AMATH 482/582: HOMEWORK 4

SATHVIK CHINTA

Abstract. Using clustering methods and semi-supervised learning on voting records, we are able to predict the party of a politician to a very high degree of accuracy!

1. Introduction and Overview

Let's see if we can predict the party of a politician based on their voting records! This is a very exciting task that has real world impications. We will use spectral clustering and semi-supervised learning on the 1984 house voting records data to see how accurately we can predict the party of a politician.

2. Theoretical Background

We will use spectral clustering in the beginning to see if we can get reliable results. Spectral clustering is a method for deeply rooted in graph theory. In our case, each of the data points would get treated as a node in our graph. In order to perform spectral clustering, we will use a Laplacian matrix on our set of points. A laplacian matrix can be defined as

$$L_{i,j} = \begin{cases} \deg(v_i) \text{ if } i = j \\ -1 \text{ if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 \text{ otherwise} \end{cases}$$

Equivalently, we can define two matrices: the degree matrix (D) and the adjacency matrix (A) and define the laplacian matrix with the equation:

$$L = D - A$$

We can then check the closely connected nodes which, in theory, should be closely related to each other as well. The components of the eigenvectors that correspond to the smallest eigenvalues of our constructed laplacian matrix, then, can be used to find clusters in our graph. The laplacian matrix will have different values depending on which weight function we choose, as you can imagine. We wil use the following weight function:

$$\eta(t) = \exp(-\frac{t^2}{2\sigma^2})$$

The second eigenvalue of the laplacian matrix is called the Fiedler value. The magnitude of this reflects how connected the graph is as a whole. We can select the Fiedler vector (the second eigenvector) and use the components whose value is positive to signify us predicting the +1 case (either democrat or republican) and the value who is negative to signify us predicting the -1 case (the other party). If we get an accuracy below 50%, this means that our Fiedler vector does not agree with our labelling of the data (ie. the Fiedler vector represents negative as Republican but we encoded the data as Democrat). In this case, we just flip the accuracy of the spectral clustering method (since we are basically in a binary classification problem). Anywhere we would

Date: March 11, 2022.

have predicted one party, we would now predict the other party. We will search through a range of differnet σ values in order to find the one that gives us the most accurate classification.

We will also attempt to use linear regression using the eigenvectors of our optimal laplacian matrix (the one associated with the sigma value that results in the most accurate classification). Upon constructing this matrix, we will expose the linear regression function to diffrent sub-matrices of our overall matrix. We will supervise how the accuracy of our linear regression function changes with the size of sub-matrices we use. We will take the sing of the resulting value as well, and use that to predict the party. Then, we can find the optimal number of rows and columns to use in our linear regression matrix.

This is a semi-supervised learning approach. This is a hybrid approach between supervised learning and clustering methods (like we used earlier).

3. Algorithm Implementation and Development

I used python along with numpy, matplotlib, and sklearn for this project. I read in the data using np.genfromtxt() and split the data into X (matrix of votes) and Y (party). I encoded the Y values with 1 if democrat and -1 if republican (and 0 if unknown). I then used the sklearn package to calculate the distance matrix (passing in X twice and p = 2 as my parameters). I used numpy to calculate the laplacian matrix, and eigenvalues/eigenvectors. Using the sign of the Feidler vector, I then predicted the party of each data point.

For the linear regression, I used the sklearn linear model package. Specifically, I used Ridge Regression with a very small alpha (which is mathematically similar to linear regression). Then, I took the coefficeints resulting after fitting my sub-matrix, and used the sign of the coefficients to predict the party of each data point.

4. Computational Results

Plotting the sigma values (1000 values between 0 and 4) against accuracy for spectral clustering yields the following plot:

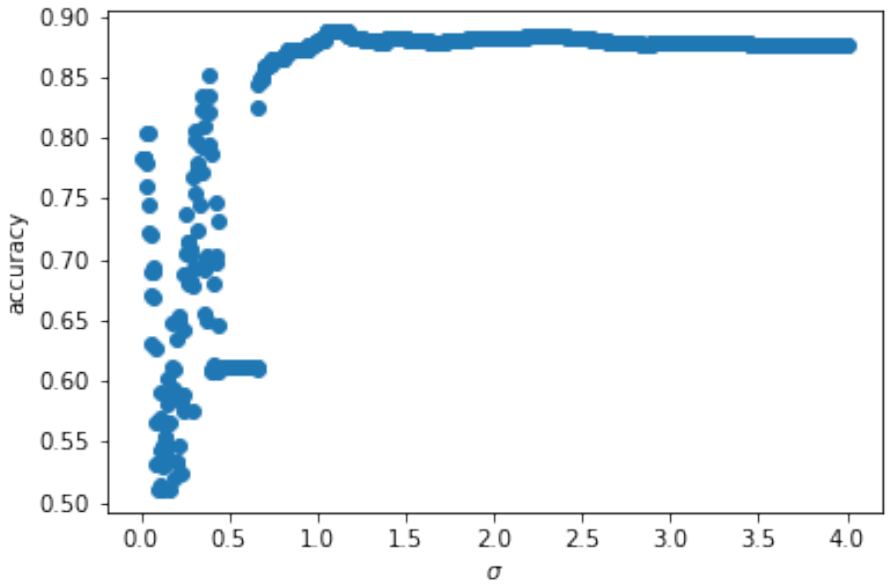

Figure 1. Plot of sigma values against accuracy for spectral clustering

We can see that the accuracy of the spectral clustering increases drastically after about $\sigma = 0.75$. Until that point, the accuracy seems to fall drastically at the smaller sigma values, then increase as well. After $\sigma = 0.75$, the accuracy seems to stabilize in the high 0.8's. The maximum accuracy occurs when $\sigma = 1.\overline{053}$. The accuracy at this point is 0.887, which indicates that even though our spectral clustering seems to be unstable, we are still getting a good accuracy.

When using the semi-supervised learning technique, we get the following results for different values of M and J.

J \ M	2	3	4	5	6
5	0.8827	0.8827	0.8620	0.8827	0.8827
10	0.8781	0.8919	0.8873	0.8873	0.8160
20	0.8850	0.8873	0.8988	0.8735	0.7816
40	0.8850	0.8942	0.9011	0.9264	0.9310

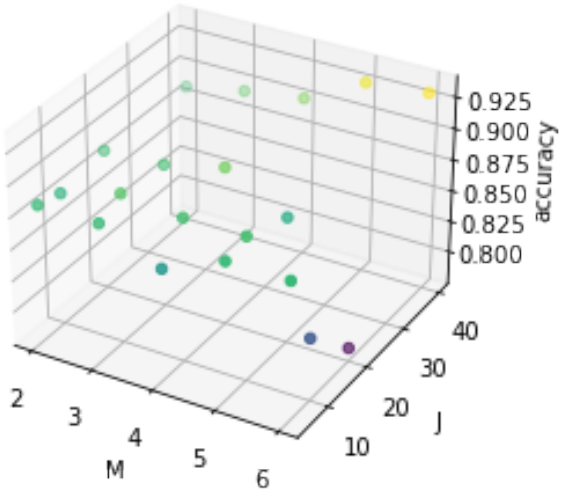

FIGURE 2. Plot of M and J values against accuracy for semi-supervised learning

We can see that the accuracy of the semi-supervised learning algorithm is almost as good, if not better, than that of the spectral clustering algorithm in most cases. The best accuracy occurs when $M = 6$ and $J = 40$. At this point, the accuracy is 0.9310. Curiously, the lowest accuracy occurs when $M = 5$ as well (though $J = 20$ in this case). This indicates we might be seeing behavior similar to that of spectral clustering, where even the slightest variation in our hyperparameters can lead to wildly different results. To test this theory, I plotted many values of M and J against eachother in order to see if there is any correlation between the accuracy and the values of M and J. The plot below shows the accuracy of the semi-supervised learning algorithm in this case

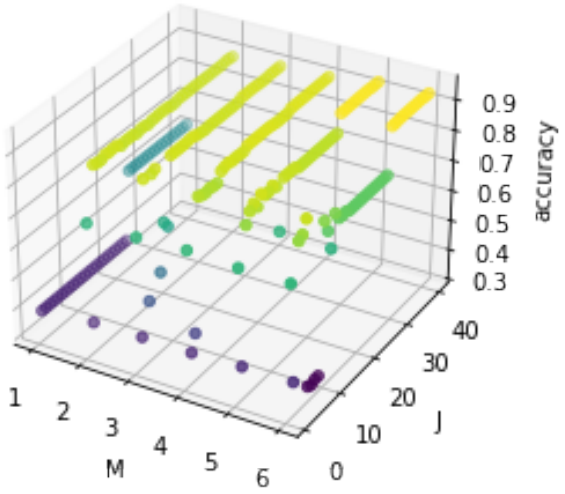

FIGURE 3. Plot of M and J values against accuracy for semi-supervised learning for a wide range of points

We can see that our hypothesis is almost correct. There does seem to be wild drifts in the accuracy, but not as erratic as we saw in the spectral clustering case. Instead, these seem to be linked to a drop of directly correlated to the J values. As J increases by a factor of 10, it seems that accuracy increases as well. As such, this methodology does seem to be much more stable (no more rapid oscillation between varying accuracies).

5. Summary and Conclusions

We can see that both methods performed very well. The spectral clustering algorithm ended up with an accuracy north of 88%. Very accurate! The semi-supervised learning method performed even better, with an accuracy of above 93%! This shows that we are able to succesfully predict the party of politicians based on their voting records.

Acknowledgements

I am thankful to Professor Hosseini for introducing us to the concept of graph laplacians, spectral clustering, and different methods of semi-supervised learning.

I am very thankful to my peers taking the class alongside me, they have helped me understand the material as well as provide a reference to compare my results against. I interacted with them both through Canvas discussion boards as well as Discord chat.

References

[1] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020.
[2] B. Hosseini. Introduction to clustering. University of Washington (LOW 216), Feb 2022. AMATH 482/582.
[3] B. Hosseini. Introduction to graph laplacians. University of Washington (LOW 216), Feb 2022. AMATH 482/582.
[4] B. Hosseini. Semi-supervised learning. University of Washington (LOW 216), Feb 2022. AMATH 482/582.

[5] B. Hosseini. Ssl demo. University of Washington (LOW 216), Feb 2022. AMATH 482/582.
[6] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.
[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[6] [1] [4] [5] [3] [2] [7]