DIRECTLY SOLVING THE ORIGINAL RATIOCUT PROBLEM FOR EFFECTIVE DATA CLUSTERING

Jing Li, Feiping Nie, Xuelong Li

School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi’an 710072, P. R. China

ABSTRACT
This paper focuses on the original RatioCut problem, which is one of the most representative clustering paradigms. The RatioCut criterion looks for a partition of the graph to achieve the mincut cost while keeping each partition reasonably large. This well-known problem is NP hard and its relaxed form has been widely used in the past several decades. However, the relaxed RatioCut usually suffers two problems: not satisfactory stable clustering performance, and undesired two-stage optimization. In this work, we solve the original RatioCut problem by learning a new similarity matrix which has as many connected components as the cluster number, so that the original RatioCut constraint can be directly satisfied. An easily implemented algorithm is derived to iteratively optimize the proposed method. Experimental results on various real-world benchmark datasets exhibit the effectiveness of the proposed method to solve the RatioCut problem.

Index Terms— RatioCut clustering, discrete constraint conditions, iterative optimization

1. INTRODUCTION
Graph-based clustering methods have been successful in data analysis. In the context of graph cut, one needs to find a partition of the graph such that the edges between different groups are exactly cut off. Since the edges between different groups have the low weights while they have the high weights within the same group, the mincut cost is often used as the objective. A very early work about the bipartition to a graph has been discussed in [1]. To prevent the solution of mincut simply separating one individual vertex from the rest of the graph, RatioCut [2] and Ncut [3] explicitly request that every cluster is reasonably large. However, such a balancing condition makes them become NP hard [4], thus many subsequent popular works trickly use eigenvalue decomposition to solve a relaxed problem.

Although many recent applications have successfully employed graph cut technique in their domains, such as graph clustering [5, 6], image segmentation [7, 8], and saliency detection [9], few of them concern about a feasible solution to the original graph cut problem. Specifically, most of these methods first adopt an optimization strategy to the given graph to get a relaxed continuous solution and then discretize it with K-means or spectral rotation algorithms (see [10] and discussion therein). One disadvantage of this kind of approaches is that the final clustering structures are not identical with the data graph (Note that postprocessing like K-means algorithm itself is not a stable operation). The similar consideration is also taken into by [11], where they tackle this problem by the graph approximation. Actually, for the graph-cut problem, a welcome expectation is that we can directly learn the discrete solution which strictly satisfies the discrete constraint conditions.

In this work we propose a novel graph-based clustering method, which can be used to effectively solve the original RatioCut problem. To be specific, we introduce a new similarity matrix, which is supposed to have \( c \) (\( c \) is the cluster number) connected components, and helps suppress the constraints to be satisfied. Hence, there is no need to switch between discrete and continuity for the target variable in our method. Different from the previous methods (e.g., unnormalized spectral clustering first calculates the eigenvectors of Laplacian matrix and then clusters the derived representation of each data with K-means algorithm), we solve the proposed objective via an easily implemented iterative optimization algorithm. To validate the effectiveness of the proposed method, empirical studies are conducted on different real-world benchmark data sets. The experimental results demonstrate the proposed method outperforms other compared methods in most cases, and more importantly it significantly improves the performance of the relaxed RatioCut.

Notation: Throughout the paper, every matrix is written as uppercase. For a matrix \( M \), the \( i \)-th row, the \( j \)-th column, and the \( ij \)-th element of \( M \) are denoted by \( m_i \), \( m_j \), and \( m_{ij} \), respectively. The trace of matrix \( M \) is denoted by \( Tr(M) \). The L2-norm of vector \( v \) is denoted by \( ||v||_2 \), and the Frobenius norm of matrix \( M \) is denoted by \( ||M||_F \).

2. PROBLEM FORMULATION
Given an undirected data graph described by a similarity matrix \( W \in \mathbb{R}^{n \times n} \), where \( n \) is the number of data points, the main task of clustering is to partition these points into \( c \) group-
s. We use $Y \in \mathbb{R}^{n \times c}$ to denote the indicator matrix, where $y_{ij} = 1$ if the $i$-th data is assigned to the $j$-th cluster, it is 0 otherwise. Let $Y = [y_1, y_2, \ldots, y_f]$, and then $c$-way RatioCut criterion is described as an optimization program of variable $Y$ [2]:

$$
\max_Y \frac{1}{c} \sum_{j=1}^{c} \frac{y_{1j}^T W y_{1j}}{y_{1j}^T y_{lj}} \quad \text{s.t.} \quad Y \in \{0, 1\}^{n \times c}, Y1_c = 1_n, \quad (1)
$$

where $1_d$ denotes a $d$ dimensional column vector of all 1’s. By defining a scaled cluster assignment matrix $F \in \mathbb{R}^{n \times c}$ as

$$
F = Y (Y^T Y)^{-\frac{1}{2}}, \quad (2)
$$

the problem (1) is easily verified to be equivalent to the following form (where the const $\frac{1}{c}$ has been omitted)

$$
\min_{F \in \text{Disc}} \text{Tr} (F^T L F), \quad (3)
$$

which is compact and has been widely used in previous clustering works [12, 13]. In problem (3), $L$ is the so-called Laplacian matrix and $L = D - W$, where the degree matrix $D \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose $i$-th diagonal element is $\sum_j w_{ij}$, $\text{Disc}$ is short for discrete and this constraint represents that $F$ should satisfy the Eq. (2), where $Y$ is discrete as constrained in the problem (1). Obviously, the problem (3) is NP hard and the optimal discrete solution cannot be directly obtained. Thus, as stated above, many previous works resort to solving the relaxed problem (i.e., the constraint is reduced to $F^T F = I_c$) and add a discretizing postprocessing, which makes it a two-stage procedure and not robust in practice.

Intuitively, if the data can be partitioned into $c$ clusters, an ideal neighbors assignment is supposed to contain exactly $c$ connected components. In this paper, we leverage this property by introducing a new similarity matrix $S \in \mathbb{R}^{n \times n}$ with $c$ connected components to help us tackle the problem (3). Since this property doesn’t go straightforward with the problem (3), we do the following transformation.

The Laplacian matrix of $S$ is defined as $L_S = D_S - \frac{s^T + S}{2}$, where the degree matrix $D_S \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose $i$-th diagonal element is $\sum_j (s_{ij} + s_{ji})/2$. $L_S$ is obviously symmetric and positive semi-definite which has the following property [14]:

**Theorem 1.** The multiplicity $c$ of the eigenvalue 0 of the Laplacian matrix $L_S$ is equal to the number of connected components in the graph with the similarity matrix $S$.

Furthermore, let $\sigma_i (L_s)$ denote the $i$-th smallest eigenvalue of $L_s$. According to the Ky Fans Theorem [15], the following equation holds

$$
\sum_{i=1}^{c} \sigma_i (L_s) = \min_{F^T F = I_c} \text{Tr}(F^T L_S F). \quad (4)
$$

Therefore, we can add the right of Eq. (4) to the problem (3), where each $F$ shares the orthogonal constraint. Since we include $S$ as a new variable, to assert that $S$ is too sparse to be capable of forming solid components, a regularized term $||S||_F^2$ is introduced. Thus we come to the new objective:

$$
\min_{F,S} \text{Tr} (F^T LF) + \alpha ||S||_F^2 + \lambda \text{Tr}(F^T L_S F) \quad \text{s.t.} \quad F^T F = I_c, s_{ij} \geq 0, \sum_j s_{ij} = 1, \quad (5)
$$

where $\alpha$ is a positive hyperparameter, each element in $S$ is constrained as nonnegative, and each row of $S$ sums to 1. Particularly, when $\lambda$ is large enough, according to Eq. (4), we know that the optimal solution $S$ to the problem (5) will make the term $\sum_{i=1}^{c} \sigma_i (L_s)$ equal to zero and thus the similarity matrix $S$ will have $c$ connected components. Under this condition, from Theorem 2 [13] the solution $F$ to the problem (5) is consistent with the $\text{Disc}$ constraint.

**Theorem 2.** If the Laplacian matrix $L_S$ has as many eigenvalues 0 as there are connected components, then the corresponding eigenvectors are the indicator vectors of the connected components.

Let us take an example. Suppose we have the Laplacian matrix $L_S$ with 3 connected components like Fig. 1 (left) (For better presentation, we rearrange $S$ to be diagonal). Instead of directly operating on $L_S$, we calculate the eigenvectors of its every diagonal block and then fill in the remaining empty positions with zeros (see Fig. 1 (right)). By arranging the eigenvectors of each block based on the corresponding eigenvalues from large to small, the right-most eigenvector of each diagonal block is $\frac{1}{\sqrt{n_i}} 1_{n_i}$ ($n_i$ is the size of a cluster), which corresponds to the eigenvalue 0. Therefore, the scaled cluster assignment matrix $F = [f^1, f^2, f^3]$, which can be verified to exactly satisfy the $\text{Disc}$ constraint.

3. OPTIMIZATION

Optimizing the problem (5) is still challenging because it involves two variables $F, S$, and simultaneously they are coupled with each other. In this section, we solve this problem by alternatingly optimizing $F$ and $S$. 

```
When $S$ is fixed, the problem (5) becomes
\[
\min_{F^T F = I} \text{Tr}(F^T LF) + \lambda \text{Tr}(F^T LSF) = \min_{F^T F = I} \text{Tr} \left( F^T (L + \lambda LS) F \right).
\] (6)
The optimal solution of $F$ is formed by the $c$ eigenvectors of $L + \lambda LS$ corresponding to the $c$ smallest eigenvalues.

When $F$ is fixed, the problem (5) becomes
\[
\min_{s_{ij} \geq 0, \sum_j s_{ij} = 1} \alpha \|S\|_F^2 + \lambda \text{Tr}(F^T LSF).
\] (7)

Let $F = [f_1; f_2; \ldots; f_n]$, then the following equation holds:
\[
2 \text{Tr}(F^T LSF) = \sum_{i,j} \|f_i - f_j\|_2^2 s_{ij}.
\] (8)
Taking Eq. (8) into the problem (7), we have the following equivalent form
\[
\min_{s_{ij} \geq 0, \sum_j s_{ij} = 1} \alpha \sum_{i,j} s_{ij}^2 + \lambda \sum_{i,j} \|f_i - f_j\|_2^2 s_{ij}.
\] (9)
Considering that the problem (9) is independent for different $i$, we solve the following problem separately for each $i$:
\[
\min_{s_{ij} \geq 0, \sum_j s_{ij} = 1} \alpha \sum_j s_{ij}^2 + \lambda \sum_j \|f_i - f_j\|_2^2 s_{ij}.
\] (10)
Denoting $v_{ij} = \|f_i - f_j\|_2^2$, and denoting $v_i$ as a vector whose $j$-th element equals to $v_{ij}$ (and similarly for $s_i$), the problem (10) can be written in vector form as
\[
\min_{s_i \geq 0, s_i, s_i = 1} \left\| s_i + \frac{\lambda}{\alpha} v_i \right\|_2^2.
\] (11)
This problem can be solved by [16] or an efficient iterative algorithm proposed in [17].

Based on the above analysis, the detailed procedure for solving the problem (5) is summarized in Algorithm 1.

**Algorithm 1** The algorithm to solve the problem (5)

**Input:** $W \in \mathbb{R}^{n \times n}$, cluster number $c$, a large enough $\lambda$

**Initialize** $F \in \mathbb{R}^{n \times c}$, which is formed by the $c$ eigenvectors of $L_S = D_S - \frac{W^T + W}{2}$ corresponding to the $c$ smallest eigenvalues

**repeat**
1. For each $i$, update the $i$-th row of $S$ by solving the problem (11)
2. Update $F$, which is formed by the $c$ eigenvectors of $L + \lambda L_S$ corresponding to the $c$ smallest eigenvalues

**until** converge

**Output:** The cluster assignment matrix $F \in \mathbb{R}^{n \times c}$, and the new similarity matrix $S \in \mathbb{R}^{n \times n}$

4. ANALYSIS OF ALGORITHM

In this section, we present more insights about the proposed algorithm.

1) **Computing the final clustering result.** When Algorithm 1 converges, we obtain a new similarity matrix $S$ with $c$ connected components, and a neat scaled cluster assignment indicator $F$. Both of them can be naturally used to partition the original data points into $c$ groups and they obtain the identical results when $\lambda$ is large enough.

2) **Convergence analysis.** The problem (5) can be divided into two subproblems and we can obtain the optimal solution to each of them. Therefore, by solving the subproblems alternatively, the proposed algorithm will converge to a local solution.

3) **The parameters.** At the first glance, there are two hyperparameters $\alpha, \lambda$ in the objective of problem (5) which need to be tuned, but actually they can be easily handled. Since $\alpha$ only appears in the subproblem (11), it can be absorbed in $\lambda$ (Throughout our experiments, we simply fix $\alpha$ to be 1). Particularly, we determine the best $\lambda$ in a heuristic way to accelerate the computation. Specifically, if there are less than $c$ components in $L_S$, we multiply $\lambda$ by two; if more, we divide $\lambda$ by two; otherwise we stop the iteration. This strategy can effectively relieve the cost for finding the optimal $\lambda$.

4) **Time complexity.** The major computation cost in each iteration involves eigenvalue decomposition (solving the subproblem (6)) which is $O(n^3)$, while optimizing the subproblem (11) is much light, only $O(n)$. Suppose the needed iterations of the proposed algorithm is $T$, we conclude that the total complexity is about $O(Tn^3)$. Comparing with the standard relaxed RatioCut ($O(n^3 + ndc)$, $d$ is the data dimension, and $i$ is the iterations), our method is expected to perform well at the expense of the larger computation.

5. EXPERIMENTS

In this section, to explore the performance of the proposed method for solving the original RatioCut problem, we conduct our experiments on seven benchmark datasets: Vehicle, Yeast, Abalone, Dermatology (Dermato), COIL20, USPS, and Umist, which are briefly summarized in Table 1.

<table>
<thead>
<tr>
<th>data set</th>
<th># of size</th>
<th># of dimensionality</th>
<th># of class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle</td>
<td>846</td>
<td>18</td>
<td>4</td>
</tr>
<tr>
<td>Yeast</td>
<td>1484</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>Abalone</td>
<td>4177</td>
<td>8</td>
<td>29</td>
</tr>
<tr>
<td>Dermatology</td>
<td>1440</td>
<td>1024</td>
<td>20</td>
</tr>
<tr>
<td>COIL20</td>
<td>1440</td>
<td>1024</td>
<td>20</td>
</tr>
<tr>
<td>USPS</td>
<td>2007</td>
<td>256</td>
<td>10</td>
</tr>
<tr>
<td>Umist</td>
<td>575</td>
<td>1024</td>
<td>20</td>
</tr>
</tbody>
</table>
Table 2. Experimental results on seven benchmark datasets.

<table>
<thead>
<tr>
<th></th>
<th>ACC</th>
<th>NMI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Vehicle Yeast Abalone Dermato COIL20 USPS Umist</td>
<td>Vehicle Yeast Abalone Dermato COIL20 USPS Umist</td>
</tr>
<tr>
<td>K-means</td>
<td>0.4421 0.3753 0.1391 0.7514 0.6944 0.6521 0.4452</td>
<td>0.1800 0.2425 0.1542 0.8616 0.7937 0.6299 0.6735</td>
</tr>
<tr>
<td>RRC</td>
<td>0.4456 0.4111 0.1386 0.9536 0.7813 0.6312 0.4313</td>
<td>0.2131 0.2652 0.1470 0.9051 0.8326 0.7355 0.6766</td>
</tr>
<tr>
<td>RNC</td>
<td>0.4656 0.3753 0.1515 0.9536 0.7708 0.6358 0.4661</td>
<td>0.2131 0.2401 0.1447 0.9037 0.8387 0.7330 0.6974</td>
</tr>
<tr>
<td>NMF</td>
<td>0.3995 0.3740 0.1585 0.9508 0.7833 0.6746 0.4887</td>
<td>0.1676 0.2655 0.1460 0.9010 0.8540 0.7568 0.6979</td>
</tr>
<tr>
<td>DRC</td>
<td>0.4492 0.4683 0.1764 0.9563 0.8271 0.7165 0.4870</td>
<td>0.2168 0.2994 0.1565 0.9098 0.8941 0.7718 0.7092</td>
</tr>
</tbody>
</table>

Fig. 2. Clustering results comparison between RRC and the proposed method DRC on seven benchmark datasets

As a convention in [18, 11], the proposed Directly solving RatioCut (DRC) method is compared with $K$-means, Relaxed RatioCut (RRC), Relaxed Normalized Cut (RNC), NMF [19] methods. The input graph is constructed using the technique proposed in [20], where the Gaussian kernel can be self-tuned (The default number of nearest neighbors is 10 for every data set). For all the methods involving $K$-means, including $K$-means itself, we use the random initialization strategy and repeat each of them for 100 times. Their respective results in terms of the minimum $K$-means value will be reported. As for our method, we run it only once with the initialization described in Algorithm 1, where the maximum number of iterations is set as 30, because we find that our algorithm always converge in 30 times. To measure the final clustering performance, we adopt two regular criteria: the clustering accuracy (ACC) and normalized mutual information (NMI). Table 2 shows the clustering results of each method, where the best result is marked in bold face.

From Table 2, we conclude that the proposed method DRC outperforms the competing methods almost on every data set (Particularly, it shows the great improvement on Yeast, COIL20, and USPS.). The distinct difference among DRC and other graph-based methods is that DRC learns a new similarity matrix after the optimization and thus it is insensitive to the quality of input similarity matrix $W$. Since $W$ has trifling influence in subproblem (6) when $\lambda$ is large enough, in this perspective, the input graph can be considered as an initialization of our algorithm.

Since RRC is a representative method for solving the RatioCut problem, it is an important baseline in our experiments. We compute the mean and the corresponding variance of RRC on each dataset over 100 times. The comparison between RRC and DRC is presented as in Fig. 2. It is observed that our once-run DRC impoves RRC both in ACC and NMI. Seeing that RRC obtains very unstable results (noticeable variances) in most datasets, we conclude that the proposed method provides a practical solver for the RatioCut problem.

6. CONCLUSIONS

In this paper, we directly solve the original RatioCut problem. By learning a new similarity matrix with as many connected components as its cluster number, we circumvent the strict discrete constraints in RatioCut and solve the new objective iteratively. The experimental results on seven benchmark datasets prove the effectiveness of the proposed method.
7. REFERENCES


