CLASSIFYING POLITICIANS

XUECHENG LIU

Department of Applied Mathematics, University of Washington, Seattle, WA
xl0306@uw.edu

Abstract. Unsupervised learning is a broad topic in machine learning where we aim to divide the input data X into meaningful groups. Spectral clustering takes advantage of Kernel idea and performs clustering on the feature matrix of input data X. Semi-supervised learning combines features of both supervised and unsupervised in which it performs clustering first and assign the labels to each cluster based on the limited labels.

1. Introduction and Overview

This project develops a spectral clustering model and a semi-supervised learning model to classify politicians into either democratic party or republican party based on the voting of 16 bills. In spectral clustering, graph Laplacian matrix is introduced to extract the features of the input data in order to break the limitations the of the clustering algorithm where most of them are based on distance in Euclidean space. Clustering task is performed based on the sign of first non-zero eigenvector (Fiedler vector). Semi-supervised regression is conducted in a similar manor where features are extracted with graph Laplacian embeddings. Results of spectral clustering are compared across different choices of σ in the weight function and results of semi-supervised regression are compared across number of known labels and eigenvectors.

2. Theoretical Background

We first briefly discuss Graph Laplacian and how it is constructed from data set.

Consider the data set X = $\{x_0,...x_{N-1}\} \in \mathbb{R}^d$ and matrix W $\in \mathbb{R}^{N\times N}$ with $w_{ij} \geq 0$, where w_{ij} represents the connections between point x_i and x_j. Let η, a non-negative continuous function be the weight function and define $w_{ij} = \eta(\|x_i - x_j\|^2)$. For example, let η (t) = $\exp(-\frac{t^2}{2\sigma^2})$ which leads to $w_{ij} = exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$.

Define the degree vector d$\in \mathbb{R}^N$ where $d_j = \sum_{i=0}^{N-1} w_{ji}$ and diagonal degree matrix D = diag(d). Define the unnormalized graph laplacian $\widetilde{L}$ = D-W. Eigendecompose $\widetilde{L}$ = QΛQ^T and q_1 be the second eigenvector and sign(q_1) is the classifier for different clusters.

For semi-supervised regression, we only have access to M labels where M < N. Let Q$^{'} \in \mathbb{R}^{M\times J}$ be the upper left submatrix of Q defined above. According to [Brunton and Kutz, 2019], in least square regression, $\hat{\beta}$ is defined as

$$\hat{\beta} = \underset{\beta\in\mathbb{R}^J}{\text{argmin}}\|Q^{'}\beta - \mathbf{b}\|^2$$

where $\mathbf{b} \in R^M$ is the labels we have. The predicted values $\hat{y}$ = sign(Q$_j\beta$) where Q$_j$ is the first j columns of matrix Q.

In both settings, true labels are used to measure the accuracy of both models after the predictions. The accuracy is defined to be $\frac{1}{N}\times$ number of correctly classified members.

Date: March 9, 2022.

3. Algorithm Implementation and Development

The main packages used in this project is **Numpy** and **Sklearn**. **Scipy** is also used to simplify the calculation of distance in Euclidean space.

The following algorithm illustrate how to perform spectral clustering over a range of σ and pick the optimal one.

Algorithm 1 Spectral Clustering

```
$X \leftarrow data$
$y \leftarrow label$
$r\_space \leftarrow np.linspace(0, 4, 101)$
$dist \leftarrow scipy.distance\_matrix(X, X, 2)$
$accList \leftarrow [\,]$
for r in r_space do
    $W \leftarrow \eta(dist, r)$
    $d \leftarrow sum(W, axis = 1)$
    $D \leftarrow diag(d)$
    $L \leftarrow D - W$
    $V \leftarrow np.eigh(L)$
    $predictions \leftarrow sign(V[:, 1])$
    $acc.append(max(sum(y == predictions), sum(y! = predictions))/435)$
end for
$\sigma^\star \leftarrow r\_space[acc.index(max(acc))]$
```

The following algorithm illustrates how to semi-supervised regression is conducted across different values of seen lables(M) and eigenvectors(J) followed from the previous algorithm.

Algorithm 2 Semi-Supervised Regression

```
for M in [5,10,20,40] do
    $Y \leftarrow y[0 : M]$
    for J in [2,3,4,5,6] do
        $W \leftarrow \eta(dist, r)$
        $d \leftarrow sum(W, axis = 1)$
        $D \leftarrow diag(d)$
        $L \leftarrow D - W$
        $V \leftarrow np.eigh(L)$
        $Vp \leftarrow V[0 : M, 0 : J]$
        $leastSquare.fit(Vp, Y)$
        $predictions \leftarrow sign(V[:, 0 : K] \cdot leastSquare.coef)$
        $acc.append(max(sum(Y == predictions), sum(Y! = predictions))/435)$
    end for
end for
```

4. Computational Results

Figure 1 shows classification accuracy verse different σ values for both normalized and unnormalized graph Laplacians.

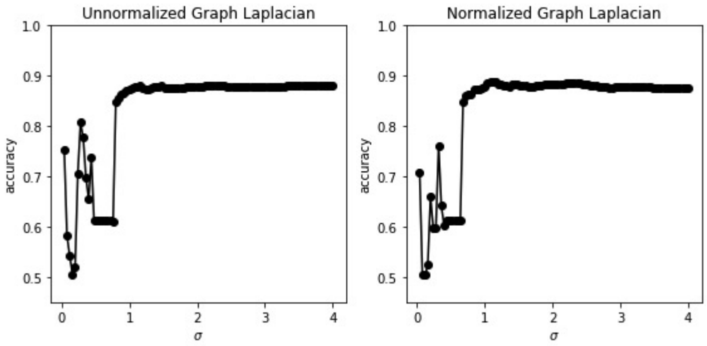
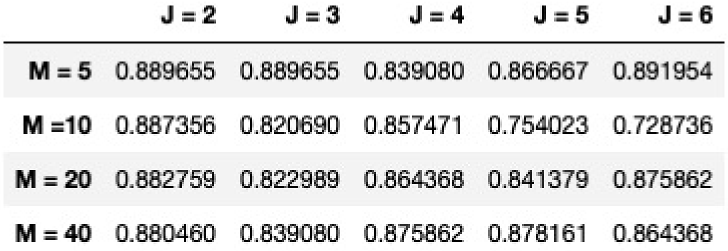


Figure 1. Accuracy v.s σ in both graph Laplacians

As we could conclude from figure 1, the classification accuracy in both graph Laplacians follow similar patterns where there exists a σ value that accompanying a sudden jump in classification accuracy. In addition, there are multiple *sigma* values where we have the same classification accuracy. We choose to pick the smallest one as $\sigma^\star$ among these values since bigger σ tends to loose the connection among pair of points in which features cannot be extracted properly.

Figure 2 shows the classification accuracy of semi-supervised regression across different number of seen labels(M) and eigenvectors(J) with $\sigma^\star = 1.16$ found in Figure 1.



	J = 2	J = 3	J = 4	J = 5	J = 6
M = 5	0.889655	0.889655	0.839080	0.866667	0.891954
M =10	0.887356	0.820690	0.857471	0.754023	0.728736
M = 20	0.882759	0.822989	0.864368	0.841379	0.875862
M = 40	0.880460	0.839080	0.875862	0.878161	0.864368

Figure 2. Accuracy for different pairs of M and J with $\sigma^\star$

As we could see from Figure 2, adding the seen label does not significantly improve the classification accuracy and generally speaking incorporating more eigenvectors does not improve the classification accuracy either.

5. Summary and Conclusions

In summary, graph Laplacian plays a vital role in both spectral clustering and semi-supervised learning tasks. The first few eigenvectors of the graph Laplacian have nice geometric interpretation and does a decent job in classifying by examining signs of each entry. Further work could be researched on why incorporating more eignevectors in training does not imply more features and lead to a better classification in the semi-supervised learning setting.

Acknowledgements

The author is thankful to Prof. Bamdad Hosseini for providing excellent lectures and well-formatted lecture notes as well as providing clean start code.

References

[Brunton and Kutz, 2019] Brunton, S. and Kutz, J. (2019). *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press.