Community Detection Using Graph Structure and Semantic Understanding of Text

Kartik Sawhney
Department of Computer Science
Stanford University
kartiks2@stanford.edu

Marcella Cindy Prasetio
Department of Computer Science
Stanford University
mcp21@stanford.edu

Suvadip Paul
Department of Computer Science
Stanford University
suvadip@stanford.edu

***Abstract*—With the increasing popularity of social networking sites, we see many users expressing their opinions online while simultaneously building new social circles, making community detection a prominent research topic in network analysis. Traditional approaches for community detection either use the network structure or some optimization metrics to detect communities as well as rate these communities. However, as real networks grow, the problem of community detection becomes more complex such that structural properties of the network might not be enough to uncover details about user interactions, and semantic information from text might provide important information as well. In this work, we use the Yelp dataset and combine information from the structure of the network with textual understanding. Our models show that using textual features brings the model closer in terms of performance with our structural approach baselines, while also successfully capturing latent information about users. We thus show the value in combining structure and semantic textual information as a robust and novel method for community detection.**

I. Introduction

Traditional approaches in community detection either use the network structure or optimization metrics to detect and rate communities, such as Girvan-Newman [5], Louvain model [4], modularity maximisation [6][12] and spectral clustering [7][2][1][17]. However, in todays social network we can represent a network as more than just nodes and edges. User profiles, users activity, and contextual information can be encoded as attributes. New techniques utilizing these rich features have led to significant improvements in community detection as we shall see throughout this paper and in related work.

Users can write opinions about services and products based on their experience. These sentiments expressed through opinions can change hidden network properties. Further, NLP allows us to determine broad topics from conversations which can provide interesting information beyond structural properties. Further, a community of users who are highly interactive might be either well connected (structural property) or constantly engaging with each other (language semantic property). To completely gauge how group dynamics change in an online communities forum, we thus must analyze both the network structure and language semantics throughout the group. We aim to do this in this work. We compare the differences in community detection using network structure and textual data. We then propose and evaluate a novel method of detecting communities as a mix of modularity maximization, sentiment minimization, spectral analysis and node embeddings. Furthermore, we use 2 methods to find node embeddings, one through semidefinite programming (SDP) relaxations [16] and the other through Node2vec [15]. We show how combining features leads to an NP-complete optimization problem and we try to solve this using techniques derived from spectral clustering via node embeddings solving it in poloynomial time.

II. Related Work

A. Structural Based Approach

A great deal of work throughout the years has been devoted to finding social circles in complex networks [14] from a structural perspective. The 2 most important measures which categorize and evaluate this problem are (1) more intra-cluster edges and (2) lesser inter-cluster edges. Edge-Betweenness as defined by Newman and Girvan [5], is the seminal concept which uses the properties stated above. They also defined a metric called Modularity to assess the quality of a particular cluster and tackling the problem of maximizing modularity is another feasible approach that can be taken for community detection, which is an (non-convex) NP-hard problem. Although easy to compute, modularity is difficult to optimize and known to perform badly for detecting small communities. However, despite this drawback, modularity-maximization approach tends to perform well in community detection for simple network.

Further, many structural based heuristics have been proposed and both the Louvain Method [4] and Infomap [8] are two examples. We will explain more about these two models in our 'Method' section and we will use both Louvain and Infomap as our potential baselines for performance comparison.

We notice that all the structural-based methods typically ignore the extra contextual features of nodes in the network, especially in today's complex social network. These extra features complement community detection algorithms and help find more precise and qualitative communities as shown in other recent work. In our combined model, we try to gauge whether a combination of modularity and other metrics derived from the textual data can help overcome this drawback.

B. User Profile Based Approach

Leskovec, J., & Mcauley, J. J. (2012) [11] shows how using similarity across different dimensions of the user profile (e.g. location, high school, major, etc.) on top of network structure

can bring improvement to the traditional community detection approach. Compared to approaches that utilize network structure or user profiles but not both, the approach performs significantly better. More importantly, the highlight of this work is showing how additional features from the user can bring out latent information from the communities.

C. Sentiment Based Approach

Xu et al. (2011) [10], discuss incorporating sentiments in community detection, to create sentiment based communities. They form the foundation stone of modelling such problems of not only using user properties, but also sentiments. However, once again they run into the problem of optimizing a non-convex NP-hard problem. Nonetheless, due to a simple model, they use a traditional rounding based SDP approach to solve this problem. Being different from [11], this approach does not combine the structural properties with sentiment score. Nevertheless, this work highlights the innovative approach that, digressing from structural properties of the network, attempts to model the community using additional features (in this case, textual sentiment features).

Even though the two approaches in [11] and [10] use different features for community detection, they give a notion that additional features besides network structure can bring improvement in the community detection performance and analysis, which is what we attempt to achieve in this work.

III. Data

We use users and reviews data from the Round 10 Yelp dataset challenge. The dataset provides a user-to-user social network with friendship links. The review data involves user reviews with ratings (on a 5-point scale). We filter the dataset to include only Yelp users since 2016. Figure 1 provides network statistics for the filtered social network.

We construct an undirected, unweighted user-to-user network from the filtered dataset. From Figure 2, the constructed network follows a power law distribution with a fitted maximum log likelihood estimate (MLLE) α of 1.31. Regarding the network structure, Figure 2 shows that most of the nodes are connected in one dense Strongly Connected Component (SCC), as most of them have high clustering coefficient with an overall average of 0.067595 and a diameter of length 24.

Nodes	35,408
Edges	66,817
Diameter	24
Average clustering coefficient	0.067595
α_{MLLE}	1.31
User reviews	1,033,124

Fig. 1: Filtered Yelp dataset (2016) statistics

We use 1,033,124 reviews (reviews from users on Yelp since 2016) to train our Latent Dirichlet Allocation (LDA) model (we discuss this model later in this paper). The average length of a review is around 17 words, and consists of multiple, often contradictory, sentences. For this reason, we cannot use conventional classification techniques (such as Naive Bayes) to extract sentiment and topics. Instead, we use word embeddings, using a combination of word2vec and tf-idf scores, to capture the semantic intent of the review.

We use topics from LDA and topic assignments for users as part of our model. Further, we use the sentiment score as provided by the dataset in our model as well. Although this dataset provides us true sentiment, in the form of average star score, we also train a sentiment analysis model using convolutional neural networks to check the effectiveness of our algorithm in the absence of true sentiment. Our generic sentiment analysis model has a fairly high accuracy (86%) which suggests that this approach can also be used without sentiment data.

IV. Definitions and evaluation metrics

In this section, we define important terms and evaluation metrics used in our work.

Modularity - Modularity (Q) [6] provides a measure of the network division strength into modules or communities and it is defined as

$$Q = \frac{1}{4|E|} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2|E|} \right] \delta_{c_i, c_j}$$

, where $|E|$ is the number of edges, k_i is the degree of node i, A_{ij} is an element of the adjacency matrix of the Graph, δ_{c_i, c_j} is the Kronecker delta symbol, and c_i is the community label for node i.

The map equation - The map equation [9] describes the quality of information or data flow in a network and follows the principle of Minimum Description Length. We will not go in-depth about the Map Equation in this work, but essentially, for a network of n nodes with optimal M modules (clusters), module codebook and index codebook for navigating between modules, the lower bound of the map equation is defined as:

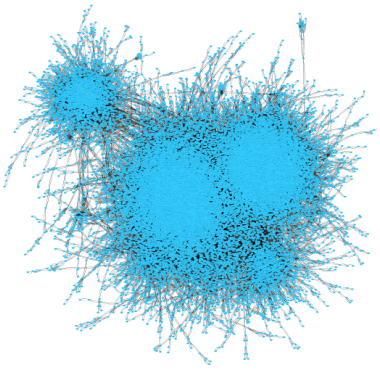
$$L(M) = q_{\rightarrow} H(Q) + \sum_{m \in M} p^m_{\leftarrow} H(P^m)$$

$q_{m\rightarrow}$: probability the walker exits module m; $q_{\rightarrow} = \sum_{m \in M} q_{m\rightarrow}$: probability the walker changing module; Q: probability distribution of $q_{m\rightarrow}$; $H(Q)$: average length entropy of codewords in the index codebook; p_α: probability of visiting node α; $p^m_{\leftarrow} = q_{m\rightarrow} + \sum_{\alpha \in m} p_\alpha$; P^m: probability distribution of p_α; $H(P^m)$: average length entropy of codewords in the module codebook m.

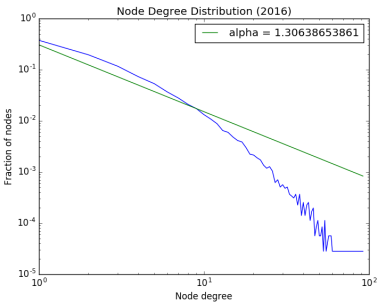
Silhouette Score - The score measures how similar a node u is to the nodes in its cluster, compared to nodes in other clusters. In other words, the silhouette score captures how appropriate the clustering assignment is for the given network. For each node u, the silhouette score $s(u)$:

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

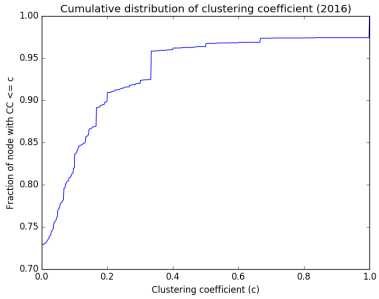
, where $a(u)$ is the average distance of node u to other nodes in its cluster and $b(u)$ is the smallest average distance of node u to other nodes in other clusters. The silhouette score ranges from -1 to 1, where a value close to 1 can be infered as node u being not similar to the nodes outside of its cluster (thus indicating good clustering). We will use the absolute differences between user sentiment scores as the distance function in our evaluation.



(a) Largest Strongly Connected Component



(b) Degree distribution follows power law



(c) Cumulative distribution for clustering coefficient

Fig. 2: Filtered Yelp dataset (2016) visualization

Adjusted Rand Index - The Adjusted Rand Index measures the similarity between two data. We use the same definition as in Emmons et al. (2016) [13], where given two community assignments X and Y, we count for each pair of node u and v:

N_{11} : u and v are in the same community in both X and Y
N_{00} : u and v are in different communities in both X and Y
N_{10} : u and v are in the same community in X but in different communities in Y
N_{01} : u and v are in different communities in X but in the same community in Y

$$ARI(X,Y) = \frac{2(N_{00}N_{11} - N_{01}N_{10})}{(N_{00}+N_{01})(N_{11}+N_{01}) + (N_{00}+N_{10})(N_{11}+N_{10})}$$

The value is in the range 0 and 1, where 1 indicates the community assignments X and Y have maximum agreement.

V. Method

A. Conventional Structural Approach

Since our data does not include any ground truth label for user communities, we explore two structural approaches as our potential baseline to compare how similar our proposed model performed to our baseline model. These approaches use network structure properties, such as edges and node degrees. We evaluate performance of each and choose the most appropriate model as our baseline.

Louvain Method - Louvain Method from Clauset-Newman-Moore [4] uses a greedy optimization method that maximises the modularity of a partition of the network. The optimization is done in two steps. The method looks for small communities by optimizing modularity locally (using the rounding method), and then creates a hierarchical sub community whose children are the two communities discovered in step one. This process is repeated until a maximum of modularity is attained. After evaluation, we decide to use the Louvain model as our primary baseline.

Infomap - Infomap by Rosvall, M., and Bergstrom, C. T. [8] uses the map equation as its greedy optimization function to detect communities or clusters. The smaller the map equation is, the better the module or community structure of the network such that movements in the network and important structures can be observed better. However, one unique properties of this approach is the massive number of modules detected. These modules are significantly smaller in size compared with the other approaches as we will see in our Evaluation.

B. Feature Mapping and Clustering

In addition to our baseline, we inspect variants of structural approaches as potential base models where we can inject textual features into the model.

Feature Mapping using Modularity - We model the problem of partitioning a set into two clusters, where each node in the cluster is represented by -1 or 1 depending on which cluster it has been assigned to. Let s be the vector which represents the clustering. It is of size n, and only has the elements -1 or 1. Now Modularity Q can be expressed as

$$\begin{aligned} Q &= \frac{1}{4|E|}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2|E|}\right](s_i s_j + 1) \\ &= \frac{1}{4|E|}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2|E|}\right](s_i s_j) \\ &= \frac{1}{4|E|}s^T B s \end{aligned}$$

, where B is the matrix with element $B_{ij} = A_{ij} - \frac{k_i k_j}{2|E|}$, and k_i represents the degree of each node i.

Note: Maximising Q in this form, with the constraint on s being $\|s_i\| = 1$ for all i, is a well known NP-complete problem and we will look at heuristics to solve this below.

One possible way of solving this is representing s as a linear combination of the normalized eigenvectors u of B, we have

$$\begin{aligned} s &= \sum_{i=1}^{V} a_i u_i \\ a_i &= u_i^T s \end{aligned}$$

As B is symmetric, it can be broken down into its complete eigen decomposition, helping write Q as

$$Q = \frac{1}{4|E|} \sum_{i=1} a_i^2 \lambda_i$$

, where λ_i is the corresponding eigenvalue.

The solution proposed by Newman [6], was to pick the highest eigenvalue and divide the nodes into 2 groups, and recurse on those two groups to get more clusters until modularity stops increasing.

In our project, we try to model the problem in one go. Our vector $S_{i,\,c} = 1$ represents if node i belongs to community c, and 0 otherwise. We can now write the modularity as

$$Q = \frac{1}{2|E|} Trace(S^T B S)$$

Using the same eigenvalue transformation, we have

$$Q = \frac{1}{2|E|} \sum_{i=1}^{V} \sum_{c \in C} \lambda_i (u_i^T s_c)^2$$

This now boils down to picking the eigenvectors corresponding to the largest possible eigenvalues as feature projections for each cluster. Clearly Q is maximised by choosing columns proportional to the eigenvectors. Thus obtaining the required communities is picking all the largest possible positive eigenvalues and projecting them in that space. However, only as the eigenvectors corresponding to the positive eigenvalues will contribute positively to the modularity, the (total number of positive eigenvalues + 1) is the upper bound for the number of clusters with this method. The other issue is the rounding of the eigenvalues to 0 and 1, and breaking ties. Newer SDP methods try to overcome this rounding. To counter this issue, we use the approach below, where we come up with node embeddings for each user/node and then from the concepts of spectral clustering, use a greedy clustering algorithm (kmeans/Expectation Maximization) to get the communities.

Spectral Clustering - As stated in spectral clustering literature, we want to project all the nodes into a subspace on which we can greedily cluster them into clusters. Based on the observation of [7][2][1][17], we try to get around the difficulty as stated above by using the vectors directly as feature projections of a particular node.

As seen in [7], the symmetry matrix can be seen as Q, and the projected eigenspace is the required feature space. This is like projecting along the first principal component of Q, and diving the clusters based on what side of 0 they belong to. Also as taken from [7], we run Expectation Maximization (EM) clustering on it, with k-mean++ initialization. We project each node onto 20/10 dimensions with 2 different node embeddings as described below. Then we add user profile information through sentiment and LDA to this embedding, following which we run EM to obtain 100 clusters for the dataset. As is evident, this is a mix of feature mapping (of each user into a new space) and spectral clustering. We run EM on this data with k=100 and we discuss the results in the'Results' section.

Feature Mapping using SDP - Given an undirected graph we construct a low dimensional embedding of each node in that graph. This embedding of each node can be represented as $x_i \in R^k$ where $k << n$. We use the method of Semi-Definite Programming (SDP) as mentioned in [16] to solve this.

To motivate this new embedding, we shall show the drawback of just using the principal eigenvector. As mentioned in [16], the above problem can be modelled as

$$argmax_{s \in -1,1} < s, Bs >$$

Which is a variation of the following SDP problem

$$\begin{aligned} &\text{Maximize } < X, Y > \\ &\text{subject to } X \succeq 0, X_{ii} = 1 \end{aligned}$$

Where $X = ss^T, Y = B$ and $< X, Y >= Trace(X, Y)$. By dropping the strict condition of $s \in -1, 1$, it is the problem as described above, which can be solved in polynomial time using the first principal component. This has the following drawbacks

1) If eigen value of the principal component $\lambda > 1$, then we have a nice solution which converges on iteration. On the other hand if $\lambda < 1$, no method can achieve a MSE smaller than 1 (other than the trivial $s = 0$ case).
2) PCA is efficient but does not exploit the fact that $s \in -1, 1$.

[16] have shown that their estimator does a near optimal job of solving the problem as stated below and we shall be using their model and see if we get better results.

They simplify the above problem to

$$\begin{aligned} \text{Maximize } &\sum_{(i,j) \in E} \sigma_i \sigma_j - \frac{\gamma}{2} \|M\|_2^2 \\ &\text{subject to } \|\sigma_1\| = 1 \end{aligned}$$

Where $M = \sum_{i=1}^{n}$ and $\sigma_i \in R^k$ is the required projection.

Feature Mapping using Node2Vec - As mentioned in [15], The node2vec model learns low-dimensional embeddings for users in a social network by using properties of random walks. Based on the paramenters of the model, it balances the exploration-exploitation tradeoff where nodes closer to the random walk need to have similar embeddings.

The walks are done using a combination of BFS and DFS. It is monitored by 2 parameters p and q.
- Return parameter, p controls the likelihood of immediately revisiting a node in the walk.
- In-out parameter, q allows the search to differentiate between inward and outward nodes.

Shown to be scalable and better than the usual spectral clustering embeddings, deepwalk and LINE, we decided to use this embedding as well. It has not yet been used for a rigorous analysis of community detection, however we try to do so in this project. As this is an algorithm in the Snap Library, we only describe it briefly here.

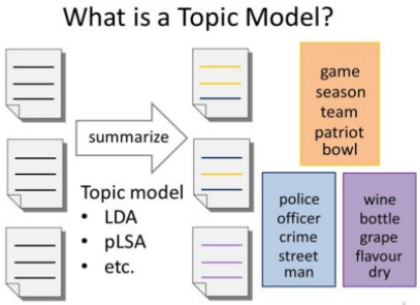


Fig. 3: LDA Topic Modeling

C. Textual Feature Based Approach

Data Preprocessing and getting word embeddings - We first perform the following pre-processing techniques on the review dataset: (1) Remove punctuation and standardize case; (2) Eliminate stopwords; (3) Lemmatize nouns and verbs. We then use a combinationf Glove vectors and TF/IDF to get word embeddings. We then combine these word embeddings into paragraph embeddings using a linear combination.

Sentiment analysis - the Yelp Dataset provides us sentiment, as stars, for reviews. We take the average of these sentiments as the overall sentiment for the user. To ensure that our approach also works for dataset without true sentiment data, we also train a sentiment model using CNN. We train the neural network for 50 epochs, and stop when the loss converges to 0.27. We use 80-20% cross-validation, and the model is able to achieve up to 86% accuracy. A manual evaluation of some reviews and sentiment suggest fairly good predictions, particularly on the shorter reviews. Given the high accuracy, we decide to use the provided sentiment score in our model.

LDA - To determine clusters using text alone, we use Latent Direchlet Allocation (2002) [3]. LDA is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. The algorithm considers a document to be a combination of topics, and topics to be composed of words. In this case, we treat reviews as documents composed of words. The algorithm then suggests that each document (review) is a mixture of a small number of topics and that each word's creation is attributable to one of the review's topics. Figure 3 captures this essence.

The input for the LDA algorithm is a linear combination of word embeddings obtained after data preprocessing (as discussed above). We run the Mallet implementation of LDA on all reviews written by users on Yelp since 2016 using 20 clusters and 10 words to describe each topic/cluster. We filter out common words and retain only nouns and verbs to give us informative topics.

The output of this algorithm is a probability over topics for each review. We tried two approaches for topic assignments for a review: (1) we take the Jaccard similarity between the contents of the review and the keywords representing the 20 topics, and assign the topic with the highest Jaccard similarity; (2) we take the probability of words in the review given the topic, combine them and use the topic with the highest probability as the topic for the review. We determine that approach (2) works better, and hence we use it for topic assignments. Further, to determine the topics for a user, we take the intersection of all topics for reviews written by the user.

LDA Quantification - For the top clusters obtained by structural properties, we analyze the topic distribution within a cluster, and compare it to the topic distribution given by the LDA assignments. We thus try to answer the question: given structural assignments, how effectively does it capture the things being discussed by people?

D. Feature Combination Approach

We propose 2 methods to combine features and rigorously test their advantages and disadvantages. The features to be combined are structural properties, average sentiment of users and the top LDA topics per user.

Concatenation of sentiment and LDA feature vectors to the node embeddings. This is done for both the PSD and node2vec embeddings. The sentiment values are represented as scaled and normalized values of the average sentiment per user. The LDA values are represented as scaled 1-hot values of the top topics a person has expressed their views about.

$$feature_{s_i} = scaling * (s_i - avg(s))$$
$$feature_{LDA_i} = scaling * (1 - hot(\text{top user topics}))$$

These values are scaled to different ranges and we've picked a scaling such that we dont disrupt the structural properties too much.

Directly combining the distances between the users and encoding it as edge weights between 2 users. Edge weights between 2 users is simply a norm between the extra feature values of sentiment and LDA between 2 users.

$$weight_s(i,j) = scaling(5 - \|s_j - s_j\|_1)$$

We pick 5 as that is the maximum value the sentiment can take as well we want people with similar sentiments to have larger weights. Similarly we have

$$weight_{LDA}(i,j) = scaling(21 - \|LDA_j - LDA_j\|_1)$$

We pick 21 as that is the maximum value the LDA difference can take as we want people with similar LDAs to have larger weights.
Now we run node2vec on this weighted graph and come up with node embeddings, followed by clustering on these values.

VI. Experiments and Evaluation

A. Experiments

Using the filtered dataset, we ran all of the models described in "Method". Our experiments can be divided into five steps:

1) Evaluate how our two potential traditional structural baselines perform in terms of modularity, silhouette score and the quality of the communities. From this result, we choose one model as our prominent baseline to compare with the other approaches.

2) Compare the suggested feature mapping and spectral clustering models without injecting any textual features to analyze how well and how different it detects communities compared to the baseline.
3) Detect communities or clusters based on topics from reviews extracted by the LDA model to (1) provide a distribution of topics in clusters of users and (2) extract the topic embedding based on this cluster assignment to be used as our textual feature in the combined models, in addition to user sentiment, which is available in the dataset.
4) Explore variants of techniques to inject and append textual features to the feature mapping process on the SDP mapping and node2vec.
5) Compare the best combined models with the chosen baseline model in terms of the evaluation metrics described in the next section.

B. Evaluation

We perform evaluation with respect to two aspects, quantitative and qualitative analysis. In our quantitative analysis, we use (1) modularity score to capture how well the communities are separated, (2) average silhouette score to evaluate how appropriate each node is assigned to its cluster, and (3) adjusted rand index to compare how different the cluster assignments between the models. In our qualitative analysis, we survey how the clusters are grouped in different models based on the network visualization and distribution of LDA topics in the clusters.

VII. Results and Discussion

For the sake of easier understanding and comparison, we show results of Algorithm1 with the SDP embeddings and Algorithm2 generating node2vec embeddings as seen in 4. Although both variations of the algorithm have been applied on node2vec as well, we want to differentiate the 2 techniques of Algorithm1 and Algorithm2 in its entirety.

A. Quantitative Analysis

General comparison - As it can be seen in Figure 4, the Louvain model performs best in terms of modularity, due to modularity being its objective function, as well as for silhouette score. This observation indicates that the Louvain's clusters have a strong division and strong similarity between members of each clusters in our dataset. The Infomap, on the other hand, has the worst modularity score out of all the models. This performance gap is caused by the sheer number of clusters the Infomap model detects (9K clusters) and the small size of these clusters, compared to the other models as we can observe in Figure 7. This makes it hard to do a meaningful comparison with the other models. Thus, we choose Louvain model as our primary baseline.

For the other models besides the baselines, the Node2Vec approach has the closest modularity score to Louvain, which means of all our models, the Node2Vec has the closest division strength to our baseline. In addition, as we inject sentiment feature to Node2Vec, the modularity score slightly increases. This is an interesting result as we see an improvement in structural properties by incorporating non structural ones.

Nonetheless, the Node2Vec models are not foremost in all aspects as in terms of silhouette score, they have the lowest score of all, indicating dissimilarity in sentiment between users in the same cluster. The Spectral clustering approach has a lead in this aspect, having a silhouette score close to Louvain which increases by adding sentiment feature to it, with the cost of modularity optimization as we can see in Figure 9.

SDP Variants - In Figure 9, the SDP variant models perform similarly on modularity score only when sentiment is added. When LDA topics are added as a feature, however, the modularity score drops significantly. SDP with LDA detects clusters that are well-spread over the network, with no significant structural properties. This might be a hyperparameter-tuning problem, as scaling the LDA embedding by 0.1 can slightly improve the modularity.

In terms of silhouette score, adding sentiment gives a significant boost and the closest score to the Louvain method, as we can observe in Figure 9. From this result, we choose SDP-Spectral Clusering with Sentiment as our combined model for our final comparison.

Node2Vec Variants - In Figure 10, we can observe that implementing scaling and sentiment can give a boost in performance for Node2Vec in silhouette score indicating a more similar cluster members. Both scaling factors 1 and 0.1 give a stabler silhouette score compared to the vanilla Node2Vec model.

In terms of LDA embeddings features variants, in Figure 10b, although all variants with LDA features have close modularity distribution with each other, the silhouette score shows a significant difference. Node2vec with a combination of LDA and sentiment features actually performs better than vanilla Node2Vec, resulting in a higher silhouette score. Within the Node2Vec variants, we choose both the Node2Vec with Sentiment scaled by 0.1 and Node2Vec with both Sentiment and LDA to be compared in our final comparison with the Louvain baseline below.

Textual + Structural Models Comparison - Lastly, we compare our baseline, Louvain model, with the three combined models: (1) SDP-Spectral clustering with sentiment, (2) Node2Vec with Sentiment and scaling = 0.1, and (3) Node2Vec with Sentiment and LDA embedding, due to the three models performing better compared to their vanilla models and other variants.

In Figure 11, we can observe that Louvain model is still the best out of the three other models. However, in relative comparison, the modularity distribution from Spectral Clustering and Node2Vec with Sentiment and scaling = 0.1 have a close similarity to Louvain, indicating that in terms of modularity, these two models perform just as well as the Louvain model, as we can see in 11a. Furthermore, notably, the SDP-Spectral Clustering model is the most stable out of the three combined models. While keeping a similar modularity distribution as the Louvain model, in Figure 11b, it has the closest silhouette score distribution relatively compared to the other two combined models.

The assignment similarity between these four models can be observed in Figure 13. The adjusted rand index show how similar two cluster assignments are as the value gets closer to 1.

Model		Modularity	Silhouette Score
Louvain		0.734972	0.643517
Infomap		0.312463	0.830456
SDP embeddings			
Algo1 - Sentiment Concatenation to SDP	Normal	0.430153	0.466081
	+ scaling = 0.1	0.387284	0.549870
Algo1 - LDA + Sentiment Concatenation to SDP	as 1-hot vector	0.073987	0.500768
	as 0.1-hot vector	0.208477	0.523771
	as 1-hot vector + Sentiment	0.073331	0.531366
Node2Vec			
Algo2 - Node2Vec with Sentiment	Normal	0.634303	0.220761
	+ scaling = 0.1	0.642011	0.254897
	+ scaling = 1	0.638632	0.267319
Algo2 - Node2Vec with LDA + Sentiment	as 1-hot vector	0.624339	0.344052
	as 0.1-hot vector	0.637974	0.203172
	as 1-hot vector + Sentiment	0.626154	0.337899

Fig. 4: Modularity and Silhouette Score for each model

	1	2	3	4	5
CMN	0,5	0,2,5	0,5,10,11,14	0,2,5	0,2
Infomap	0,2,5	0,2,5,8,10	0,2,5,10	0,2,5	0,2,5,10
SDP	0,5	0,2,5	0,5,10,11,14	0,2,5	0,2
SDP + Sentiment	0	0,2,5	0,5,10,11,14	0,2,5	0,2,5
Node2Vec	0,5	0,5	0,2,5	0,2	0,5
Node2Vec + Sentiment	0,5	0,5	0,2	0,2	0,5

Fig. 5: Top topics 'talked' by 10% of the users in the top 5 cluster

Topic Number	Topic Description
0	place,food,service,love,Great,staff,recommend,I've,favorite,time
1	pizza,food,place,order,it's,eat,price,lunch,cheese,time
2	recommend,service,experience,customer,work,time,staff,feel,care,job
3	chicken,food,sauce,salad,good.,rice,order,it.,meat,wasn't
4	don't,didn't,told,customer,I'm,place,time,rude,service,people
5	food,wait,order,time,service,minute,table,server,didn't,waitress
6	place,bar,beer,drink,night,food,music,fun,hour,selection
7	car,work,service,company,time,recommend,fix,day,move,job
8	hair,nail,time,salon,cut,massage,I've,color,place,job
9	store,shop,selection,price,dress,place,grocery,staff,love,purchase
10	food,time,place,order,service,experience,wait,star,review,customer
11	ice,cream,tea,chocolate,place,sweet,cake,taste,milk,dessert
12	burger,cheese,order,food,place,sandwich,fry,breakfast,chicken,taco
13	staff,dog,care,time,office,doctor,recommend,feel,class,work
14	place,food,sushi,restaurant,order,rice,soup,dish,chicken,noodle
15	place,coffee,location,area,work,spot,shop,lot,staff
16	restaurant,food,order,menu,dish,service,dinner,salad,meal,sauce
17	call,time,service,phone,customer,day,told,pay,manager,business
18	room,hotel,time,place,stay,fun,area,show,people,park
19	de,le,la,pour,est,en,une,trs,du,je
20	The rest

Fig. 6: Description of top topics we get from LDA

Model	Number of clusters
Louvain	2900
Infomap	9024
Feature Mapping-SDP	100
Node2Vec	100

Fig. 7: Number of clusters/communities detected by each model

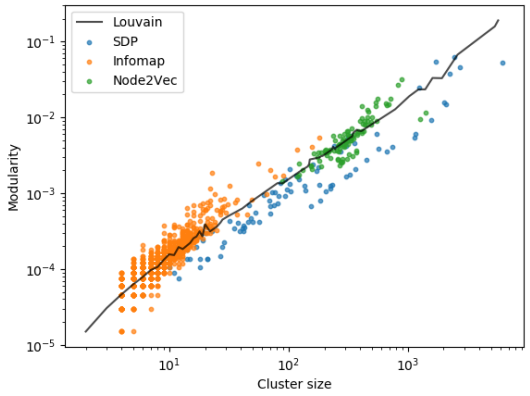


Fig. 8: Modularity comparison for structural approach and base proposed models

Overall, SDP-Spectral Clustering with sentiment has the most similarity with Louvain, with Node2Vec with both sentiment and LDA embedding coming up second. We are aware that the rand index scores are significantly small for the three models compared to the Louvain model as our baseline. However, it serves as a measure of how similar these combined models can perform to Louvain, despite the different approach they use.

From this result, we can infer that SDP-Spectral Clustering with Sentiment feature can give a comparable division distribution to Louvain with the drawback of slight dissimilarity within the members of its clusters.

B. *Qualitative Analysis*

Visual Observation - We observe how the combined models assign nodes into clusters in the dense SCC component of the network. We colorize the different clusters in the network, with color red as the biggest cluster. From the visualization in Figure 12, we observe that Louvain's biggest cluster occupies most of the SCC. Interestingly, SDP-Spectral Clustering with Sentiment has a similar assignment to Louvain. In Figure 12b, we can note that the clusters are mostly grouped together. On the other hand, the Node2Vec models have a more distributed cluster assignment in the SCC.

LDA topic distribution - As seen in 5, we see that in general what people are talking about in a particular cluster are the same things. We see that the top topics for CMN and SDP-Spectral Clustering are exactly the same. This is an interesting result leading us to ask if the top clusters across varied algorithms are the same in terms of topics people are talking about.
Node2vec does not capture the diversity in topics as well as inn the SDP Method. It seems to pick clusters consisting of users interested in the same things and we arent able to exploit any other information on top of it.
Further, interestingly, the LDA model does really well in identifying the kind of food people are talking about, besides differentiating food from salons and other non-dining services, and identifying sentiment to some extent.

- Kinds of Foods -
  ice,cream,tea,chocolate,place,sweet,cake,taste,milk,dessert
  burger,cheese,order,food,place,sandwich,fry,breakfast,taco
  place,food,sushi,restaurant,order,rice,soup,dish,chicken,noodle
- Service and Experience -
  place,food,service,love,Great,staff,recommend,favorite,time
  recommend,service,experience,customer,time,staff,feel,care
  call,time,service,phone,customer,day,told,pay,manager,business

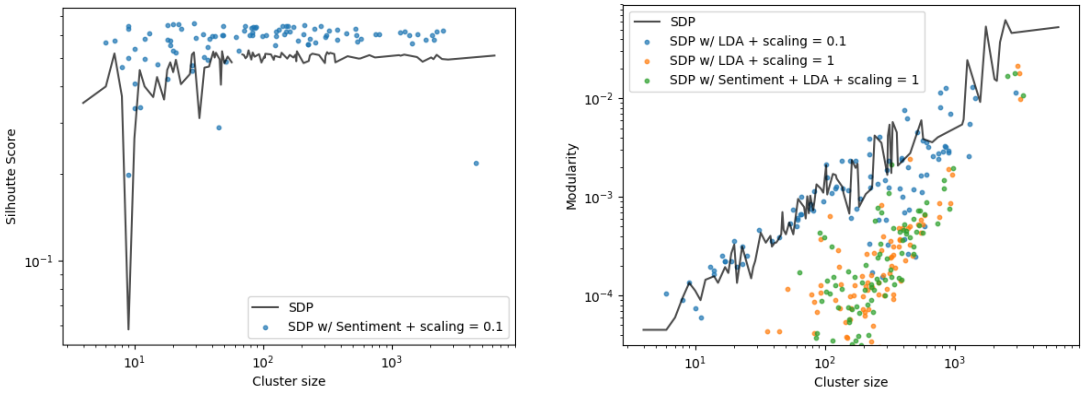
The biggest conclusion we can draw from this are the topics 0,2 and 5 being well-represented in the top clusters, as well as CNM and SDP-Clustering capturing similar information about its people. This means that the major social circles in yelp comprise of people who are recommending food, restaurants and other places as well as interacting with similar people on a regular basis.

VIII. Conclusion

We observe that our combined models didn't perform on-par with the traditional Louvain model in terms of modularity and silhouette score. However, from our experiments, we want to highlight how close each model managed to perform with the Louvain method. Although not optimal, the SDP-Spectral clustering with sentiment approach shows the most potential in our combined models. It manages to have a stable, albeit not the best, performance in terms of cluster division strength (modularity), similarity between cluster members (silhouette score), and cluster assignment similarity with Louvain (adjusted rand index). Regarding the quality of the clusters, SDP-spectral clustering manages to keep the structural aspect of the communities while introducing differences in how the clusters are grouped as we can see in the graph visualization. This finding itself points out the potential of feature combination between structural and user textual content in social network.

The evaluation of all our models have been quantitative over structural metrics. This is another drawback we need to address. Although we are performing better in terms of finding similar clusters with similar sentiments and topics, we are not able to match purely structural metrics. One possible change is to have a new metric which also evaluates some of the sentiment and LDA parameters. As this path has not been rigorously explored before, we have used the qualitative understanding of the clusters to highlight the identification of latent properties that are otherwise lost by considering structural properties alone.

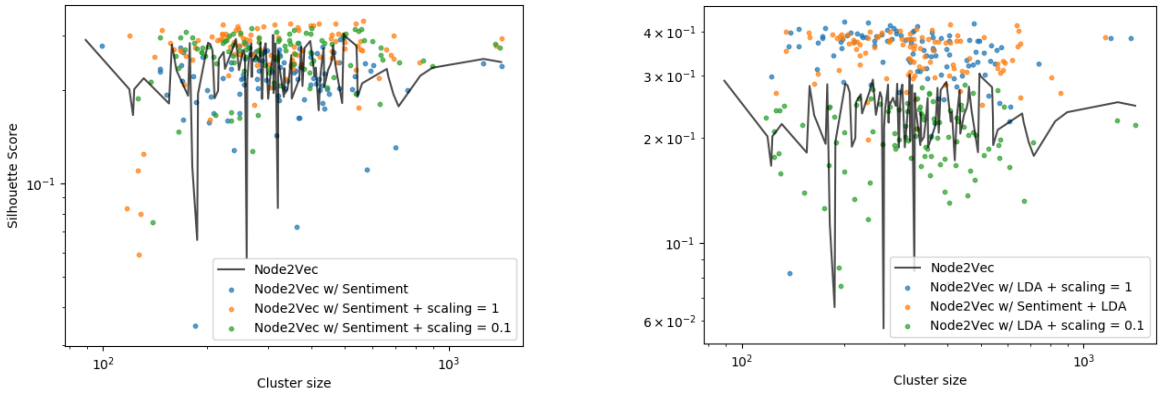
Further, we believe there are other variants on how we can combine textual with structural features in community detection that are yet to be explored, and can be the subject of future work. Furthermore, it will be interesting to see how these models perform under social network with ground truth label of communities in addition to comparing them to only



(a) Silhouette score for SDP with Sentiment variants

(b) Modularity for SDP with LDA variants

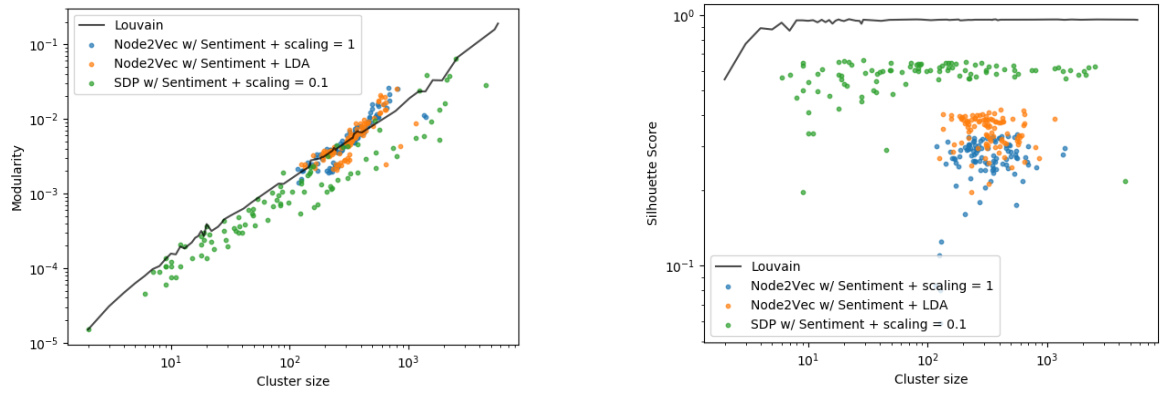
Fig. 9: Comparison between SDP variants



(a) Silhouette score for Node2Vec with Sentiment variants

(b) Silhouette score for Node2Vec with LDA variants

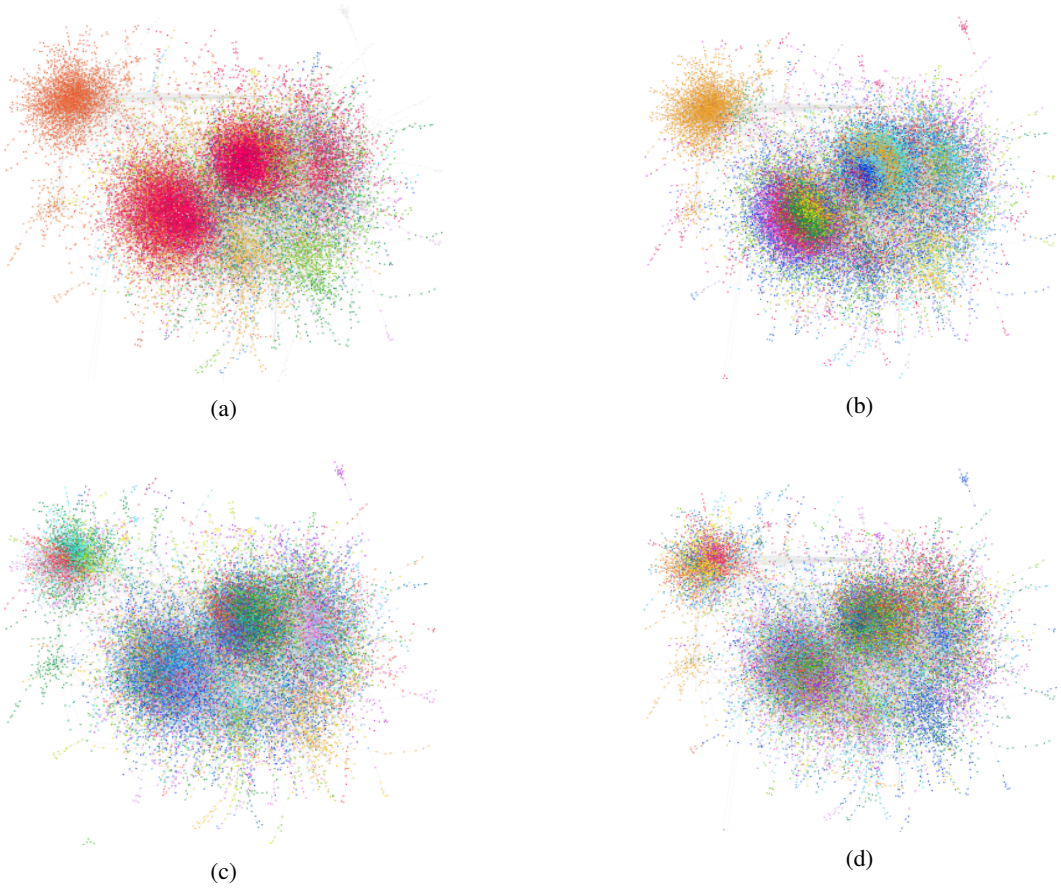
Fig. 10: Silhouette score comparisons for vanilla Node2Vec variants with Sentiment and LDA embedding features



(a) Modularity

(b) Silhouette score

Fig. 11: Modularity and Silhouette score comparisons for Louvain (baseline), SDP-Spectral Clustering + Sentiment, Node2Vec + Sentiment, and Node2Vec + Sentiment + LDA topics



(a) (b) (c) (d)

Fig. 12: Network Visualization comparison between (a) Louvain (baseline), (b) SDP + Sentiment (c) Node2Vec + Sentiment, (d) Node2Vec + Sentiment + LDA

	Louvain	SDP + Sentiment	Node2Vec + Sentiment	Node2Vec + Sentiment + LDA
Louvain	-	0.238485	0.095317	0.101943
SDP + Sentiment	0.238485	-	0.120538	0.106276
Node2Vec + Sentiment	0.100806	0.120538	-	0.258063
Node2Vec + Sentiment + LDA embedding	0.101943	0.106276	0.258063	-

Fig. 13: Adjusted Rand Index for SDP and Node2Vec approaches with Louvain

baseline model. This work can give a better insight on the prospective and contribution of textual features in community detection.

IX. Contributions

Equal contributions from all project members.

References

[1] Ravi Kannan, Santosh Vempala, and Adrian Vetta. "On clusterings: Good, bad and spectral". In: *J. ACM*. 2000.

[2] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. "On Spectral Clustering: Analysis and an algorithm". In: *NIPS*. 2001.

[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation". In: *Journal of Machine Learning Research*. 2003.

[4] Aaron Clauset, Mark E. J. Newman, and Cristopher Moore. "Finding community structure in very large networks." In: *Physical review. E, Statistical, nonlinear, and soft matter physics* 70 6 Pt 2 (2004), p. 066111.

[5] Mark E. J. Newman and Michelle Girvan. "Finding and evaluating community structure in networks." In: *Physical review. E, Statistical, nonlinear, and soft matter physics* 69 2 Pt 2 (2004), p. 026113.

[6] Mark E. J. Newman. "Modularity and community structure in networks." In: *Proceedings of the National*

Academy of Sciences of the United States of America 103 23 (2006), pp. 8577–82.

[7] Ulrike von Luxburg. “A tutorial on spectral clustering”. In: *Statistics and Computing* 17 (2007), pp. 395–416.

[8] Martin Rosvall and Carl T. Bergstrom. “Maps of random walks on complex networks reveal community structure.” In: *Proceedings of the National Academy of Sciences of the United States of America* 105 4 (2008), pp. 1118–23.

[9] M. Rosvall and Carl T. Bergstrom. “The map equation”. In: 2009.

[10] Kaiquan Xu, Jiexun Li, and Stephen Shaoyi Liao. “Sentiment community detection in social networks”. In: *iConference*. 2011.

[11] Julian J. McAuley and Jure Leskovec. “Learning to Discover Social Circles in Ego Networks”. In: *NIPS*. 2012.

[12] Mingming Chen, Konstantin Kuzmin, and Boleslaw K. Szymanski. “Community Detection via Maximization of Modularity and Its Variants”. In: *IEEE Transactions on Computational Social Systems* 1 (2014), pp. 46–65.

[13] Scott Emmons et al. “Analysis of Network Clustering Algorithms and Cluster Quality Metrics at Scale”. In: *PloS one*. 2016.

[14] Santo Fortunato and Darko Hric. “Community detection in networks: A user guide”. In: *CoRR* abs/1608.00163 (2016).

[15] Aditya Grover and Jure Leskovec. “node2vec: Scalable Feature Learning for Networks”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016.

[16] Adel Javanmard, Andrea Montanari, and Federico Ricci-Tersenghi. “Phase Transitions in Semidefinite Relaxations”. In: *Proceedings of the National Academy of Sciences of the United States of America* 113 16 (2016), E2218–23.

[17] Alex Pothen, Horst D. Simon, and K. P. Liu. “Partitioning Sparse Matrices with Eigenvectors of Graphs”. In: