Independent Study

Yun-shao Sung*

Abstract. This is independent study mid-way report

1. Abstract. Hi1!

2. Introduction. Hi2!

3. Baseline Approach. Here is the baseline approach which can be roughly divided into two parts: the first part is constructing symmetric matrix from the feature similarity along time, and the second part is to construct symmetric normalized Laplacian matrix and obtain the top m eigenvectors with the top m smallest eigenvalues, and then perform k-means for boundary detection.

Algorithm 1 Baseline Approach

Input: number of top m smallest eigenvalues

```
M = getSymmetricMatrix()
L = scipy.sparse.csgraph.laplacian()
eigVals, eigVecs = np.linalg.eig(L)
Y = getMthSmallest(eigVals, eigVecs, m)
return boundaryDetection(Y)
```

3.1. Symmetric and Laplacian Matrix. Hi2sub!

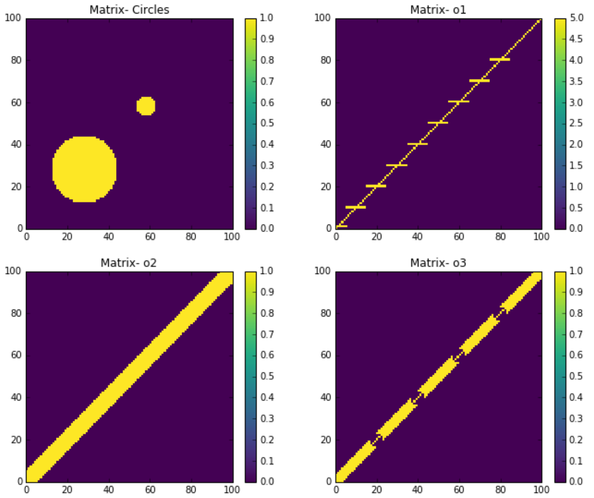

Figure 3.1. *The training and test accuracy with epochs*

*yss265@nyu.edu

3.2. Boundary Detection. As the pseudocode in Algorithm 2, once the Laplacian matrix is constructed, each row is the representation of eigen-features at specific time. Therefore, running k-means for eigen-features of all time points will yield the results of which centroids of this time point belongs to, and therefore the place where the $t_i \neq t_{i+1}$ is where the boundary is. As showed in figure 3.1, boundary is correctly detected at each time point, but when doing experiments I noticed the initiation and number of iteration during k-means will affect the correctness of boundary, and the more iteration usually will gives better boundary.

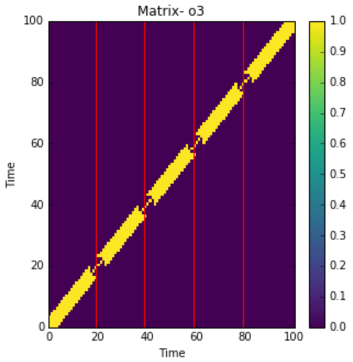

Figure 3.2. *The original symmetric matrix and detected boundary*

Algorithm 2 boundaryDetection

Input: Laplacian eigenvectors $Y \in R^{n \times m}$

Output: Boundary b, Centroids c

1: $\overline{y}_i = \frac{Y_i}{\|Y\|}$ //normalize each row Y_i
2: Run k-means on $\{\overline{y}_i\}_{i=1}^{n}$
3: Let c_i denote the cluster containing $\overline{y}_i$
4: $b \leftarrow \{i | c_i \neq c_{i+1}\}$
5: return b, c

3.3. ToDos. Hi3!

REFERENCES

[1] A Tutorial on Spectral Clustering
[2] Analyzing Song Structure with Spectral Clustering
[3] Deep clustering: Discriminative embeddings for segmentation and separation
[4] Learning Deep Representations for Graph Clustering
[5] Hierarchical Evaluation of Segment Boundary Detection