AMATH 482/582: HOME WORK 4

YIXUAN LIU

University of Washington, Seattle, WA
yixuanl@uw.edu

Abstract. In the project, we will use the 1984 house voting records data set to test the performance of spectral clustering and a simple semi-supervised regression algorithm. By optimizing a parameter in spectral clustering to get a maximum clustering accuracy, we use the optimal parameter to investigate the accuracy of the simple semi-supervised regression algorithm with another pair of parameters.

1. Introduction and Overview

The 1984 house voting records data set consists of 435 members of the House's voting records on 16 bills. There are 276 members of the democratic party and 168 members of the republican party. We distinguish them with two different labels $\{-1, +1\}$ to get a output vector $y \in \mathbb{R}^{435}$ and reconstruct the input matrix $X \in \mathbb{R}^{435\times 16}$ by replacing `'y'` votes with $+1$, `'n'` votes with -1 and `'?'` with 0. We construct the unnormalized graph Laplacian matrix on X using the weight function with a range of σ and get the Fiedler vector as a classifier to compute classification accuracy. By finding the maximum clustering accuracy we find the optimal σ to calculate a range of semi-supervised learning accuracy by linear regression.

2. Theoretical Background

Consider our data set $X = \{\underline{x}_0, ..., \underline{x}_{N-1}\} \in \mathbb{R}^d$ and asymetric matrix $W \in \mathbb{R}^{N\times N}$ with non-negative entries $w_{ij} \geq 0$ [2]. We then define a (weighted undirected) graph $G = \{X, W\}$ where the $\underline{x}_j \in \mathbb{R}^d$ are the vertices of G and the entries w_{ij} of W denote weights that are associated to edges that connect $\underline{x}_i$ to $\underline{x}_j$. In this project we will use

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

so

$$w_{ij} = exp(-\frac{||\underline{x}_i - \underline{x}_j||_2^2}{2\sigma^2})$$

With the matrix W we define the graph Laplacian matrix of G:

- Define the degree vector

$$\underline{d} \in \mathbb{R}^N, d_j = \sum_{i=0}^{N-1} W_{ji}$$

Date: March 12, 2022.

- Define the diagonal degree matrix

$$D = diag(\underline{d})$$

- Define the (Unormalized) graph Laplacian

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

$\hat{L}$ hace real eigenvalues and eigenvectors $\{\lambda_j, \underline{u}_j\}_{j=0}^{N-1}$

$$0 = \lambda_0 \leq \lambda_1 \leq ... \leq \lambda_{N-1}$$

Number of zero eigenvalues of $\hat{L}$ correspond to "clusters" in the graph.

There are two key steps in spectral clustering [4]:

- Construct feature map (Laplacian embedding)

$$F : \mathbb{R}^d \to \mathbb{R}^M$$

- Cluster/classify the mapped data $F(x)$

$\hat{\underline{q}_0}$ and $\hat{\underline{q}_1}$ are precisely the indicators of the X_0 and X_1, i.e., the set of points belonging to the sub graph G_0 and G_1. In fact the first eigenvector is always $\underline{q}_0 = 1$ and so the second vector also known as the Fiedler vector.

Semi-supervised learning (SSL) combine features of both supervised and unsupervised learning [3]. In SSL we assume our data is avaliable in the following form: we have inputs

$$\mathbb{R}^{d \times N} \to X = \{\underline{x}_0, \underline{x}_1, ..., \underline{x}_{N-1}\}$$

along with some outputs,

$$\mathbb{R}^M \to Y = \{\underline{y}_0(\underline{x}_0), \underline{y}_1(\underline{x}_1), ..., \underline{y}_{M-1}(\underline{x}_{M-1})\}$$

We call the pairs $\{(\underline{x}_0, \underline{y}_0), ..., (\underline{x}_{M-1}, \underline{y}_{M-1})\}$ the labelled data subset and the set $\{\underline{x}_M, ..., \underline{x}_{N-1}\}$ the unlabelled data subset.

Semi-supervised regression (SSR) is a particular case of SSL where the outputs $y(\underline{x})$ are real valuess then $y = (\underline{y}_0(\underline{x}_0), \underline{y}_1(\underline{x}_1), ..., \underline{y}_{M-1}(\underline{x}_{M-1})) \in \mathbb{R}^M$.

Pseudo-algorithm

- Given $X \in \mathbb{R}^{d \times N}$ and $y \in \mathbb{R}^M$
- construct a similarity graph $G = \{X, W\}$ along with a graph Laplacian $L \in \mathbb{R}^{N \times N}$.
- compute the first $k¿0$ eigenvectors $\{\underline{q}_k\}_{k=0}^{k-1}$
- solve the Ridfe regression problem

$$\hat{\underline{c}} = \underset{\underline{c} \in \mathbb{R}^k}{\arg\min} \sum_{j=0}^{M-1} |\sum_{k=0}^{k-1} c_k q_{jk} - y_j|^2 + \lambda ||\underline{c}||_2^2$$

In this project, rather than labeling each point $\underline{x}_j \in X$ with a label $y_j \in \mathbb{R}$ we instead take $\underline{y}_j \in \mathbb{R}^K$ where K is the number of classes in data set [5].

3. Algorithm Implementation and Development

We will use

- `numpy` [1] for mathematical operations
- `matplotlib.pyplot` [6] to plot graph
- `scipy.spatial` [8] to calculate distance matrix
- `sklearn.linear_model` to use `Ridge` [7] to apply linear regression

4. Computational Results

We first reconstruct the input and output data. We replace all `'republican'` as +1 and `'democrat'` as -1 to reconstruct a new output vector y and replacing `'y'` votes with +1, `'n'` votes with -1 and `'?'` with 0 in the input data. We get a vector $y \in \mathbb{R}^{435}$ and an input matrix $X \in \mathbb{R}^{435\times 16}$.

By constructing the unnormalized graph Laplacian matrix on X using the given weight function with σ in range of (0,4], we can get the second eigenvector (i.e., the Fiedler vector) which we denote as q_1. Take $sign(q_1)$ as classifier and compare with y to compute its classification accuracy $1 - \frac{1}{435}\times$ number of misclassified members. Here since it is unsupervised learning, the classifier might take the opposite sign with y, we calculate the clustering accuracy in the following way: compare the classifier with both y and $-y$, take the smallest number of misclassified members, and then calculate the accuracy. We plot accuracy vs. σ:

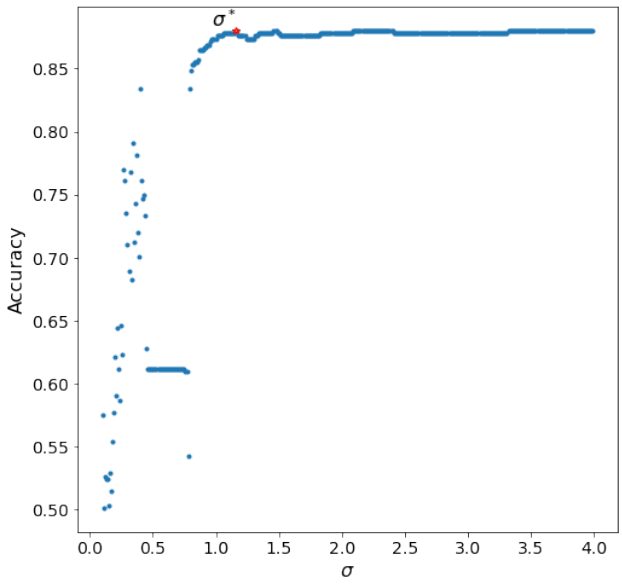

Figure 1. Clustering Accuracy with $\sigma \in (0,4]$

By finding the maximum clustering accuracy, we get the optimal value of $\sigma^* = 1.1599999999999995$, which is denoted as a red star in the figure.

With the optimal σ^*, we now use the Laplacian matrix to get the Laplacian embedding

$$F(x_j) = ((q_0)_j, (q_1)_j, ..., (q_{M-1})_j) \in \mathbb{R}^M$$

where $M \geq 1$. We now have $F(X) \in \mathbb{R}^{435\times M}$. Then get submatrix $A \in \mathbb{R}^{J\times M}$ and vector $b \in \mathbb{R}^J$ which are the first J rows of $F(X)$ and y, where $J \geq 1$. Use linaer regression to get

$$\hat{\beta} = \arg\min_{\beta\in\mathbb{R}^M} ||A\beta - b||_2^2$$

and get predictor of all points in X $\hat{y} = \text{sign}(F(X)\hat{\beta})$. We have $M = 2, 3, 4, 5, 6$ and $J = 5, 10, 20, 40$. For each combination of M and J, we compare the calculated predictor $\hat{y}$ and label y to get the SSL accuracy. Here is the table of SSL accuracy:

M \ J	5	10	20	40
2	0.88965517	0.88735632	0.88275862	0.88045977
3	0.88965517	0.82068966	0.82298851	0.83908046
4	0.83908046	0.85747126	0.86436782	0.87586207
5	0.86666667	0.75402299	0.84137931	0.87816092
6	0.89195402	0.72873563	0.87586207	0.86436782

The maximum accuracy is when $M = 6$ and $J = 5$, and the minimum accuracy is when $M = 6$ and $J = 10$, however, the accuracy does not change so much while adjusting the value of M and J. In general, the choice of M and J will not affect too much on the output, and $J = 5$ will be sufficient.

5. Summary and Conclusions

By calculating the accuracy of spectral clustering and semi-supervised learning we can test the performance of these two algorithms. With the data set of the 1984 house voting records, we can see that these algorithms do well on the data set with relatively high accuracy.

Acknowledgements

The author is thankful to every thought in the AMATH 582/482 Discord server, and Professor Hosseini for the code in lecture 20 and 21 of spectural clustering and semi-supervised learning.

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 graph laplacians. University of Washington (LOW 216), Feb 2022.

[3] B. Hosseini. Semi-supervised learning. University of Washington-Seattle (LOW 216), Feb 2022. AMATH 482/582.

[4] B. Hosseini. Spectral clustering. University of Washington (LOW 216), Feb 2022.

[5] B. Hosseini. Ssl demo. University of Washington-Seattle (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.

[8] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.