Graph Clustering Methods Derived from Column Subset Selection (Student Abstract)

Wei Mao[1], **Guihong Wan**[2], **Haim Schweitzer**[1]

[1]Department of Computer Science, University of Texas at Dallas, Texas, USA
[2]Department of Dermatology, Massachusetts General Hospital, Harvard Medical School, MA, USA
wei.mao@utdallas.edu, guihong_wan@hsph.harvard.edu, haim@utdallas.edu

Abstract

Spectral clustering is a powerful clustering technique. It leverages the spectral properties of graphs to partition data points into meaningful clusters. The most common criterion for evaluating multi-way spectral clustering is NCut. Column Subset Selection (CSS) is an important optimization technique in the domain of feature selection and dimension reduction, which aims to identify a subset of columns of a given data matrix that can be used to approximate the entire matrix. In this study, we show that CSS can be used to compute spectral clustering and use this to obtain new graph clustering algorithms.

Definition of the Graph Clustering Problem

General graph definitions: Let G be an undirected graph of n nodes and m edges. Let w_{ij} be the non-negative weight between nodes i and j, so that the matrix $W = (w_{ij})$ is $n \times n$. The degree of node i is $d_i = \sum_j w_{ij}$. The degree vector is $d = (d_1, \ldots, d_n)^T$, and the $n \times n$ degree matrix is $D = \mathsf{diag}(d)$. The graph Laplacian is $L = D - W$.

Graph clustering: Consider the vertices of the graph G partitioned into the k disjoint subsets: $A_1, \ldots, A_k$. For $t{=}1, \ldots, k$ define: $V_t = \sum_{i \in A_t} d_i$, $C_t = \sum_{i \in A_t, j \notin A_t} w_{ij}$. Here V_t is the cluster volume and C_t is the cluster cut. The following clustering criterion is the most common in the spectral clustering literature (Von Luxburg 2007):

$$\mathsf{NCut}(A_1, \ldots, A_k) = \frac{1}{2} \sum_{t=1}^{k} \frac{C_t}{V_t}. \tag{1}$$

Edge Matrix

The edge matrix represents the graph as an $n \times m$ matrix. Each column of the matrix contains the information of a single edge of the graph. This matrix has some important properties related to our technique of obtaining graph clustering methods with CSS. Different kinds of normalization can be applied to the edge matrix. Here we focus on the specific normalized matrix which can be used to minimize the NCut with CSS.

Definitions and Notation

Edge vectors and the edge matrix:
The **edge vector** e associated with the edge between the nodes i, j is defined as the following n-vector:

$$e = \sqrt{w_{ij}}\,(0, \ldots, 0, -1, 0, \ldots, 1, 0, \ldots, 0)^T, \quad -1 \text{ is in the } i\text{th location, } 1 \text{ is in the } j\text{th location.} \tag{2}$$

The **edge matrix** E associated with the graph G is the following $n \times m$ matrix:

$$E = (e_1, e_2, \ldots, e_m)\,, \tag{3}$$

where e_i is the edge vector of the ith edge. (The edge order is arbitrary.)

Normalized edge vectors and the edge matrix:
Recall that $D{=}\mathsf{diag}(d)$. We write $D^{-\frac{1}{2}}$ for $(D^{+})^{\frac{1}{2}}$, where the D^{+} is the pseudo-inverse of D.
The **normalized edge vector** $\tilde{e}$ associated with the edge between the nodes i, j is defined as the following n-vector:

$$\tilde{e} = D^{-\frac{1}{2}} e, \tag{4}$$

where e is the edge vector of the edge between i, j.
The **normalized edge matrix** $\tilde{E}$ associated with the graph G is the following $n \times m$ matrix:

$$\tilde{E} = (\tilde{e}_1, \tilde{e}_2, \ldots, \tilde{e}_m) = D^{-\frac{1}{2}} E. \tag{5}$$

Properties of an Edge Matrix

The edge matrix has the following properties which can help us build the connection between spectral clustering and CSS:

Theorem 1. *Let G be a graph with the corresponding normalized edge matrix $\tilde{E}$. Let S be a subset of G edges, and let $\tilde{E}_S$ be its normalized edge matrix. If the edges in S can form a circle in G then:* ***The columns of $\tilde{E}_S$ are not linearly dependent.***

A simple corollary of **Theorem 1** is:

Theorem 2. *Let G be a graph with the corresponding normalized edge matrix $\tilde{E}$. Let S be a subset of G edges, and let $\tilde{E}_S$ be its normalized edge matrix. If $\tilde{E}_S$ is full rank (the columns of $\tilde{E}_S$ are linearly independent) then:* ***The edges in S can not form a circle in G.***

The Column Subset Selection Problem

Let $X=(x_1,\ldots,x_m)$ be an $n \times m$ matrix. Let X_S be an $n \times k$ matrix constructed from k columns of X. The linear approximation of X by X_S can be written as: $X \approx X_S B$, where B is a coefficient matrix of size $k \times m$. The squared Frobenius norm of the approximation error is given by:

$$\mathsf{ApproxErr}(X, X_S) = \frac{1}{2}\|X - X_S X_S^+ X\|_F^2. \qquad (6)$$

Here X_S^+ is the pseudo inverse of X_S. The CSS problem is the problem of finding S that minimizes the error criterion (6). See e.g., (He et al. 2019).

Equivalence

The following theorem shows an equivalence between spectral clustering and CSS:

Theorem 3. *Let G be a graph with the corresponding normalized edge matrix $\tilde{E}$. Let S be a subset of G edges, and let $\tilde{E}_S$ be its normalized edge matrix. Define r as the rank of $\tilde{E}_S$. Then:*

(a). *Let $k = n-r$. The edges in S form a subgraph G_S with k connected components $A = \{A_1,\ldots,A_k\}$.*

(b). $\mathsf{NCut}(A) = \mathsf{ApproxErr}(\tilde{E}, \tilde{E}_S)$.

See the full paper for the proof.

The Black Box Algorithm

As a result of **Theorem 3**, spectral clustering can be solved by CSS. We have the following "black box" algorithm:

Algorithm 1: The black box graph clustering algorithm

Input: G, a graph with n nodes to be clustered. k, the desired number of clusters. A black box CSS algorithm.

Output: The k clustering of the graph.

1. Compute $\tilde{E}$ from G according to (5).
2. Run the CSS algorithm to select linearly independent S containing $n-k$ columns from $\tilde{E}$.
3. Construct the subgraph G_S from the selection.
4. Return the connected components of G_S as output.

Examples

We tested the following three algorithms:

NJW. This is the classical spectral clustering algorithm from (Ng, Jordan, and Weiss 2001).

QRP. QR Factorization with Column Pivoting (QRP) is a classical CSS algorithm known for nearly 60 years (Businger and Golub 1965).

It is greedy, and starts by selecting the column with the largest norm. In each iteration it subtracts projections on the selected column from each remaining column, and again selects the one with the largest norm among the reduced columns.

GREEDY. The GREEDY algorithm is greedy, and starts by selecting the column that best predicts the entire matrix. In each iteration it subtracts projections on the selected column from all remaining columns, and then selects the column that gives the best prediction for the reduced matrix (Farahat et al. 2015).

The following panels show results obtained on graphs computed from an image taken from the publicly available "COIL" data set. The graph was computed by creating edges between all adjacent pixels. In terms of NCut, GREEDY performs the best, NJW second, and QRP third. However, looking at the images the QRP produced "very reasonable" clusters.

Figure 1: experiments with various algorithms.

For additional results see (Wan et al. 2024).

References

Businger, P.; and Golub, G. H. 1965. Linear Least Squares Solutions by Householder Transformations. *Numer. Math*, 7: 269–276.

Farahat, A.; Elgohary, A.; Ghodsi, A.; and Kamel, M. S. 2015. Greedy column subset selection for large-scale data sets. *Knowledge and Information Systems*, 45: 1–34.

He, B.; Shah, S.; Maung, C.; Arnold, G.; Wan, G.; and Schweitzer, H. 2019. Heuristic search algorithm for dimensionality reduction optimally combining feature selection and feature extraction. In *AAAI*, volume 33, 2280–2287.

Ng, A.; Jordan, M.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. *NeurIPS*, 14.

Von Luxburg, U. 2007. A tutorial on spectral clustering. *Statistics and computing*, 17: 395–416.

Wan, G.; Mao, W.; Semenov, Y. R.; and Schweitzer, H. 2024. Equivalence between Graph Spectral Clustering and Column Subset Selection (Student Abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*.