Survey Analysis with Statistical Machine Learning and Graph Learning

Baichen YANG
20493198
byangak@connect.ust.hk

Chengzhong LIU
20327961
chengzhong.liu@connect.ust.hk

Zeyu HUANG
20493631
zhuangbi@connect.ust.hk

Zijun CHEN
20583101
zchendg@connect.ust.hk

ABSTRACT

Survey has been a long standing research tool for social scientists. With the newly emerged trend of the computational social science, quantitative analysis on the survey data from the social science perspective has grown increasing attention. While the traditional statistical machine learning method has been widely used on such survey analysis, few state-of-the-art new machine learning techniques has been used in this area of study. In this paper, we used graph learning methods including Graph Convolutional Networks (GCN) to complete three pre-defined survey analysis tasks and also proposed methods in statistical machine learning to do similar analysis. To this end, we can have a comprehensive comparison between these two methods, which can shed light on future survey analysis with GCN.

INTRODUCTION

How to analysis social science data has long been a troubling issue for social scientists, as survey data can be inaccurate, time-sensitive and not organized [13]. Moreover, with the development of computational social science, the volume of the survey data collected by social scientists has grown drastically and the demand for a data-driven style analysis is also trending in recent decade [11].

In this emerging interdisciplinary area of study, social scientists generally collect independent survey specialized for onetime publication which causes a serious issue on data reusability [1]. The data collection word is generally time-consuming which calls for a need to effective to use the survey data collected by previous work. The current solution is generally use conventional statistical machine learning method to predict the unforeseen survey questions, but the limitation is also predominant as the reliability is questionable albeit such methods can provide certain level of explainability [2].

COMP5212'21, Feb. 1 – May. 7, 2021, Hong Kong, China

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ISBN 5212-5212-5212-5212. . . $66.66
DOI: http://home.cse.ust.hk/~lzhang/teach/5212/

Yet with the recent rapid development of deep learning based algorithms, some approaches have yielded extraordinary results in certain tasks, but the prediction contexts are usually limited to computer vision or highly-structured dataset [16].

Our approach is to use graph learning method including Graph Convolutional Networks (GCN) [9] to incorporate response similarity into the embeddings to individual survey participants. In theory, such approach can be robust to misinformation in survey data and adopt some relationships between people. The current research gap is that general survey analysis still treats people's response as independent [6], where similarities between people in survey responses are still under-explored. By studying this gap, we can effectively understand participants and with assumption that most of the responses from survey data is trustworthy, we can also establish a pipeline to identify unexpected responses in the survey which might bring new insights to social scientists.

In this work, we propose three tasks to evaluate the effectiveness of using graph learning methods to analyze the survey data, which can be concluded as: **T1:** Outlier detection, **T2:** Single response prediction, **T3:** Domain response prediction. We use various statistical machine learning tools as the baseline and compare them with the results by graph learning methods. The research contributions of this work are three-fold: 1) we conducted a systematic examination of a social science survey dataset; 2) we designed the tasks that reflect the need of the social scientists and analyzed them with mature statistical machine learning models; 3) we used graph learning method to embed survey participants to enhance the interpretablity of the prediction results and lay foundations for future work in understanding people2vec [10], which is the first to use the graph learning method approach in this context to our best knowledge.

RELATED WORK

Dataset

Our dataset is obtained from a social science research conducted in 2015 [14]. It is a survey about Chinese Political Compass (CPoC), which is modeled on *The Political Compass* (http://www.politicalcompass.org), originally from UK. The actual survey site was set up by individuals affiliated with Peking University and can be found at **https://www.zuobiao.me/**. CPoC collected the respondents' attitudes in political, economic and social/cultural in agreement level of 50 statements, distributed in pattern 20-20-10. The questions were designed in 4-point Likert scale, which means forced choice is required. CPoC also collected some background information from the respondents, but we are not going to discuss here. By the time we access the data, there has been over 500k respondents. To sum up, the survey data we are going to deal with is around a 500k 50 dimensional matrix with integer 1-4, representing the Likert scale in design.

Graph Convolutional Networks

GCN Principle Introduction

As traditional Convolutional Neural Networks applies for the image analysis that essentially relies on *local translational invariance* of image. The graph network doesn't preserve the property as well as the local structure is not preserved. Traditional CNN couldn't be applied on graph networks analysis. We introduce GCN on analysis the database to give outlier detection, point prediction and cross domain analysis.

GCN introduce the Graph Laplacian:

$$\mathbf{L} = \mathbf{D} - \mathbf{W}$$

Where $\mathbf{D}, \mathbf{W}$ are the degree matrix and adjacent matrix. We define the Fourier transformation on the graph network. For Fourier transformation is a special type of the Laplacian which is the gain obtained after small changes in all the degree of freedom. Such a concept reflect the cumulative gain generated when the current node disturbs the surrounding nodes, Fourier transformation on graph network is introduced:

$$\mathcal{F}_T(\lambda_k) = \hat{f}_k = \sum_{i=1}^{N} f(i)u_k(i)$$

As $f_k = (f_k(1), \cdots, f_k(n))$ is the vector defines on the graph, $f_k(i)$ represent the k-th component for node i. For particular eigenvalue f_k, the corresponding Fourier transformation for f is the inner product for f and the eigenvector corresponding to $u(k)$.

Fourier Transformation carries the convolution procedure in frequency domain, thus defines the convolution operation on graph:

$$\begin{aligned}(f * g)_G &= \mathcal{F}^{-1}[\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}] \\ &= \mathcal{F}[U^T f \cdot \hat{h}] = \mathcal{F}^{-1}[diag[\hat{h}_1, \cdots, \hat{h}_n]U^T f]\end{aligned}$$

Where the inverse transformation is equals to the left product for U:

$$(f * g)_G = U diag[\hat{h}_1, \cdots, \hat{h}^n]U^T f$$

The goal for GCN is to learn the shared parameter $diag[\hat{h}_1, \cdots, \hat{h}_n]$

The layer-wise propagation between hidden layers t and $t+1$ are:

$$H^{(t+1)} = \sigma(D^{-\frac{1}{2}} A D^{-\frac{1}{2}} H^{(t)} W^t)$$

Where σ is the activation function, and A and D are the adjacent matrix and degree matrix.

The cross-entropy function is:

$$L = -\sum_{l \in y_L} \sum_{f=1}^{F} Y_{lf} \log Z_{lf}$$

GCN grab the gain near the node through graph convolution, get the graphic feature, pass the processed graph network to the next layer and continue to get the graphic feature according to the label assigned then get the classification.

GCN on Survey Analysis

The idea for the survey analysis is inspired by [9]: In previous work, the paper discuss on the Cora dataset graph that consists of academic publications as nodes and the citations as the links (undirect). The node are classified into one of seven subjects and GCN model on the graph will learn to predict the subject.

a graph: this notebook uses the Cora dataset from https://linqs.soe.ucsc.edu/data. The dataset consists of academic publications as the nodes and the citations between them as the links: if publication A cites publication B, then the graph has an edge from A to B. The nodes are classified into one of seven subjects, and our model will learn to predict this subject. an algorithm: this notebook uses a Graph Convolution Network (GCN) [1]. The core of the GCN neural network model is a "graph convolution" layer. This layer is similar to a conventional dense layer, augmented by the graph adjacency matrix to use information about a node's connections. This algorithm is discussed in more detail in "Knowing Your Neighbours: Machine Learning on Graphs".

The basic idea for carrying GCN method on our survey analysis is to treat people's answer as human vector $v \in R^{50}$. Connect the vector according to **Similarity Threshold: Two human-vectors in the graph are connected if the number of same answer exceed the Similarity Threshold**

Our pre-training GCN model contains one input layer, 15 hidden propagation layers and one final output layer. The layer-wise propagation in hidden layers is:

$$V^{(t+1)} = ReLU(D^{-\frac{1}{2}} E D^{-\frac{1}{2}} V^{(t)} W^{(t)}) \tag{1}$$

where $t = 0, 1, \cdots, 15$ and $D = diag(d_1, d_2 \cdots d_T)$ is the diagonal matrix with $d_i = \sum_{j=1}^{T} E_{ij}$. $W^t \in R^{d_t \times d^{k+1}}$ iwth initially $d_0 = 49$ is a layer-specific weight matrix that we trained for the model. Between each layer, We use $ReLU(\cdot)$ to increase the training efficiency. The output layer we define is:

$$Z = softmax(D^{-\frac{1}{2}} E D^{-\frac{1}{2}} V^{(16)} W^{(16)})$$

The final output Z represents the embedded vectors of the input. The cross-entropy loss function that is going to be minimized as:

$$\mathcal{L}_{GCN} = -\sum_{i=1}^{T} \sum_{j=1}^{4} Y_{ij} \log Z_{ij}$$

where $Y_{ij} = 1$ only when the i-th nodes j is the class that i-th node subject to, otherwise, $Y_{ij} = 0$, and Z_{ij} is the corresponding possibility. The accuracy metric is the percentage of correct predictions. Where error metric is just the mean and std of the absolute difference between the ground truths.

OUTLIER DETECTION

Problem Framing

The task on outlier detection is defined as detecting the participants with abnormal answers in a survey regarding their backgrounds. Such a task can be basically formulated as a data points clustering problem: to cluster similar peoples together and recognize the noises among the clusters.

The main motivation of this task is that, there might be respondents who answered the questions in a pure random manner or injected malicious answers deliberately. Solutions to this outlier detection problem can be useful for survey analysts to spot these noise or malicious inputs in the result.

Assumption

To frame this problem, we made an assumption that most people answered the questions correctly, and similar answers will be given to the same questions by people who have similar backgrounds. In the following discussions on detection methods, we conduct data point clustering based on this similarity assumption.

Validation

To validate that an effective clustering can be made on this dataset, we checked its *Hopkins statistics* [7]. This metric measures the cluster tendency of one dataset. Consider to randomly sample n data points as Y from the original dataset X. The definition of the *Hopkins statistics* is the following,

$$H = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}$$

where y_i is the distance of some data point $p_i \in Y$ to its nearest neighbor in X, and x_i is the distance of some random data point $q_i \in X$ to its nearest neighbor in X.

From this formula, we can see that a value close to 1 shows that the data is highly clustered, while a value close to 0 indicates a near-uniform distribution. In practice, the common threshold to determine whether the dataset is worth clustering or not is 0.5. For this specific dataset, we get its

$$H = 0.562 > 0.5$$

Thus, such a dataset has the potential to be clustered into different meaningful groups.

Metrics

Our evaluations are based on two metrics, *Silhouette Score* [15] and *Calinski Harabasz Score* [3].

1) The *Silhouette Score* is used to measure how well samples are clustered with samples that are similar to themselves. A higher score indicates a denser and better separated clustering. Its definition for one single point is

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

, where $a(i)$ is the mean distance between data point i and all the other points in the same cluster, and $b(i)$ is the smallest mean distance between data point i and all points in any other cluster. From the formula we can see that the best score is 1 and the worst score is -1. Values near 0 indicates overlapping clusters.

2) The *Calinski Harabasz(CH) Score* is one general metric to measure the effectiveness of clustering. It is defined as the ratio between the intra-cluster dispersion and inter-cluster variance. Consider a dataset E being separated into k clusters, the

$$s = \frac{trace(B_k)}{trace(W_k)} \times \frac{n_E - k}{k - 1}$$

, where

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T$$

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T$$

with C_q the set of points in cluster q, c_q the center of q, and n_q the number of points in q. From the above definition, we can see that a denser and better separated clustering is made if the CH Score is higher.

Conventional Method

There are plenty of mature clustering algorithms in conventional statistical machine learning, including *K-Means* [12], *DBSCAN* [4] and *Affinity Propagation* [5]. Here we conducted experiments on and compared the first two methods, which are of heavy usages.

K-means

The basic idea of *K-Means* algorithm is to partition n data points into K clusters, where each point is classified into the cluster with the nearest mean. Such a method minimized the intra-cluster variances, but are sensitive to abnormal points as they will hugely affect the cluster mean.

We consider the data points in this dataset as human vector $v \in \{1, 2, 3, 4\}^{50}$. And we utilized the Euclidean distance as the distance metric for clustering.

Under the settings of this method, we define an outlier as a data point v_i which satisfies that

$$dist(v_i, c_E) \geq \text{threshold} \cdot \max_{v_j \in E}(dist(v_j, c_E))$$

, where E is the cluster that v_i is classified into and c_E is the centroid of the cluster E.

Since the k value of *K-means* algorithm need to be manually determined, we performed our evaluation on different settings of $K = 3, 4, 5$. The threshold for outlier detection is set to be 0.8. Results on both of the metrics are listed in the Table 3.

From the above results, we can see that under the setting of $K = 3$, both the Silhouette Score and the CH Score are the

K	CH Score	Silhouette Score
3	905.6	0.061
4	705.2	0.042
5	582.1	0.022

Table 1. Metrics on K-means under different K settings

highest among the three settings, indicating a better separation of outliers and normal points. Thus, we take the *K-means* method with $K = 3$ setting as one of the baseline candidates. Under such setting, the outlier percentage detected is 0.0109 among the whole cluster.

DBSCAN

An alternative approach to this problem is to conduct clustering based on density. One conventional algorithm is $DBSCAN$, which is basically grouping points into clusters based on their distance. Points within a certain distance from points in a cluster will be grouped into that cluster. Those points out of the range of any other cluster will be marked as outliers and put into the same category.

Since outliers have already been labelled in this method, there is no need to set an extra threshold to classify outliers. However, DBSCAN requires two parameters, ϵ and minPts, corresponding to the radius of a neighborhood to some point and the minimum number of points to form a cluster respectively. Thus, we conducted experiments to find the optimal ϵ and minPts based on the two metrics. Results are shown in Table 12.

ϵ, minPts	CH Score	Silhouette Score
6, 5	23.59	0.052
6, 10	51.62	0.180
6, 15	77.51	0.182
7, 5	23.17	0.247
7, 10	20.54	0.255
7, 15	20.82	0.255
8, 5	17.16	0.276
8, 10	5.793	0.301
8, 15	5.793	0.301
9, 5	3.571	0.329
9, 10	3.571	0.329
9, 15	3.571	0.329

Table 2. Metrics on DBSCAN under different ϵ and minPts settings

From the results, we can see that under the setting combination of $\epsilon = 7, \text{minPts} = 5$, the CH Score and Silhouette Score achieve a satisfying condition, both of them are of a comparatively high value. Under such a setting, we obtained an outlier percentage of 0.0209.

By comparing the two baseline methods, we can see that in the Silhouette Score, $DBSCAN$ with $\epsilon = 7, \text{minPts} = 5$ performs better than $K - means$ algorithm with $K = 3$. And due to the fact that CH Score is generally larger in convex clusters than density-based clusters, we should not take CH Score into the consideration in the comparison of these two baseline methods. Hence, based on those experiments and observations, we propose the result of $DBSCAN$ clustering with $\epsilon = 7, \text{minPts} = 5$ as our baseline towards the following GCN method.

The GCN Method

K-means

Our approach to this problem is using GCN method for filtering the dataset. In GCN method apply for outlier prediction, the model output a 128 dimension vector that contains the information of the embedded human vector itself as well as the association with its neighbors. After embedding the human vectors, *K-Means* [12], *DBSCAN* [4] apply on the output vectors, select outlier follows the same principle with the previous. We performed the evaluation on same setting of $K = 3, 4, 5$ for convenient to make comparison with what we have down in previous with the directly use of **Conventional Method**. Results on the metrics are listed in Table 3.

K	CH Score	Silhouette Score
3	22.2	1.990
4	16.7	1.972
5	14.3	2.194

Table 3. Metrics on K-means under different K settings with GCN processing

From the above results, we can see the CH Score are significant small, indicating that the human vectors after the GCN processing are tending to be more clustered. And the Silhouette Score are relatively large implies the data overall presents a scattered distribution from a larger perspective. Thus, for the *K-means* method with $K = 3$ gives the maximal value for CH Score,, setting $K = 3$ as the base line candidates. The outlier percentage detected is 0.0045 among the whole cluster for the training set.

DBSCAN

Similarly with what we have done in Conventional Method, we carry the *DBSCAN* for the human vector that processed by the GCN method. The assumption carries same with what we have down in previous, the goal for the experiment is to find the optimal ϵ and minPts based on the CH Score and Silhouette Score. Results are shown in Table 4.

ϵ, minPts	CH Score	Silhouette Score
0.1, 1	0.09	0.050
0.1, 2	1.64	-1.192
0.1, 3	2.36	-0.163
0.3, 1	35.33	0.053
0.3, 2	1.24	-0.253
0.3, 3	1.72	-0.252
1, 1	9.228	0.559
1, 2	9.228	0.559
1, 3	9.228	0.559

Table 4. Metrics on DBSCAN under different ϵ and minPts settings with GCN processing

From the results, we see that when $\epsilon = 1$, and minPts=1 both gives the CH Score and Silhouette Score to be relatively large simultaneously. At that value, we have the outlier percentage of 0.00955.

By comparing the two baseline methods with the human vectors that processed by GCN method, we see that performs better*K-means* algorithm with $K = 3$ than that of *DBSCAN* algorithm.

SINGLE POINT PREDICTION

Problem Framing

Single point prediction is defined as predicting the answer to a single question in a survey. This may be useful if the survey data is partially lost or contaminated, or a extra question is added during the distribution of the survey, and hence not all respondents answer those questions. In addition, we require that at least some respondent have answered this question. These "complete" responses will be utilized to analyze the relationship between this question and the remaining questions, and predict the answers to this question in other "incomplete" responses.

To approach to this problem, we follow the assumption made in the *outlier detection* section that people's general ideas should cluster. If two people give similar answers to the remaining questions, then they should also answer similarly to this question. In our following experiments, this assumption is proven hold within an acceptable error range.

Conventional Method

KNN

Similar to *outlier detection*, the response vectors themselves are considered as human vectors $v \in \{1, 2, 3, 4\}^{50}$. Then, a heuristic approach inspired by the previous section is KNN.

For a incomplete test sample $u \in \{1, 2, 3, 4\}^{50}$ whose answer to the q-th question u^q is missing, consider the subspace $\mathbb{R}^{49}_q$ formed by the remaining questions. Since both the test sample and the complete samples have answered the remaining questions, we can compute the Euclidean distance of the vector projections on this subspace (*i.e.,* the distance of the vectors in $\mathbb{R}^{49}_q$ formed by the remaining questions excluding the q-th), and take the nearest K complete samples (namely $v_1, v_2, \ldots, v_K$) to this incomplete test sample.

The weighted average of $v^q_1, v^q_2, \ldots, v^q_K$ (*i.e.,* the K samples' answers to the q-th question) is then computed. The weights are the softmax of the negative distances, so a complete sample with a shorter distance will gain a larger weight.

$$weight(i) = \frac{\exp(-dist(v_i, u))}{\sum_{j=1}^{K} \exp(-dist(v_j, u))}, \quad 1 \le i \le K \tag{2}$$

$$u^{q*} = \sum_{j=1}^{K} v^q_j weight(j) \tag{3}$$

The averaged u^{q*} is fractional, it is then rounded to $\{1, 2, 3, 4\}$ by regular Thresholds $1.5, 2.5, 3.5$.

Table 5 shows the experiment results of masking 20% answers to the last question and predicting each sample based on the rest 80% complete samples with $K = 5$. The accuracy metric is the percentage of correct predictions. The error metric is the mean and std of the absolute differences between the ground truths and the predicted values (before and after rounding). Another unweighted version is also shown in Table 5, where $u^{q*} = \frac{1}{N}\sum_{j=1}^{K} v^q_j$ is the unweighted average of the $K = 5$ nearest samples' answers.

	Accuracy	Error mean/std (before rounding)	Error mean/std (after rounding)
w/ weights	53.91%	0.539/0.449	0.508/0.587
w/o weights	52.07%	0.560/0.441	0.527/0.588

Table 5. Single point KNN prediction on q50

As is shown in the table, weighted average introduces a slight improvement to the accuracy and the errors. After rounding, the prediction error has a slightly increased std as fractional predictions are snapped to integers. On overall, KNN method can reach to 54% accuracy; an error of mean 0.51 and an std 0.59 suggest that most incorrect predictions are close to the ground truth, within a ±1 range.

To further test the stability of the method, more tests are conducted on 10 more randomly picked questions. For each question, 20% randomly selected samples are masked and predicted. We use weighted averages and $K = 5$.

The results are shown in Figure 1. On average, we reached an accuracy of 53.82%, and an error mean/std of 0.541/0.449 before rounding and 0.508/0.586 after rounding. This is consistent with the previous results, and the performance is quite stable.

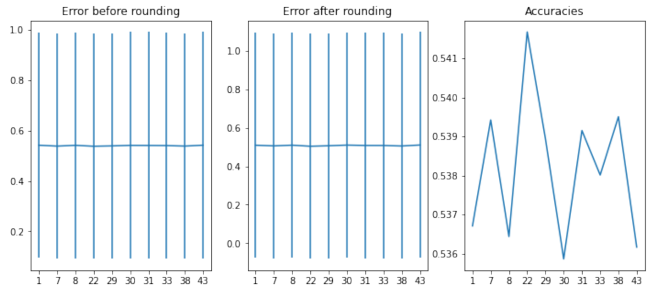

Figure 1. Single Point KNN predictions on 10 questions

Gradient Boosting Decision Tree

Another baseline approach to the single point prediction problem is experimented using LightGBM [8], a state-of-the-art Gradient Boosting Decision Tree (GBDT) algorithm. The general settings is similar to the KNN approach: 20% randomly selected data have the q-th answer masked, and the remaining data as vectors in the $\mathbb{R}^{49}_i$ subspace is used as features. Instead of computing the Euclidean distances, the subspace vectors of the 80% complete samples, are shifted to $\{0, 1, 2, 3\}$ and fed into the decision tree as numerical data. Their q-th answers are fed as numerical targets. The model is trained for 100 iterations with default parameters.

Then the subspace vector of each masked sample u is fed into the model to get a predicted, fractional u^{q*}. In addition to the regular rounding with Thresholds 0.5, 1.5, 2.5 (we have shifted the inputs by -1), an alternative set of Thresholds 0.75, 1.5, 2.25 is also experimented. This is derived from dividing the range $[0, 3]$ to four splits. We concern that the model may learn not to exceed the input range and thus compress the output values to $[0, 3]$. For KNN, although the output range is also $[0, 3]$, the range is determined by the input values, not the algorithm. From another perspective, the most "hard-to-say" prediction should be when the two closest vector v_1, v_2 have the same subspace distance from the predicted u, and their q-th answers are $v_1^q = v_2^q \pm 1$. These cases always result in a .5 decimal, hence the regular rounding Thresholds are used in KNN.

Table 6 shows the prediction results masking the last question and set $K = 5$. The alternative rounding introduces a very slight (nearly negligible) improvement on the accuracy.

	Accuracy	Error mean/std (before rounding)	Error mean/std (after rounding)
reg. rounding	54.54%	0.551/0.389	0.491/0.569
alt. rounding	54.69%	0.551/0.389	0.492/0.574

Table 6. Single point LightGBM prediction on q50

More experiments on predicting 10 randomly selected questions are conducted using the alternative rounding. Figure 2 show the detailed results. On overall, the GBDT approach is on par with, but slightly less stable than the KNN approach. We have reached an average accuracy of 52.92% and error mean/std of 0.582/0.456 before rounding and 0.528/0.606 after rounding.

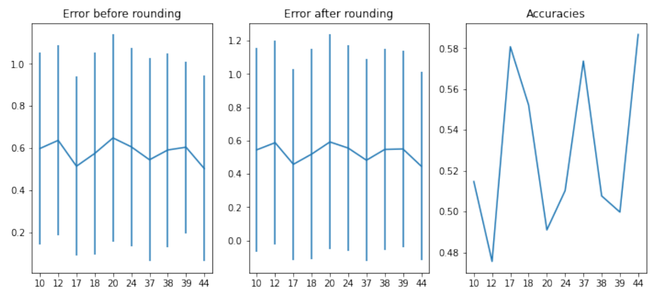

Figure 2. Single Point LightGBM predictions on 10 questions

The GCN Method

For GCN Method analysis on the survey. Fix the 50-th question to be the question that we want to predict, the node for the input graph is represented by a vector $v_i \in R^{49}$ associated with remaining 49 questions. The entries for $v_i \in \{1, 2, 3, 4\}^{49}$ are the answer that people answered for k-th question by i-th participant. Let $(v_1, \cdots, v_T) \in R^{T \times 49}$ be the collection of T data vectors with 49 dimension forming the training set. And four options for 50-th question are the categories labelled for each node.

The edges express the correlation between different people. Here, we follow a natural assumption that people are more likely to give the similar answer for 50-th question if they give same results on more questions. If two human vectors have the number of same answers that exceed Similarity Threshold, then we connect the two nodes.

During the training procedure, we set the training set to be different sizes with: $T = \{200, 500, 1000\}$ and Similarity Threshold= $\{12, 18\}$ to construct the graph.

After clearly defined the graph $G(V, E)$. The validation set is with size V, the Accuracy is measured by:

$$Accuracy = \frac{1}{V} \sum_{i=1}^{V} \mathbb{1}(\hat{c}_i = c_i)$$

Where $\hat{c}_i$ is the predicted class for i-th nodes subjects to and c_i is the actual class for i-th nodes subjects to.

T	Similarity Thresholds	Accuracy	loss entropy
200	12	0.4600	1.3101
200	18	0.4400	1.2496
500	12	0.4857	1.2678
500	18	0.4857	1.1663
1000	12	0.5141	1.218
1000	18	0.5132	1.2678

Table 7. Single point prediction on q50 with GCN

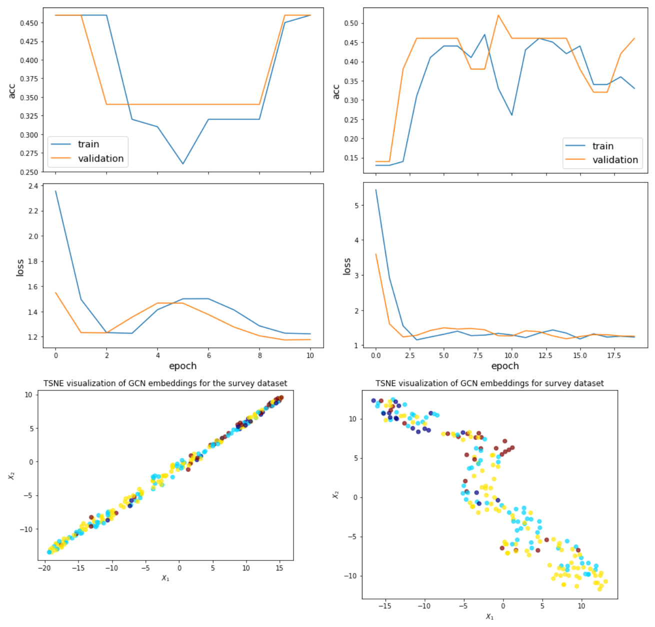

Figure 3. Left column: T=200, Threshold=12; Right column: T=200, Threshold=18

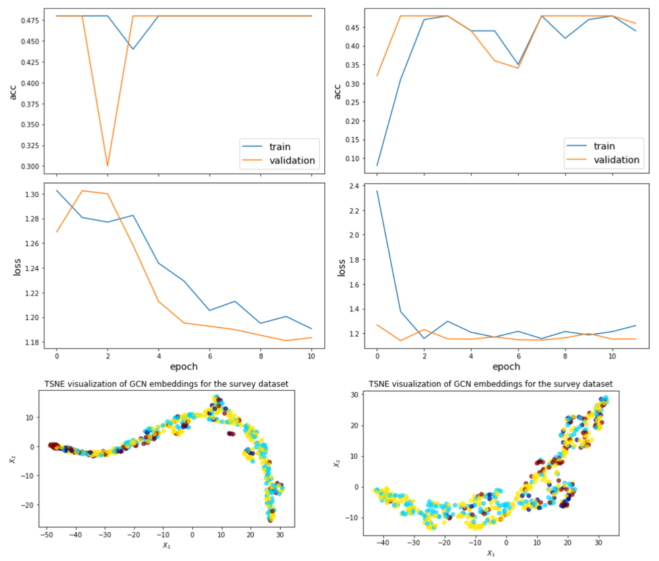
Figure 4. Left column: T=500, Threshold=12; Right column: T=500, Threshold=18

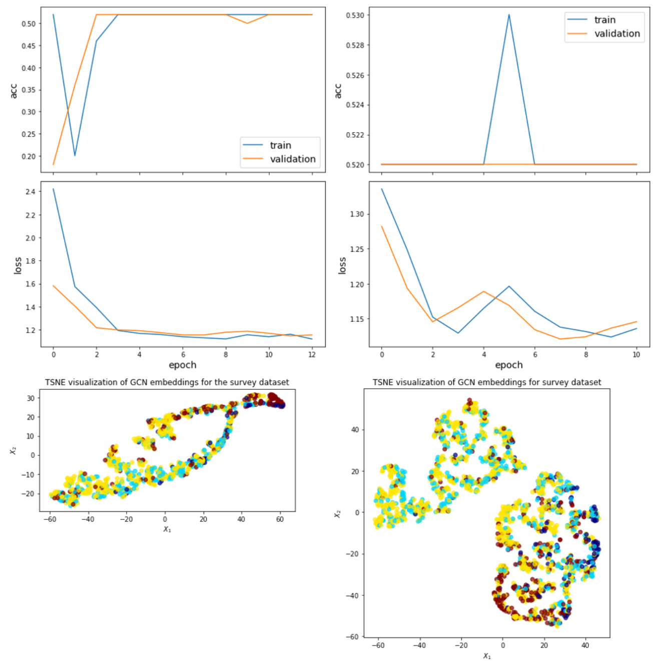
Figure 5. Left column: T=1000, Threshold=12; Right column: T=1000, Threshold=18

Where the colors represents:

- red: option 1
- blue: option 2
- green: option 3
- yellow: option 4

Table 2 show that the optimal accuracy are achieved when T=1000 and Similarity Thresholds=12 or 18 (18 is better, the difference are small).

Figures 3, 4, 5 show the the accuracy and entropy's variation during GCN training progress under different training size T and Similarity Threshold. Moreover, the graphs with colorred points in last column of each large graphs (3, 4, 5) are the visualization of GCN embedding for the survey dataset that reflect the classification for single point prediction accuracy with different training parameters.

From the visualization classification graphs, they show that when the training size T becoming larger, then the GCN embedding will preserve the better classification to the graph. And the GCN method on prediction survey answer is slightly weaker than the conventional machine learning method. The analysis will provided in discussion part.

CROSS DOMAIN PREDICTION

Problem Framing

Cross domain prediction is defined as predicting the *entire* survey response of a person, based on his/her responses to other surveys and other peoples responses to this and other surveys. This may be useful if we want to collect data in another domain from the same group of people, but cannot reach to some of them any more. This may also be useful if the surveys require a comprehensive coverage, and we want to reduce the distribution cost of the second survey.

Technically, this problem is similar to single point prediction, but the prediction output becomes a vector instead of a single scalar. We follow a similar assumption that people's general ideas in several domains should cluster similarly. If a group of people think the same in one domain, they tend to also think the same in another domain. In our following experiments, this assumption is proven hold within an acceptable range.

Conventional Method

Since LightGBM only deals with scalar target, we only used KNN for this problem. The settings are the same, except that all answers of a domain (20 for political & economic, 10 for social/cultural) are masked for the 20% test samples. The Euclidean distance is calculated based on the remaining feature subspace $\mathbb{R}^{30}$ or $\mathbb{R}^{40}$. During prediction, the weighted average of the K nearest samples is calculated element-wise.

The accuracy metric becomes the percentage of correct predictions of all masked questions of all test samples. The error metric also becomes the mean and std over all test samples of the average distance between the ground truth and the predicted values over all masked questions.

Let M denotes the numbers of test samples, C denotes the number of masked questions (the dimension count of the target subspace), $u_i, \hat{u}_i \in \mathbb{R}^C$ denotes the ground truths and the predicted values of the i-th test sample's projection onto the target subspace. We have

$$\text{Accuracy} = \frac{1}{MC} \sum_{i=1}^{M} \sum_{q=1}^{C} \mathbb{1}(\hat{u}_i^q = u_i^q) \tag{4}$$

$$\text{ErrorMean} = \operatorname*{mean}_i \left\{ \frac{1}{C} \| u_i - \hat{u}_i \|_{L_1} \right\}_{i=1}^{M} \tag{5}$$

$$\text{ErrorStd} = \operatorname*{std}_i \left\{ \frac{1}{C} \| u_i - \hat{u}_i \|_{L_1} \right\}_{i=1}^{M} \tag{6}$$

Table 8 shows the detailed results of predicting the "social/cultural" domain based on "political" and "economics". The results are consistent with Table 5 in *Single Point Prediction* section. The error std is smaller since we take the average of $\left|u_i^q - \hat{u}_i^q\right|$ over all $1 \le q \le C$ before calculating the std. The averaged std should be roughly $\frac{1}{\sqrt{C}}$ of the single point std.

	Accuracy	Error mean/std (before rounding)	Error mean/std (after rounding)
w/ weights	53.46%	0.558/0.248	0.516/0.277
w/o weights	51.62%	0.578/0.224	0.535/0.258

Table 8. Cross Domain KNN prediction on "social"

Figure 6 shows the results predicting each of the "political", "economic" and "social" domains. We have reached a 50.70% accuracy and error mean/std of 0.598/0.236 before rounding and 0.558/0.261 after rounding.

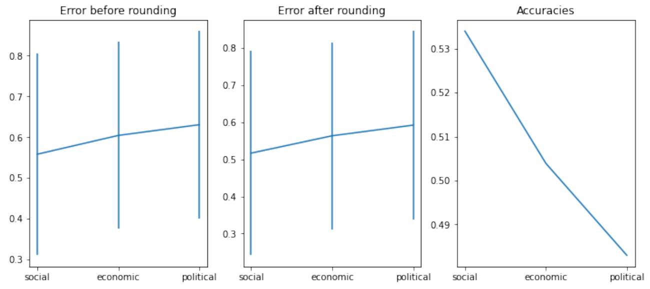

Figure 6. Cross Domain KNN predictions on all 3 domains

The GCN Method

GCN Method in Cross domain prediction is similar with what we have done in single point prediction part. To predict the response of a person in the whole domain. The survey include 3-area: politics, economics and social. We implement GCN method on the data graph to produce a model that accepting two domains to give a vector representing the answers to the questions of the third domain. As label is abandoned from the problem, it is an unsupervised learning.

The idea for the implementation for the problem is carried as follow: The training set and Similarity Threshold are as same as what we have done in the "Single point prediction" part. We first using the training set to carry the training progress. When embedding a human-vector that we want to predict for cross domain answer, the model finds the 5-nearest neighbors of the vectors after embedding into the graph. Then the final prediction result is determined by the mode of each result corresponding to these 5-nearest neighbor vectors.

The final metric becomes the percentage of correct predictions of all masked questions of all test samples:

$$Accuracy = \frac{1}{VC} \sum_{i=1}^{T} \sum_{q=1}^{C} = \mathbb{1}(\hat{c}_i^q = c_i^q)$$

Where V is the size of the whole, and C is the question number of the cross domain that is going to predict. c_i and $\hat{c}_i$ is the i-th prediction with dimension C. We measure the total answer that we predict correct divides the total questions we predict. To be coherent with Single point detection, set the training set to be different sizes with $T = \{200, 500, 1000\}$ and Similarity Threshold with $\{12, 18\}$ for prediction:
Prediction on social domain : Performance: 0.558 accuracy.

T	Similarity Thresholds	Accuracy	loss entropy
200	12	0.505	0.6937
200	18	0.538	0.6803
500	12	0.506	0.6933
500	18	0.545	0.6775
1000	12	0.500	0.6931
1000	18	0.558	0.6675

Table 9. Cross domain prediction on Social domain with different parameters

The optimal accuracy is reached when training set T=1000 and Similarity Threshold=18

Prediction on economics domain: Performance: 0.647

T	Similarity Thresholds	Accuracy	loss entropy
200	12	0.522	0.6913
200	18	0.592	0.6570
500	12	0.520	0.6898
500	18	0.639	0.6124
1000	12	0.535	0.6851
1000	18	0.647	0.5971

Table 10. Cross domain prediction on Economics domain with different parameters

accuracy. The optimal accuracy is reached when training set T=1000 and Similarity Threshold=18

Prediction on politics domain: Performance: 0.629 accuracy. The optimal accuracy is reached when training set T=1000 and Similarity Threshold=18

The above result show that, the GCN model performs better than the conventional method for the cross domain prediction. And the model gets better results when the Similarity Threshold is relatively large. For stick Similarity Threshold

T	Similarity Thresholds	Accuracy	loss entropy
200	12	0.509	0.6930
200	18	0.569	0.6698
500	12	0.515	0.6914
500	18	0.600	0.6463
1000	12	0.515	0.6884
1000	18	0.629	0.6020

Table 11. Cross domain prediction on Politics domain with different parameters

condition implies a sparse adjacent matrix performs better as well as the correlation between human-vectors contributions a positive role in the prediction of the model if and only if the correlation is strong enough. If we build the connection edge with vectors with weak correlation, the model will overfitting hence leading to a low accuracy.

At the same time, we also use GCN method to optimal the Similarity Threshold for contructing the human vector graph. Notice that increasing on the training size don't increase accuracy a lot, such a fact inspires us that. Contrast with that, increasing of the Similarity Threshold increase the accuracy a lot. We do the further test on a training set with appropriate size $T = 500$ with different Similarity Threshold to find a better threshold for constructing the adjacent matrix. The result is as below (Stride=2):

Prediction on economics domain:

T	Similarity Thresholds	Accuracy	loss entropy
500	12	0.520	0.6898
500	14	0.558	0.6734
500	16	0.602	0.6478
500	18	0.639	0.6124
500	20	0.658	0.6319
500	22	0.629	0.6284

Table 12. Cross domain prediction on Economics domain with different Similarity Thresholds

From the result, when Similarity Threshold=20 the model make the optimal prediction for the cross domain (the accuracy starts to decrease when greater than 20). That has the accuracy with 0.658.

GCN method on cross domain prediction have a better performance compare with conventional machine learning method. And increasing sizes of T for training GCN model would not influence the accuracy a lot. The detailed analysis is provided in discussion part.

DISCUSSION

In discussion, we are mainly focus on the comparison between the accuracy in prediction of conventional machine learning methods with GCN method.

GCN method analysis with conventional methods

For GCN method for outlier detection, different from the convolution methods, the human vectors are processed by GCN before the K-means and DBSCAN algorithm. Compare with the original vector distribution, embedded vector distribution of GCN is relatively more uniform and even. A explanation to the phenomenon is that the embedded vector collect the relation between different vectors, which means the original information for the survey is superimposed a layer of chaotic relationship that represents the similarity between different people. The adjacent relationship GCN added for the vectors make nodes more inclined to be distributed.

For GCN method for single point prediction, GCN method doesn't perform as as good as we expect. The highest accuracy on the training set with training $T = 1000$ and Similarity Threshold reaches 0.5141. The model is not extremely good on predicting single point answer. Here we come up with a potential interpretation for the phenomenon that as for survey, the answer to a question is often affected by several previous choices. The result we want does not require overall global data. But in GCN processing, we include exceed and redundant data. The model prediction somewhat similar to overfitting. The structure that we really are interested are hidden due to invalid information. Hence the GCN method doesn't perform a good accuracy on single point prediction

For GCN method for Cross Domain Prediction, GCN method performs significantly better than the conventional statistic learning methods under the training size T=500 and Similarity Threshold=20, the accuracy achieves 65.8%. And the GCN method in Cross domain prediction is not sensitive to the training size T. That is because for the information of the graphic structure is inherent and can be represented by a small amount of the vectors. The GCN method introduce a better results as GCN take "connection" between the nodes into consideration for making prediction, which is ignored by conventional learning methods. Cross domain prediction rely on the relationship and structure of nodes on the larger picture, our GCN method extracts this structure from the similarities between people and people that fit for the cross domain prediction.

This perspective of view also inspires us that: when we focus on larger-scale features, local single-point predictions have limited correlation with global properties and are easily affected by dataset disturbances, like when people are doing the survey, they are more likely to be affected by the question he/she answered in previous. Therefore, GCN's performance on single point predictions is not better than traditional statistical learning. When we are interested in structural issues on a larger scale, GCN performs better, but for problem that are more dependent on local property, GCN might not performs as good as we expect.

Future work

In the future, we can further explore human-vector derived from social surveys by conducting a detailed study of different types of survey questions and responses from natural language perspective.

When calculating similarity between people, similarity of questions may be considered since different questions reveal a person's point of view from different perspectives. We may incorporate the similarities between survey questions in human-vector or in the graph definition to better represent a person's opinions in addition to current methods which mainly deal with responses.

We will also try to generalize our approaches to different questionnaires that contain multiple types of responses. In addition to the current survey with Likert scale ratings, there are also surveys with different kinds of responses, including numerical response, ordinal response (e.g. Likert scale) and nominal response. For numerical response and ordinal responses, we can re-scale them and calculate the similarity directly. For nominal responses, we may use state-of-art natural language processing approach to process and embed them.

CONCLUSION

In this work, we have designed three tasks to evaluate the effectiveness of using graph learning related methods in analysis of the survey data. For the outlier detection task, we have used K-means and person embedding method to separately calculate the Silhouette score and Calinski Harabasz score to give a quantitative reference. In the single point prediction task, we random chose the survey question to predict and the two approaches give similar result. Moreover, for the cross domain prediction task, the graph learning methods outperforms the convention method by around 10% in average. Due to the time and the resources constraint, the methods still have much room for the fine tuned either in parameters or frameworks used. We also observed the context of the survey questions would affect the prediction results. In the future work, more context-related features can be applied to the models to improve the task performance.

ACKNOWLEDGEMENT

We would like to express special thanks to Dr. Han ZHANG from social science department to provide us with the original data. The detailed contributions of our group are:

- Baichen YANG: coding on the implementation of the statistical machine learning models with Zeyu HUANG and writing corresponding parts in the report.
- Chengzhong LIU: organizing the report and the presentation and doing data preprocessing as well as graph learning methods coding.
- Zeyu HUANG: coding on the implementation of the statistical machine learning models with Baichen YANG and writing corresponding parts in the report.
- Zijun CHEN: coding on the GCN modeling, elaborating GCN principle, discussion part and writing corresponding parts in the report.

We would like also express great attitude towards the teaching team of COMP5212 in Spring 2021 who had paid so much efforts to facilitate our learning process.

REFERENCES

[1] R Michael Alvarez. 2016. *Computational social science.* Cambridge University Press.

[2] Maryam Amiri and Leyli Mohammad-Khanli. 2017. Survey on prediction models of applications for resources provisioning in cloud. *Journal of Network and Computer Applications* 82 (2017), 93–113.

[3] Tadeusz Caliński and Jerzy Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods* 3, 1 (1974), 1–27.

[4] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, and others. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Kdd*, Vol. 96. 226–231.

[5] Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *science* 315, 5814 (2007), 972–976.

[6] Lior Gideon. 2012. *Handbook of survey methodology for the social sciences.* Springer.

[7] Brian Hopkins and John Gordon Skellam. 1954. A new method for determining the type of distribution of plant individuals. *Annals of Botany* 18, 2 (1954), 213–227.

[8] Guolin Ke, Qi Meng, Thomas Finely, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems 30 (NIP 2017).*

[9] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[10] Sumeet Kumar and Kathleen M. Carley. 2018. People2Vec: Learning Latent Representations of Users Using Their Social-Media Activities. In *Social, Cultural, and Behavioral Modeling*, Robert Thomson, Christopher Dancy, Ayaz Hyder, and Halil Bisgin (Eds.). Springer International Publishing, Cham, 154–163.

[11] David Lazer, Alex Pentland, Lada Adamic, Sinan Aral, Albert-Laszlo Barabasi, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, and others. 2009. Social science. Computational social science. *Science (New York, NY)* 323, 5915 (2009), 721–723.

[12] S. Lloyd. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28, 2 (1982), 129–137. DOI: **http://dx.doi.org/10.1109/TIT.1982.1056489**

[13] James N. Morgan and John A. Sonquist. 1963. Problems in the Analysis of Survey Data, and a Proposal. *J. Amer. Statist. Assoc.* 58, 302 (1963), 415–434. DOI: **http://dx.doi.org/10.1080/01621459.1963.10500855**

[14] Jennifer Pan and Yiqing Xu. 2018. China's ideological spectrum. *The Journal of Politics* 80, 1 (2018), 254–273.

[15] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.

[16] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, and Xiuwen Yi. 2016. DNN-based prediction model for spatio-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 1–4.