AMATH 482 HOME WORK 4
CLASSIFYING POLITICIANS

LANG QIN

Amazing Department, University of Washington, Seattle, WA
`lq25@uw.edu`

Abstract. The goal of this work is to test the performance of spectral clustering and a simple semi-supervised regression algorithm on the 1984 house voting records data set. The data set consists of voting records of 435 members of the House on 16 bills. The goodness of the performance is measured by accurate classification of house members' affiliation.

1. Introduction and Overview

This work develops a semi-supervised classification model by linear regression with spectral clustering. The *1984 United States Congressional Voting Records Database* includes 17 attributes and consists of voting records of 435 members of the House on 16 bills. There are 267 members of the democratic party and 168 members of the republican party. The voting record of each house member on the 16 bills will be our input x while the corresponding output/class y is that members party affiliation (republican or democrat embedded as ± 1).

This work includes the following three tasks: 1) data prepossessing that gives feature matrix $\mathbf{X} \in \mathbb{R}^{435\times 16}$ and label vector $\mathbf{y} \in \mathbb{R}^{435\times 16}$, 2) spectral clustering with Gaussian weight that yields the unnormalized graph Laplacian matrix on $\mathbf{X}$ and the Fiedler vector $\mathbf{q}_1$, and 3) semi-supervised learning with linear regression over the Laplacian matrix which produces a party affiliation classifier. The performances of spectral clustering and semi-supervised learning are evaluated by corresponding accuracies: *clustering accuracy* and *semi-supervised accuracy*. In particular, the selection of the variance parameter σ for spectral clustering and Laplacian embedding parameters M and J are guaranteed to construct the optimal classifier.

2. Theoretical Background

To achieve the goal, this work primarily relies on the following three points: 1) spectral clustering and 2) semi-supervised learning.

2.1. Spectral Clustering.

Let $X = \{x_0, \ldots, x_{N-1}\} \in \mathbb{R}^d$ be a data set. The goal of spectral clustering is to find a feature map $F : \mathbb{R}^d \to \mathbb{R}^m$ so that the k-means method [1] on $F(X)$ lead to better clustering than X.

2.1.1. *Similarity Graphs.* Similarity graphs are to partition the graph into clusters. A similarity graph is defined as a weighted undirected graph $G = \{X, W\}$ where the $x_j \in \mathbb{R}^d$ are the vertices of G and the entries w_{ij} of W denote weights that are associated to edges that connect x_i to x_j. To construct a similarity graph over a data set, we take $w_{ij} = \eta(||x_i - x_j||_p)$, where η is a non-negative and continuous weight function $\eta : [0, \infty) \to [0, \infty)$. This work choose a Gaussian weight function

$$\eta(t) = \exp\left(-\frac{t^2}{2\sigma^2}\right) \text{ and } p = 2, \tag{1}$$

which leads to

$$w_{ij} = \exp\left(\frac{||x_i - x_j||_2^2}{2\sigma^2}\right). \tag{2}$$

Date: March 11, 2022.

2.1.2. *Unnormalized Graph Laplacian Matrix.* With the matrix W defined above, the graph Laplacian matrix L of of graph G (defined in Section 2.1.1) with the help of a diagonal matrix D, which is defined by

$$D_{pq} = \begin{cases} 0 & (p \neq q) \\ \sum_{j=0}^{N-1} w_{pj} & (p = q) \end{cases}. \tag{3}$$

Then, the unnormalized graph Laplacian matrix L is obtained by $L = D - W$. Notice that L is non-negative definite and symmetric matrix. Thus, L has non-negative real eigenvalues and real eigenvectors, and the eigendecomposition of L is

$$L = Q\Lambda Q^T, \tag{4}$$

where columns of Q are orthonormal eigenvectors of L and Λ is a diagonal matrix whose diagonal entries are eigenvalues of L. Notice that

$$L\mathbf{1} = (D - W)\mathbf{1} = \mathbf{d} - \mathbf{d} = \mathbf{0}. \tag{5}$$

Thus, $\lambda_0 = 0$ and $q_0 = \mathbf{1}$. In addition, the similarity graph W and graph Laplacian matrix L can be expressed as

$$W = \begin{pmatrix} W_1 & 0 \\ 0 & W_2 \end{pmatrix}, \ L = \begin{pmatrix} L_1 & 0 \\ 0 & L_2 \end{pmatrix}, \tag{6}$$

where the subscription distinguishes the cluster. (Note: the subscription of 1 and 2 is chosen because there are only two clusters in this work.)

2.1.3. *Laplacian Embedding.* Denote q_j as the normalized eigenvector corresponding to the j-th large eigenvalue of L (defined in Eq. 4). Then, the feature map F is defined as

$$F(x_j) = \begin{pmatrix} q_{1j} & q_{2j} & \dots & q_{Mj} \end{pmatrix}^T. \tag{7}$$

Notice that q_j is ordered in increasing order of λ_j, the eigenvalues of L, and the normalization of q_j is important. In addition, Eq. 4 gives an important fact that the first eigenvector is always $q_0 = \mathbf{1}$ and the second vector q_1 has the form

$$q_1 = \begin{pmatrix} \mathbf{1}_{n_1} \\ (-n_1/n_2)\mathbf{1}_{n_2} \end{pmatrix}, \tag{8}$$

where n_1, n_2 is length of $\mathbf{1}$ equals the number of rows of L_1, L_2 respectively. q_1 is known as the **Fiedler vector** and it is adjusted so that $q_0^T q_1 = 0$.

2.2. Semi-supervised Regression with Graph Laplacians.

2.2.1. *Semi-supervised Learning (SSL).* In SSL, one assumes that the data is available in the following term: 1) input $X = \{x_0, x_1, \dots, x_{N-1}\}$ and 2) some output puts $Y = \{y_0(x_0), y_1(x_1), \dots, y_{M_1}(x_{M_1})\}$, where $M \leq N$. The goal of SSL is to predict the outputs $\{y_M(x_M), \dots, y_{N-1}(x_{N-1)})\}$ given the labelled and unlabelled data sets.

2.2.2. *Semi-supervised Regression with Graph Laplacian Embedding.* This process is analogous to kernel regression, which has been discussed multiple times in previous works; i.e. we wish to find a function

$$f(x) = \sum_{j=0}^{J} c_j \phi_j(x) \tag{9}$$

such that $f(x_j) \approx y(x_j)$ for $M \leq j \leq N-1$. Applying graph Laplacians introduced in Section 2.1 in this process, we need to compute the first $K > 0$ eigenvectors $\{q_k\}_{k=0}^{K-1}$ of graph Laplacian matrix L and then solve the Ridge regression problem

$$\hat{c} = \underset{c \in \mathbb{R}^k}{\operatorname{argmin}} \sum_{j=0}^{M-1} |\sum_{k=0}^{K-1} c_k q_{jk} - y_j|^2 + \lambda ||c||_2^2. \tag{10}$$

Note: one may easily set $\lambda = 0$ to implement linear regression.

3. Algorithm Implementation and Development

The implementation of this project can be divided into three parts: 1) Environment Setup, 2) Spectral Clustering, and 3) Semi-Supervised Learning. This programming work is done by Google Colab. The `Python` libraries imported for this work include: `google.colab.drive`, `numpy`, `matplotlib.pyplot`, `scipy`, and `sklearn`.

3.1. Environment Setup.

Import data from `house-votes-84.data` and construct the input vectors x_j corresponding to the voting records of each member by replacing 'y' votes with $+1$, 'n' votes with -1 and '?' with 0. This gives a vector $\mathbf{y} \in \mathbb{R}^{435}$ as labels and input matrix $X \in \mathbb{R}^{435\times 16}$ as features in `sklearn` convention.

3.2. Spectral Clustering.

This step is to perform spectral clustering over the data sets and distinguish the data instances into two clusters. The optimal σ of the weight funciton $\eta(t)$ defined by Eq. 1 is determined to be the one with the maximum clustering accuracy for $\sigma \in (0, 4]$, which is defined as

$$\texttt{clustering accuracy} = 1 - \frac{1}{435} \times \texttt{number of misclassified members}. \tag{11}$$

The clustering accuracy is computed according to the algorithm below. For $\sigma \in (0, 4]$ with increment 0f 0.01,

i. Construct the similarity graph weight matrix W using Gaussian weight function $\eta(t)$ defined as Eq. 1 by methods described in Section 2.1.1;
ii. Compute the unnormalized graph Laplacian matrix L via methods described in Section 2.1.2 using previously compted W;
iii. Derive the the Fiedler vector q_1 defined as Eq. 8 from the computed L following the the methods described in Section 2.1.2, and adjust the entries of q_1 to 1 if it is positive or -1 otherwise;
iv. Use the sign of q_1 obtained in previous step as clustering result and calculate the corresponding clustering accuracy defined by Eq. 11. **Note:** Since there are only two clusters, the accuracy can not be lower than 0.5, i.e. the clustering is completely wrong. Thus, if the accuracy for a particular σ is less than 0.5, then adjust the accuracy by 1 - itself.
v. Set σ^* to be the one that corresponding to the first maximum accuracy.

3.3. Semi-Supervised Learning.

This step is to develop semi-supervised linear regression model with graph Laplacians, using the best spectral clustering obtained in Section 3.2. In order to tune the regressor, two hyper-parameters need to be decided: 1) the number of columns of eigenvectors for Laplacian embedding: $M = 2, 3, 4, 5, 6$ and 2) the number training instances for training: $J = 5, 10, 20, 40$. The optimal parameter set (M^*, j^*) is chosen by comparing the semi-supervised learning accuracy (SSL accuracy)(the higher the better), which is defined as

$$\texttt{SSL accuracy} = 1 - \frac{1}{435} \times \texttt{number of misclassified members}. \tag{12}$$

The SSL accuracy is computed according to the algorithm as the following. For each pair of parameters (M, J),

i. Take the Laplacian embedding using the first M eigenvectors $F(X) \in \mathbb{R}^{435\times M}$;
ii. Extract the submatrix $A \in \mathbb{R}^{J\times M}$ and vector $\mathbf{b} \in \mathbb{R}^J$ consisting of the first J rows of $F(X)$ and label vector $\mathbf{y}$;
iii. Use linear regression (least squares) to find

$$\hat{\beta} = \underset{\beta\in\mathbb{R}^M}{\operatorname{argmin}} ||A\beta - \mathbf{b}||_2^2, \tag{13}$$

and take $\hat{\mathbf{y}} = \operatorname{sign}(F(X)\hat{\beta})$ as the predictor of the classes of all instances in X.
iv. Compute the SSL accuracy using defined in Eq. 12;
v. Set the set (M, J) corresponding to the maximum SSL accuracy as optimal parameters (M^*, J^*).

4. Computational Results

4.1. **Spectral Clustering.** The optimal parameter for spectral clustering is $\sigma^* = 1.17$ with clustering accuracy of 0.88. The plot of accuracy as a function of σ is shown as Figure 1. From Figure 1, the clustering accuracy increase fast from 0.5 to 0.84 as σ increase from 0.1 to 0.5, and tend to stabilize at around 0.88. For $\sigma \in [1, 4]$. Interestingly, there is not only one choice of σ that gives the maximum clustering accuracy of 0.88045977. Here, we choose the first/smallest σ corresponding to the max accuracy as the optimal parameter σ^*. The non-uniqueness of the potential optimal σ-s shows that the goodness of spectral clustering is insensitive to the value of the variance parameter σ in weight function $\eta(t)$ defined by Eq. 1.

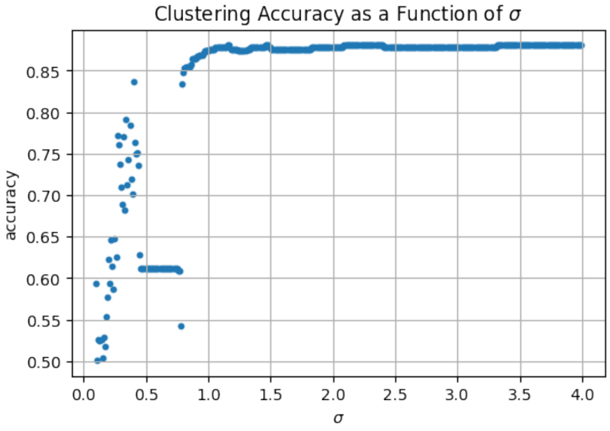

Figure 1. Clustering accuracy plot as a function of σ

4.2. **Semi-Supervised Regression with Graph Laplacians.** The semi-supervised accuracy obtained by Section 3.3 is listed in Table 1. From this table, one may observe that the only maximum SSL accuracy is 0.8896551 at $(M, J) = (2, 5)$. Thus, the optimal parameters for semi-supervised regression described in Section 3.3 is $(M^*, J^*) = (2, 5)$

	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 6$
$J = 5$	0.88965517	0.88735632	0.84137931	0.86896552	0.86896552
$J = 10$	0.88735632	0.81609195	0.85057471	0.70114943	0.68965517
$J = 20$	0.88275862	0.82298851	0.86436782	0.84137931	0.87586207
$J = 40$	0.88045977	0.83908046	0.87586207	0.88045977	0.86436782

Table 1. SSL accuracy for all possible combination of (M, J) pairs.

According to Table 1, the SSL accuracies tend to decrease as M increase, while changes in J could be both increase and decrease in SSL accuracies. The changes in SSL accuracies imply that the model is sensative to M instead of J. This is because the Fiedler vector introduced in Section 2.1.3 contains the most useful and distinct information about how to cluster the input data. The additional information brought by increasing M may contain false or ambiguous information which cause the SSL accuracy to decline.

The plot of the optimal regression output is shown as Figure 2, which confirms that the model performs well with some unexceptionable errors.

5. Summary and Conclusions

Through spectral clustering and semi-supervised linear (least squares) regressor with graph Laplacian embedding, this work successfully develops a classifier that cluster politicians into democrat and republican given information in data set *1984 United States Congressional Voting Records Database*. By comparing the clustering accuracies obtained from Section 3.2 and SSL accuracies from Section 3.3, this work concludes two points: 1) the choose of variance parameter σ in Gaussian weight $\eta(t)$ (Eq. 1) does not significantly influence

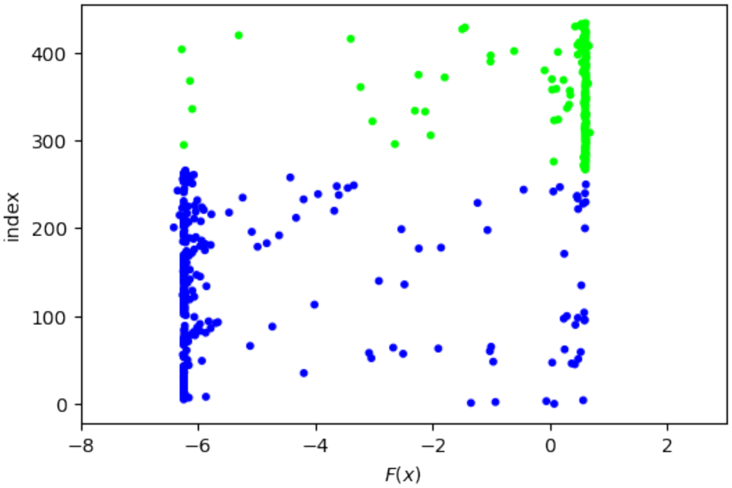

Figure 2. SSL accuracy plot with optimal parameter $M^* = 2, J^* = 5$

the performance of clustering; 2) Fielder vector obtained in Laplacian embedding as in Section 2.1.3 plays an important role in regression by providing distinct and useful clustering information.

Acknowledgements

The author is thankful to Professor Bamdad Hosseini's detailed course notes and sample works.

References

[1] K. Krishna and M. Narasimha Murty. Genetic K-means algorithm. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 29(3):433–439, June 1999.