 3035084366

(in alphabet order)

Date of Submission: 22/1/2016

Abstract

Graph based learning is one of the most significant fields in semi-supervised learning. The existing methods mainly focus on objects with pairwise relationships, which can be illustrated as normal graphs. However, relationships among objects are always too complex for normal graphs to summarize. Based on the fact, hypergraph learning methods have become increasingly significant. Among the related methods, Hubert Chan provides us an effective approach based on directed hypergraph which is easier to understand compared to Hein's method, but whether it can provide a result of higher accuracy, less time and space complexity in multiple cases remains to be examined.The purpose of this project is to verify the above questions by implementing and comparing these two existing methods. Finally, an improved method will be proposed in both programming and mathematical ways. Currently, a number of previous papers related to these two methods have been studied, datasets have been selected for both examining the consistency of the implementation as well as testing. Besides, preliminary version of Python implementation of both methods have been completed. Moreover, a detailed experiment procedure has been designed and a number of comparison results has been plotted through experiments. In the next semester, research of our own approach will be pursued.

II.Acknowledgment

We would like to express our deepest appreciation to all those who provided me the possibility to complete this report. A special gratitude we give to our supervisor Dr Chan, Hubert Chan and supervisor helper Mr Zhang, Zhang Chenzi. As we have very limited knowledge about machine learning, their suggestions and instructions provided us a number of shortcuts, helped us to complete this report.

Furthermore I would also like to acknowledge with much appreciation the crucial role of the staff of Computer Science Department HKU, who gave the permission to use all required equipment and the necessary materials to complete the tasks. Last but not least, many thanks go to our CAES class instructor, Ken Ho. I have to appreciate the guidance given by him especially in our project presentation as well as report writing that has improved our skills.

Thanks and appreciations go to all of the individuals who has given precious comments and suggestions in developing the project.

III.Table of Contents

IV.List of Figures

V.List Of Tables

VI.Abbreviations

- RBF: Radial basis function, a popular kernel function used in various kernelized learning algorithms

- SVM: Support vector machine, is supervised learning model with associated learning algorithms that analyze data used for classification and regression analysis.

- UCI: University of California Irvine

1.Introduction

Machine learning, theoretically speaking, is the process to give a computer the ability to imitate the learning behavior of human to acquire new knowledges or skills. Within recent decades, the topic of machine learning has attracted increasing attention. Because of the tremendous improvement of machines' computing ability, this technology experiences a rapid development. Examples of utilization can be found in various fields such as computer vision, search engine, computer security as well as daily issues like weather forecasting.

Semi-supervised learning is one type of machine learning skills with high learning accuracy and efficiency. Compared with supervised and unsupervised learning, this training process makes use of both labeled and unlabelled data and provides a better performance in a great many realistic problems. There have been a number of semi-supervised learning methods proposed by scholars already, such as Expectation Maximization, transductive support vector machines and so on. Among them, graph-based method plays a very significant role.

Since learning on normal graphs are relatively well-researched, hypergraph is the focus of this project, which is an extension of normal graph, so that more information can be included, connected and utilized. In general, two existing hypergraph based semi-supervised learning methods are the candidates to implement, analyze and compare in the first phase. In the next phase, our own method will be proposed. Implementation, analysis, comparison of this method with the previous two methods will also be conducted.

2.Literature Review

2.1 Semi-supervised Learning

Based on supervised and unsupervised learning methods, semi-supervised learning produces an improvement by combining both of them. Training with a large size of labeled data, supervised learning is accurate but time-consuming, expensive and annotation demanding, while unsupervised learning is just the opposite. However, in real life examples, the amount of unlabeled data is far more easily-accessible than labeled data. With the approach introduced in Figure 1, the training can be conducted with a small labeled dataset and a large unlabeled dataset combinedly. Accordingly, semi-supervised learning can give a sufficiently high accuracy with a small labeled dataset and, hence, higher efficiency can be obtained[5]. Moreover, since labeling data is usually expensive, time-consuming and highly demanding(e.g. Labeling species according to DNA sequence), semi-supervised learning has the capability of reducing human effort and cost.

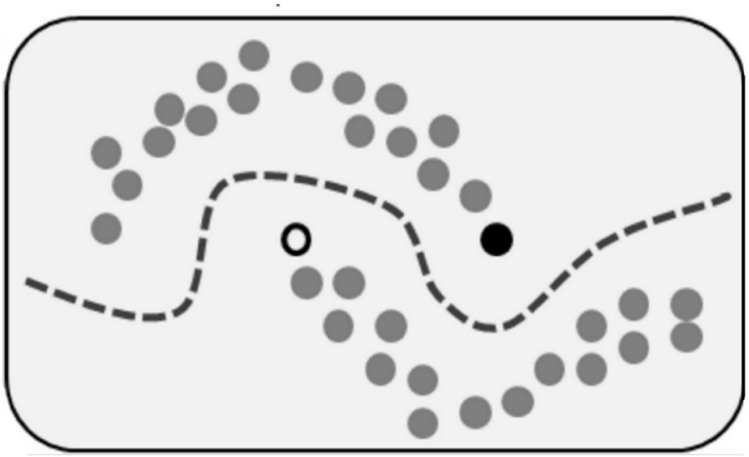


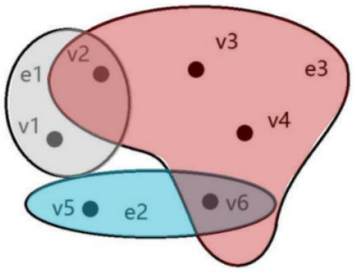
Figure 1: The learning scenario of semi-supervised learning algorithms. Similar data points are clustered into the same group, then groups are labeled according to the labeled data in this group. This scenario can provide a solution with high accuracy using a small training set.

2.2 Hypergraph v.s. Normal Graph

Traditionally, relationships among objects are assumed to be pairwise, thus normal graphs are often used to model problems. However, in many real world cases, relations may be much more complex so that modeling with normal graphs may degenerate them. Hence, hypergraphs are introduced for modeling. As illustrated in Figure 2, the complicated relationships between mushrooms are expressed concisely with hypergraph modeling. To be specific, which of the mushrooms share which choice of which feature is definite. In contrast, information is lost in normal graph since an edge between two vertices only implies that these two mushrooms share one same feature but not what exactly the feature is, while hypergraph can provide detailed

illustration[2]. Therefore learning on hypergraph can provide solutions to the cases with much more complicated relations.

Mushroom	Color	Habitat
1	grey	Brazil
2	grey	China
3	brown	China
4	brown	China
5	blue	Brazil
6	blue	China



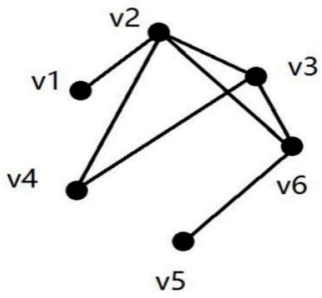


Figure 2: Representing 6 instances of mushrooms with 2 features(right) using both hypergraph(middle) and normal graph(left). Left: In hypergraph modelling, a hypergraph edge is a representation of a particular choice in a feature (e.g. grey, brown or blue in color feature). In this case 3 hyperedges are picked to illustrate. e1 represents grey mushrooms, e2 represents brown mushrooms while e3 represents mushrooms sharing China as their habitat. Right: In normal graph, an edge represents the existence of same choice of some feature between two mushrooms (e.g. edge between v1, v2 indicates that v1, v2 are both grey for color feature).

2.3 Apply Hypergraph to Semi-supervised Learning

As hypergraph is more powerful and can deliver a more explicit modeling of cases, the combination of its theories and semi-supervised learning algorithms is capable of broadening the application field of semi-supervised learning in real life. Hence, it is rather popular in recent years to research on different mathematical approaches to such combination and improve the efficiency of the implementation. For instance, D. Zhou, J. Huang, and B. Scholköpf introduced a method viewing the hyperedge as a clique (fully connected normal graph, explained in Figure 3 below (left)) of all the vertices and generalized clustering, classification and embedding to hypergraphs accordingly.[3]

2.4 True Hypergraph Approach (Hein's method)

M.Hein, S.Setzer, L.Jost and S.S.Rangapuram delivered an improvement motivated from the high time complexity of clique approximation. The team overcame the limitation by utilizing a family of regularization functions based on the total variation on hypergraphs and a balanced graph cutting method to avoid the time consuming normal-graph-clique construction process, as illustrated by Figure 3. Hence, they succeeded in the discovery of a purely hypergraph structural approach as well as decreasing the time complexity of the learning process[3].

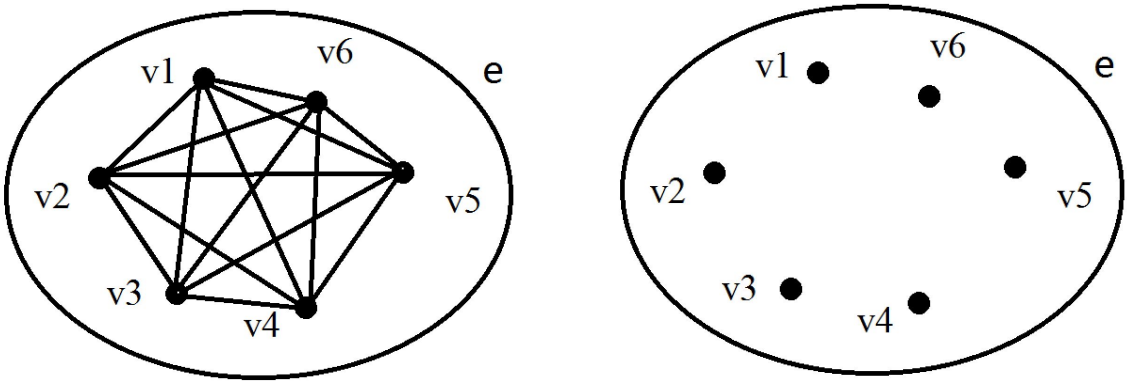


Figure 3: Clique Approximation(left) v.s. True Hypergraph(right). Left: In the clique approximation, a clique is constructed on each hypergraph to figure out a minimum graph cut. All constructed normal edges will participate in the computation. Therefore, in the case of an edge containing n vertices(6 in this example), complexity raise from n to n^2-n(30 in this example). Right: In Hein's method, no clique construction and the original hypergraph will be the only participant in finding the balanced cut. Hence, complexity of n(6 in this example) remains.

2.5 Subgradient Approach and Directed Hypergraph (Hubert's method)

Hubert Chan and his team, inspired from diffusion process on hypergraph, constructed a Markov operator that gives a subgradient of the solution in each iteration. Hence, although the objective function is not differentiable everywhere, optimal solutions can still be found. Besides, a framework on directed hypergraph was also delivered.

Directed hypergraph, as Figure 4 illustrated, is a modified version of normal hypergraph. Hyperedges are no longer directionless in a directed hypergraph, instead, a set of the vertices in one edge is defined to be head set while the other set of the vertices is tail set. With the assistance of directed hypergraph, more complicated relationships among vertices can be captured. For example, still take the mushrooms into consideration, it is reasonable to claim that mushrooms with identical cap-shape belong more likely to the same class. The directed hypergraph can help to capture such feature. As is in Figure 4 , hypothesis can be captured that the class of mushroom is unlikely to switch from "poisonous" to "edible", along the direction of e3.

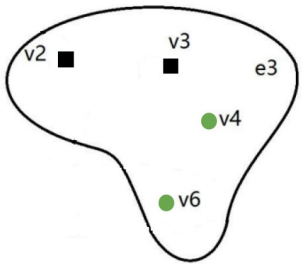


Figure 4: Directed hyperedge. A hyperedge modified from e3 in Figure 1(middle). Recall that e3 is the hyperedge representing set of all mushrooms live in China. Further design v2, v3 have round cap, v4, v6 have triangle cap, and accordingly v2, v3 are defined as heads and v4, v6 are defined as tails, denoted separately by square and circle. The hyperedge now has a direction from {v2, v3} to {v4, v6}.

3.Motivation

Both Hein's and Hubert's method have provided effective approaches to semi-supervised learning on hypergraph. However, currently there is no research analyzing the performance of the two methods and other popular approaches. It remains unknown how these two semi-supervised learning algorithms perform (time, space complexity & accuracy), in terms of being compared with each other. Meanwhile, performance under different typical machine learning conditions, e.g., large dataset with only a few learning data; small dataset with more classes to be examined and so on. In addition, we tend to explore whether the two methods are advanced and how significantly advanced they are, compared with the popular supervised learning approaches.

Moreover, as previously mentioned, hypergraph and semi-supervised learning provide effective methods in the machine learning field, and both Hein's method and Hubert's method are rather new among them. Therefore, if the implementation of our project can be conducted correctly and optimized sufficiently, it will be a contribution to this field. The programming language used for this project is Python, the reason will be discussed in section 4.

Additionally, since regularization in machine learning field is quite flexible. There exist a large number of alternatives for the activation functions, parameter choices, data pre-preprocessing choices and heads and tails choices. All these changes will result in different outcomes, and it is likely that one among them can achieve a better prediction.

4.Scope

4.1 Comparison Candidates

Our project only concentrates on the chosen two hypergraph based semi-supervised learning algorithms, which are Hein's method and Hubert's method. Hein's method was published in 2013 while Hubert's method has not been published yet, hence both methods are relatively new and worth further studying. Although no other machine learning algorithms are included as major research objects, a number of famous supervised learning algorithms such as kernel SVM and logistic regression may be included to make a comparison with the two methods we have chosen. What supervised learning methods to include will be discussed after completing the implementation of the two major methods. All the supervised learning methods can be directly called through Python Module APIs.

4.2 Programming Platform

Programming language Python is chosen as the programming language to implement and analyze the two methods. In the designing process, both Python and Matlab are popular in the machine learning field since both of them are very convenient in performing numerical computation and plotting. They both have a number of powerful supporting libraries like Statistics and Machine Learning Toolbox™ for Matlab, and SciPy, NumPy, sklearn for Python. Ultimately, Python is selected for the reason as follows:

a. Python is open source, which means it is free. Normal users as us are allowed to modify the source.
b. Python is more coder friendly. Python is well designed with useful functions are provided like map, concat etc., popular data structures as set, matrix, array and powerful build-in functions for them. These will simplify our code and shorten the coding time.
c. Python has powerful machine learning modules like sklearn, which contains a large number of written machines learning algorithms. Modules can be called within only a few lines.

4.3 Data Source

When conducting the implementation, datasets are selected from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml), which is a free online collection of datasets, whose home page is shown in Figure 5.

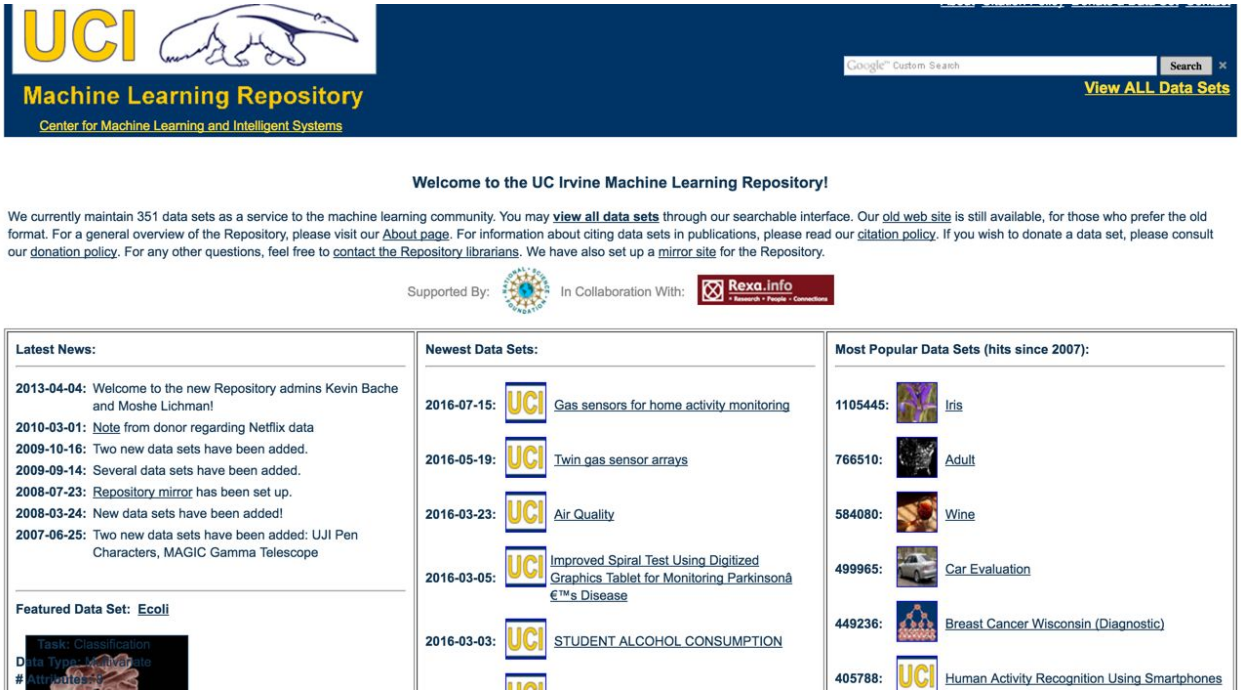


Figure 5: Homepage of UCI Machine Learning Repository. Abundant datasets of various types can be accessed freely from this site which is widely used in machine learning research field. All the dataset used by this project are retrieved from this repository.

This repository was created in 1987 by David Aha and fellow graduate students at UC Irvine. Since then, it has been widely used by students, educators, and researchers all over the world as a primary source of machine learning datasets. Furthermore, as most of the papers studied by our team during the literature review process adopted datasets from UCI Machine Learning Repository, we are quite familiar with this repository, which will make the testing experiment proceed more smoothly.

5.Experiment Setup

In this section, experiment setup will be discussed for the comparison and analysis of Hein's method and Hubert's method. Specifically, description of the data sets, data pre-processing, comparison methodology, experiment process will be introduced.

5.1 Data Description

All the datasets used by this project are retrieved from UCI machine learning repository for convenience (Reasons in detail have been explained in Section 5). The criteria of the data selection is whether the this dataset has complex relational features because our project aims to study hypergraph based algorithms. Compared with simple relational data, i.e. who has no more than two features, complex relational data has more. As is explained in section 2, hypergraph approaches has significant superiority in learning and predicting complex relational data.

Currently 3 datasets are selected and following is the description. More datasets will be probably included in the future research and implementation.

5.1.1 Mushroom Dataset

Complex relation dataset with ***abundant features:***
This dataset contains 8124 instances of gilled mushrooms in Agaricus and Lepiota. Each instance is described by 22 attributes, each attribute is a small number representing the category of that attribute. The data is accordingly classified as edible or poisonous.

5.1.2 Zoo Dataset

Complex relation dataset with ***small dataset size and more classes:***
Within the dataset, there are 101 instances. Instances contain 17 attributes like hair, feature, egg to help classify them into 7 classes.

5.1.3 Letter Recognition Dataset

Complex relation dataset with ***more classes and relatively less features:***
There are 2000 instances with 16 attributes like height, width, x variance, y variance, xy correlation etc. to describe how this letter is written and the classification is to identify which specific letter the instance is, which means there are 26 classes in total.

5.2 Data Preprocessing

5.2.1 Abandon Missing Data

The datasets selected from the UCI machine learning repository probably contain data with missing feature values. The reasons for the missing are data collection error or inapplicable measurements. For example, IP information of some visitors has a possibility to be lost in a visiting record of a website. In this case, different strategies will be adopted to eliminate this problem according to the characteristic of the dataset. Within the datasets selected, for dataset with more features than necessary like zoo dataset and mushroom dataset, feature with missing values will be dismissed since the remainings are sufficient for learning; for dataset with all features essential like letter recognition dataset, data points with missing values will be removed as removing some of the features will lead to suffering of information loss.

5.2.2 Training Data Selection

To pre-process the data, 20, 40, 60, 80, 100, 120, 140, 160, 180, 200 random data points are iteratively retrieved as training set (for the zoo set, 20, 25, 30, 35, 40, 45 instances are taken because of the small size of the dataset, and training set is supposed to be maintained as a small portion of the entire dataset). The main reason for such arrangement is to keep consistent with the previous research. In addition, using a small training set, as normally less than 3% of the entire set (as indicated by Figure 6), can effectively display the advantage of semi-supervised

learning.

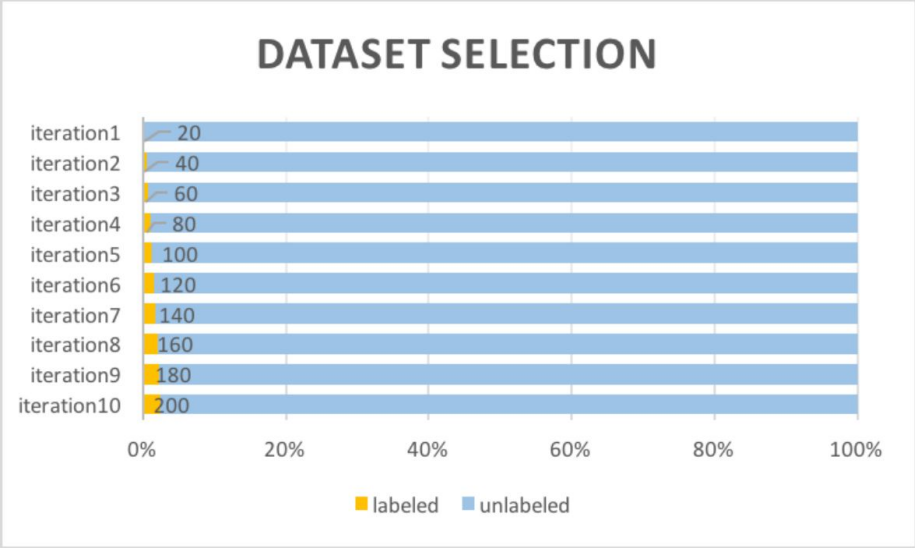


Figure 6: Illustration of dataset selection for mushroom dataset. For each iteration, increase the amount of labeled data by 20 in order to see the variation tendency of accuracy. Even though, the amount of labeled data is still a tiny part of the entire dataset.

5.3 Comparison Methodology

Comparison experiments are conducted using different sizes of training dataset, corresponding to the iterations in the section 5.2. For each iteration, accuracy of the two methods as well as other popular supervised learning approaches are computed. In order to make a thorough comparison, consideration are given not only to the two targets, but also to semi-supervised learning and supervised learning.

To obtain the “accuracy”, in each iteration, training approaches and the training set described above are used to train the learning parameters. The remaining data other than the chosen training set will be used to be predicted. The accuracy is defined as:

$$accuracy = \frac{correctPrediction}{testSize}$$

The process for the ten iterations will be conducted repeatedly, final result will be derived as an average of the repetitions. The reason for repeated experiment will be discussed in detail in section 8. A curve about the “accuracy”-“test set size / data set size proportion” will be plotted for each method, Figure 7 will be an example, in which mushroom is the dataset and 100 repetitions were conducted.

6.Project Procedure

In this section, a general timetable for this project will first be delivered. Moreover, three key phases will be discussed: implementation of two methods, comparison and analysis process and future plan for our own approach.

6.1 Timetable

Date	Procedure	Member in charge	Remark
Sept 30	Project Plan	Jiali, Chen Ying, Zhang	Completed
Sept 30	Project Website http://i.cs.hku.hk/fyp/2016/fyp16005/	Ying Zhang	Completed
Oct 30	Literature Review	Jiali,Chen Ying, Zhang	Completed
Nov 10	Experiment Setup & Data preprocessing	Jiali, Chen	Completed
Dec 02	Implementation of Hein's method (2 algorithms)		Completed
	Solution of prox problem	Ying, Zhang	Completed
	PDHG for ΩH,2	Jiali, Chen	Completed
Dec 15	Implementation of Hubert's method (3 algorithms)		Completed
	Markov Operator	Jiali, Chen	Completed
	Subgradient Method SGM	Jiali, Chen	Completed
	Semi-Supervised Learning	Ying, Zhang	Completed
Dec 20	Testing and plotting	Jiali, Chen Ying, Zhang	Completed
Jan 10	Optimization of the implementation		Not Completed
	Parallel Computing With Cuda	Jiali, Chen	Not Completed
	Optimize speed with Cython	Ying, Zhang	Not Completed

Jan 9-13	Interim Presentation	Jiali Chen, Ying Zhang	Completed
Jan 22	Interim Report	Jiali Chen, Ying Zhang	Completed
Feb 20	Mathematical Inference of Our Own Approach		Not Completed
Mar 15	Implementation of Our Own Approach with Python		Not Completed
Mar 30	Analysis of the Performance and Comparison with the Two Methods		Not Completed
April 16	Final Report		Not Completed

Table 1: Project schedule

6.2 Implementation of the Two Methods

As discussed in section 4.2(Programming Platform), python is selected as the programming language for our project. To implement the machine learning algorithms, libraries and tools such as numpy, pandas, sklearn and so on participate in the implementation process.

The objective function is the kernel of machine learning approach. Learning result is derived from optimal solution minimizing the objective function. Accordingly, a number of specific algorithms are designed to achieve the optimal solution towards the objective function. The task for implementation is to translate these algorithms into Python and optimize the time and space efficiency of the program on coding level. To conduct the optimization, parallel computing, special data structures are used to improve the performance. Details will be discussed in the following section.

6.2.1 Algorithms implemented

Hubert's Method	Hein's Method
Markov Operator	Solution of prox problem
Subgradient Method SGM	PDHG for ΩH,2
Semi-Supervised Learning	-

Table 2: Algorithm implemented for both methods

6.2.2 Programming Techniques

For hubert's method, originally it takes about 700 seconds to complete 2000 iterations, which is regarded as dissatisfactory. Thus parallel programming has been applied on Hubert's method to promote the running speed. However, it proved that parallel programming didn't produce a good result currently which may due to the overhead generated by creating as well as joining a large number of threads. GPU is considered as a good alternative to run the program as it has much more cores than CPU. Besides, data structure heap is considered to be applied in the next stage to improve the processes of finding extreme values. Moreover, Cython will be utilized seeking further improvement in speed since it conducts basic linear computation with C language which is more efficient than Python.

6.3 Comparison and Analysis Process

The experiment that analyzes and compares the performance of the two methods is consist of two major phases: performance of Hubert's method compared with supervised learning algorithm, performance of Hein's method compared with Hubert's method.

Before the experiment, data will be pre-processed according to the pre-process methodology introduced in section 5.2. Following APIs are used to deal with the rough data.

- dropna() from pandas module to abandon missing data
- train_test_split() from sklearn.model_selection class to select training data

6.3.1 Performance of Hubert's Method Compared with Supervised Learning Algorithm

The accuracy of the two methods as well as the picked supervised learning approaches have been plotted to see how these two methods perform compared with the existing classic methods. In this experiment, five classic supervised learning methods are selected for comparison, listed as follows: 1. Logistic regression, 2.Perceptron, 3.Support Vector Machine, 4.K Nearest Neighbor, 5. Naive Bayes. The line chart has been generated detailedly according to different sizes of learning sets.

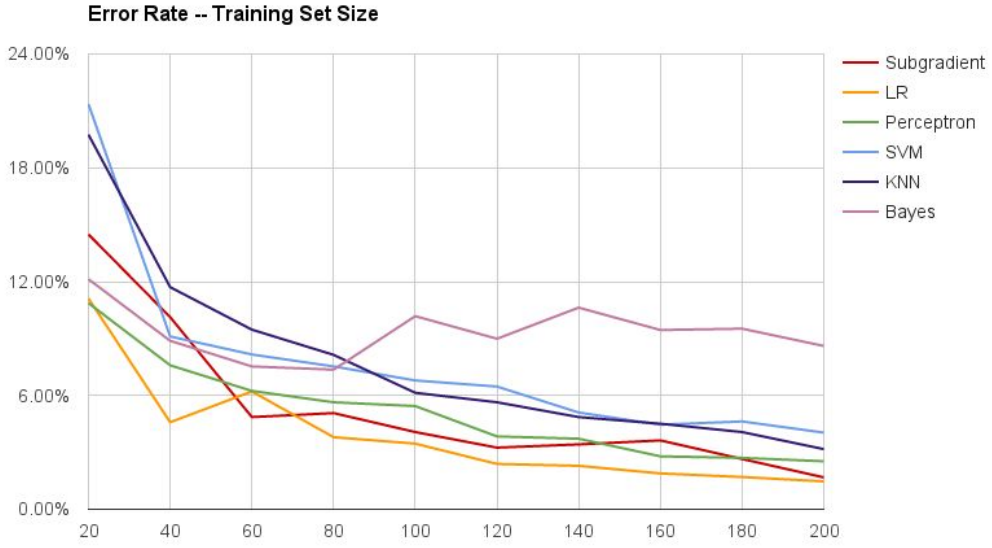


Figure 7: Error rate for Hubert's subgradient method v.s. A number of popular supervised methods. The total size of the dataset is 8124. The subgradient method(Hubert's methods) is proved to outperform most of the supervised methods selected.

From figure 7, we can draw the conclusion that the performance of subgradient method(Hubert's methods) is above of the average of all the five supervised learning method.

6.3.2 Implementation of Hein's Method

As mentioned in section 6.2.1, two algorithms of Hein's method has been implemented. Currently the validity of the implementation has not been examined, thus the comparison experiment cannot be performed at this stage. In the future, the performance of Hein's method compared with supervised learning algorithms will be tested. Moreover, the comparison of Hein's method with Hubert's method will also be conducted.

6.4 Future Plan for Own Approach

In the next semester, research on our own approach will be conducted. In our project plan there are four phases in this stage: mathematical inference, implementation of the algorithms, plots and analysis of the result, final analysis report.

6.4.1 Future Direction -- Generalization to Multi-class Cases & Distinguishing Heads and Tails

First of all, we plan to extend the current version of Hubert's method so that it is also applicable to multi-classes classifications, because currently this method can only be used in binary ones. The method planed to use for multi-classes is one-hot-encoding(e.g. If we are classifying a dataset with 3 classes, vectors [1 0 0] [0 1 0] [0 0 1] will be the represents). Ways to calculate loss and objective will be designed and experimented accordingly.

Also, we are interested in generating a specific method to distinguish the head and tail set of an edge in order to capture higher order causal relationship.

6.4.2 Mathematical Inference

Mathematical inference and explanation will be the first task for this phase. With a deeper understanding of the algorithms built in the implementation and experiment stage, we will have more information about the performance of the methods. Thus analyzing the limitation of the implemented two methods will be the first step of the mathematical inference. Aiming to the limitation, additional concepts other than direction will be defined on hypergraphs, accordingly, new version of objective function(function used to analyze the hypergraph) will be delivered mathematically. With the objective function, algorithms to minimize it will be further provided.

6.4.3 Implementation of the Algorithms

To examine the accuracy of our own approach and prepare for the potential future application of this approach, the implementation of our own approach will also be conducted. In general, the implementation process will also be translating the algorithms into code with Python, similar to section 6.2 (Implementation of the Two Methods).

6.4.4 Plots and Analysis of the Result

After the implementation, similar experiment methodology with what has been conducted in the first semester will be applied. Specifically, the same datasets and same size of training data will be adopted for comparison convenience. For each dataset, "accuracy"-"test set size" curve will also be derived and used to make a comparison with the former methods to verify the existence and the extent of the improvement.

6.4.5 Final Analysis Report

With the former derived figures and the statistics, a final report will be proposed. The report will cover not only the comparison result of Hein's method and Hubert's method, but also a detailed form of description, mathematical equations and pseudocode of our own method. In addition, the comparison result of our own method and the former two method will also be included as a major achievement of this project.

7.Deliverable

- Analysis and Comparison report for Hubert's method and Hein's method
    - Python implementation of Hubert's method and Hein's method
    - Plotting accuracy-training set size figures for both methods as well as a number of other selected methods
    - Analysis of the statistics of the experiment outcome
    - Comparison report of the two methods

      (Compare two methods using different datasets.)
    - Plots of selected supervised learning methods and comparison report of two target methods with these methods.

- Own Approach
    - Mathematical algorithms for our approach
    - Python interpretation of the algorithms
    - Accuracy-training set size figure for our own approach
    - Analysis of the statistics of the experiment outcome
    - Comparison report of our own method with the above two methods and other existing approaches

8.Difficulties and Limitations

8.1 Randomness of the Computation

Attribute to the mechanism of computer, arrangement of processes is not completely under control. Which means the computation time may vary even if the test data and the program are identical. Therefore, the randomness of computation will likely affect the time complexity as well as accuracy analysis when the difference is not significant.

Moreover, the training set selected from the entire data set is chosen randomly by the API provided by Python sklearn module. As illustrated by Figure 9 (left), there is no guarantee for a particular case, and such uncertainty will probably challenge the accuracy analysis result and confuse the reader.

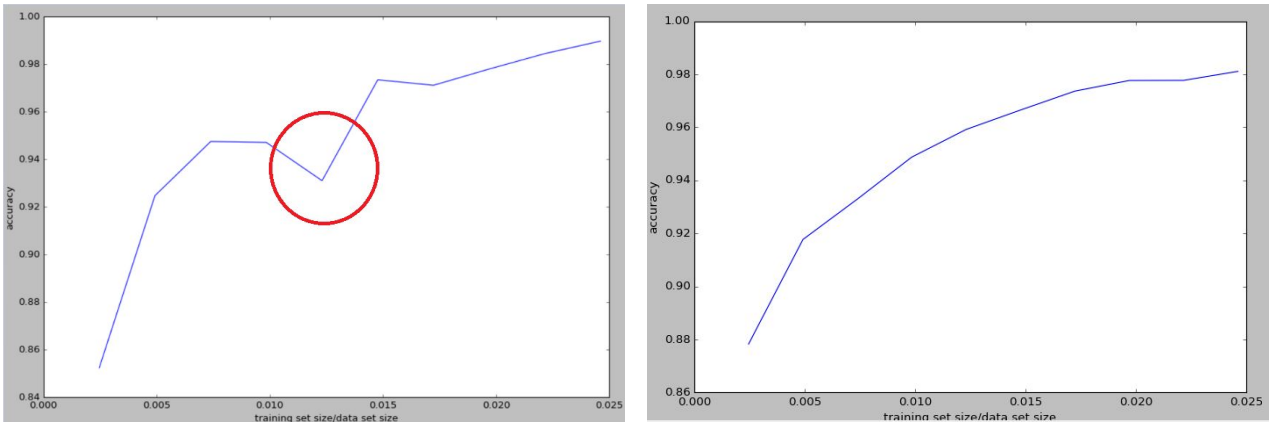


Figure 8: (KNN method) Curve without Repetitions for Average(left) v.s. Curve with 100 Repetitions for Average(right). Left: As labeled by the red circle, the curve encounters a sudden drop. The reason for the drop remains unknown because all the data points in the training set are randomly selected and the computation detail is difficult to trace due to the large dataset size. Right: With 100 repetitions, a trend is now considered instead of a particular case. The drop of the curve is now eliminated and the average accuracy smoothly increases with the increase of training set size, as the trend is a guaranteed outcome.

In order to relieve the adverse effects, experiments will be conducted with more repetitions, and the final result is defined as the average of all. Example of 100 repetitions are raised in Figure 9(right), Through this way, a general trend will be reflected instead of single results, which can reduce the influence of computation randomness and produce a more convincing result.

8.2 The Limitation of the Computation Ability

Due to the limited computation power, size of the test cases will be restricted. Thus it will be a problem whether the limited size will have adverse affection on the final performance analysis. To relieve or even resolve the impact, methods discussed in section 6.2.2 will be the assistance.

8.3 The Uncertainty of the Existence of Our Own Approaches

Since research work is always accompanied with uncertainty, it is difficult to guarantee that there always exist a solution which is overall dominating compared with the existing methods. Such uncertainty may make our project not fruitful as we planned. However, it can be guaranteed that there are many alternatives for the method. A substitution of loss computation, generalization of the current version, etc., are good provider of research direction. Hence, the project team keeps optimistic in estimating the possibility of finding our own approach.

9.Conclusion

In conclusion, this project aims to compare and analyze time complexity, space complexity and accuracy of two semi-supervised learning methods in the first stage and further propose an improved method in the second stage. Currently in the end of the first stage, literature review has been finished, a detailed plan of the experiment has been produced and primitive version of implementation of the two methods have been completed. Based on the implementation, a number of comparison results has been plotted through experiments. We will keep optimizing the implementation and make attempts on different choices of alternatives of the current algorithm. Overall, the progress is on schedule and the project team believe that we are heading to the right direction. In the future, with more efforts devoted, more progress will be achieved soon.

References

[1] Zhu, Xiaojin, Zoubin Ghahramani, and John Lafferty. "Semi-supervised learning using gaussian fields and harmonic functions." *ICML*. Vol. 3. 2003.

[2] Zhou, Dengyong, Jiayuan Huang, and Bernhard Schölkopf. "Learning with hypergraphs: Clustering, classification, and embedding." *Advances in neural information processing systems*. 2006.

[3] Hein, Matthias, et al. "The total variation on hypergraphs-learning on hypergraphs revisited." *Advances in Neural Information Processing Systems*. 2013.

[4] Zhou, Dengyong, et al. "Learning with local and global consistency."*Advances in neural information processing systems* 16.16 (2004): 321-328.

[5] Blum, Avrim, and Shuchi Chawla. "Learning from labeled and unlabeled data using graph mincuts." (2001): 19.