Application of Spectral Clustering Algorithm

Danielle Middlebrooks
dmiddle1@math.umd.edu

Advisor: Kasso Okoudjou
kasso@umd.edu
Department of Mathematics

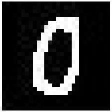
University of Maryland- College Park
Advance Scientific Computing II

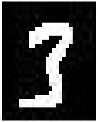
May 11, 2016

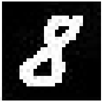
Outline

Background Information

- Spectral Clustering is technique that makes use of the spectrum of the similarity matrix derived from the data set in order to cluster the data set into different clusters.
- Implement an algorithm that groups same digits from the MNIST Handwritten digits database in the same cluster.
- In practice this algorithm and my code will work for any database that wants to group together similar objects.







Motivation

Motivated by the N cut problem.

$$\min \mathrm{NCut}(A_1, ..., A_k) := \min \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{vol(A_i)}$$

where

- A is a subset of the vertices V
- the compliment $\bar{A} = V \setminus A$
- $W(A_i, A_j) = \sum_{i \in A_i, j \in A_j} w_{ij}$
- $vol(A) = \sum_{i \in A} d_i$

The idea is that the eigenvectors serve as indicator functions in order to easily cluster the database in a reduced dimension.

Implementation

- Personal Laptop: Macbook Pro.
    - Matlab R2016b
    - 4GB Memory
- Desktop provided by Norbert Wiener Center
    - Matlab R2015b
    - 128GB Memory

Normalized Laplacian Matrix

- Guassian Similarity Function: $s(X_i, X_j) = e^{\frac{-||x_i - x_j||^2}{2\sigma^2}}$ where σ is a parameter.
- W- Adjacency matrix $w_{ij} = \begin{cases} 1, & \text{if } \; s(X_i, X_j) > \epsilon \\ 0, & \text{otherwise} \end{cases}$
- D- Degree matrix
- Unnormalized Laplacian Matrix: $L = D - W$
- Normalized Laplacian Matrix: $L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$

Normalized Laplacian Matrix

- As validation we know the smallest eigenvalue of the Normalized Laplacian will be zero with eigenvector $D^{1/2}\mathbb{1}$
- To choose the best parameters, we implement the entire algorithm a number of times, changing epsilon each time until we reach some tolerance for the total error

$$\sigma = 2000$$

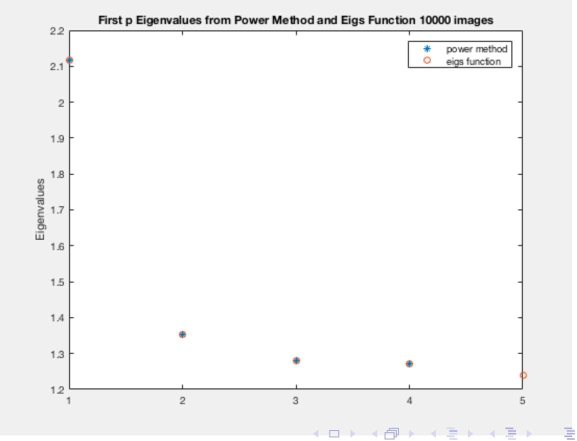
$$\epsilon = 0.3575$$

Modified B Matrix

- Normalized Laplacian Matrix:
  $L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2} = I - B$
- Computing the first p eigenvalues of B using the power method give us the largest eigenvalues in magnitude.
- Let $B_{mod} = B + \mu I$ where $\mu =$ max(sum(B,2))

Computing first p Eigenvectors

Using the Power Method with Deflation on B_{mod} we compute the first p eigenvalues.



Computing first p Eigenvectors

By changing convergence criterion and increasing max iterations we obtain

	λ_1	λ_2	λ_3	λ_4
r	6.90E-15	1.18E-14	2.44E-10	2.84E-09

$r = \text{norm}(\frac{B}{\lambda} v - \frac{B}{\lambda^*} v^*, 2)$
(λ, v) came from power method
(λ^*, v^*) came from eigs function

Row Normalization

Let $T \in \mathbb{R}^{nxk}$ be the eigenvector matrix with norm 1.
Set

$$t_{i,j} = \frac{v_{i,j}}{(\sum_p v_{i,p}^2)^{1/2}}$$

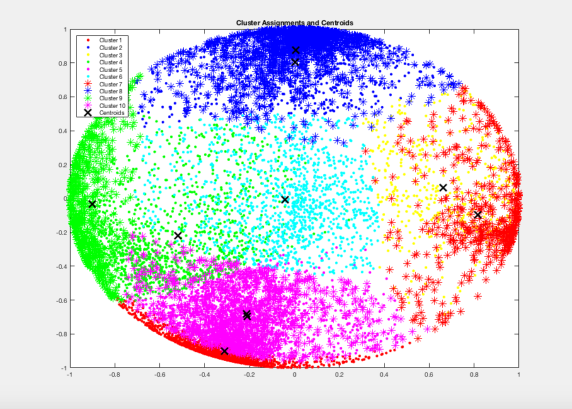
$$\begin{bmatrix} v_{11} & v_{12} & v_{13} & \dots & v_{1p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_{i1} & v_{i2} & v_{i3} & \dots & v_{ip} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & v_{n3} & \dots & v_{np} \end{bmatrix} \Rightarrow \begin{bmatrix} t_{11} & t_{12} & t_{13} & \dots & t_{1p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{i1} & t_{i2} & t_{i3} & \dots & t_{ip} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{n1} & t_{n2} & t_{n3} & \dots & t_{np} \end{bmatrix}$$

K-means Clustering

Let y_i be the ith row of T

- Randomly select k cluster centroids, z_j.
- Calculate the distance between each y_i and z_j.
- Assign the data point to the closest centroid.
- Recalculate centroids and distances from data points to new centroids.
- If no data point was reassigned then stop, else reassign data points and repeat.

K-means Clustering



Assign the original point X_i to cluster j if and only if row i of the matrix T was assigned to cluster j.

Cluster Classification

Next we classify each cluster as a particular digit.

Digit	0	1	2	3	4	5	6	7	8	9
Cluster Class	6	5	2	3	7	9	8	4	1	10

Run time: 23mins

Results

Below is a table of error for each cluster on 2000

$$\text{Error}=\frac{\text{Number of incorrect digits in cluster}}{\text{Total number of digits in cluster}}$$

1	2	3	4	5	6	7	8	9	10
78%	82%	48%	65%	39%	13%	69%	58%	65%	72%

$$\text{Overall Error}=\frac{\text{Total number of incorrect digits}}{\text{Total number of digits}}=59\%$$

Overall Error on 1000 images=64%
Overall Error on 10000 images=49%

Results

Cluster 6

Cluster 4

Cluster 3

Addition of New Datapoint- Standard Method

Proposition (Nystrom Method)

Method for out-of-sample extension
Goal: Use a similarity kernel function $K(x, y)$ in order to embed the new data point x in the reduced dimension.
Benjio, Y, et al. Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering

Addition of New Datapoint- Another Method?

We can determine which cluster a single new datapoint belongs to without re running the entire code.

- Create a similarity vector, denoted as X_{sim} of 0's and 1's
- Normalize the similarity vector by multiplying it by $D^{1/2}$
- Compute the projection of the similarity vector onto the eigenvectors of the Normalized Laplacian matrix and normalize. Denoted as C_{sim} that lives in $\mathbb{R}^p$.
- Find the centroid that is closest to C_{sim}

Results

Implementation on a random subset of 100 digits.

	Error	Runtime
Averaged over 100 digits	61%	12.6sec

Yale Face Database

- Contains 165 grayscale images of 15 individuals.
- 11 images per subject, one per different facial expression or configuration.
- Each image is 32x32 pixels

Results

Using 10 subjects and 5 images per subject with $\sigma = 2000$ and $\epsilon = 0.465$

Image	1	2	3	4	5	6	7	8	9	10
Cluster Class	5	6	8	4	2	7	9	10	3	1

Below is a table of error for each cluster classification

$$\text{Error} = \frac{\text{Number of incorrect faces in cluster}}{\text{Total number of faces in cluster}}$$

1	2	3	4	5	6	7	8	9	10
71%	33%	60%	83%	0%	66%	44%	40%	60%	66%

$$\text{Overall Error} = \frac{\text{Total number of incorrect faces}}{\text{Total number of faces}} = 54\%$$

Results

Cluster 5

Cluster 4

Cluster 2

Project Schedule

- End of October/ Early November: Construct Similarity Graph and Normalized Laplacian matrix. ✓
- End of November/ Early December: Compute first k eigenvectors validate this. ✓
- February: Normalize the rows of matrix of eigenvectors and perform dimension reduction.✓
- March/April: Cluster the points using k-means and validate this step. ✓
- End of Spring semester: Implement entire algorithm, optimize and obtain final results. ✓

Conclusion

- Spectral Clustering is a relatively good clustering technique.
- Better performance when dataset is sufficiently large.
- May obtain better results by using a different Normalized Laplacian or different similarity graph.

References

[1.] Von Cybernetics, U. A Tutorial on Spectral Clustering. Statistics and Computing, 7 (2007) 4.

[2.] Shi, J. and Malik J. Normalized cuts and image segmentation. IEEE Transations on Pattern Analysis and Machine Intelligence, 22 (2000) 8.

[3.] Chung, Fan. Spectral Graph Theory. N.p.: American Mathematical Society. Regional Conference Series in Mathematics. 1997. Ser. 92.

[4.] Vishnoi, Nisheeth K.$Lx = b$ Laplacian Solvers and their Algorithmic Applications. N.p.: Foundations and Trends in Theoretical Computer Science, 2012.

[5.] Benjio, Y, Paiement, J, Vincent, P. Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering. 2003

Thank you

Proposition

Let $K(x_i, x_j)$ denote a kernel function of L_{sym} such that $L_{sym}(i, j) = K(x_i, x_j)$. Let (v_k, λ_k) be an (eigenvector,eigenvalue) pair that solves $L_{sym} v_k = \lambda_k v_k$. Let (f_k, λ'_k) be an (eigenfunction,eigenvalue) pair that solves $K f_k = \lambda'_k f_k$. Then $y_k(x)$ is the embedding associated with a new datapoint x.

$$\lambda'_k = \frac{1}{n}\lambda_k$$

$$f_k(x) = \frac{\sqrt{n}}{\lambda_k}\sum_{i=1}^{n} v_{ik} K(x, x_j)$$

$$y_k(x) = \frac{f_k(x)}{\sqrt{n}} = \frac{1}{\lambda_k}\sum_{i=1}^{n} v_{ik} K(x, x_j)$$