Directly Solving the Original RatioCut Problem for Effective Data Clustering

Jing Li, Feiping Nie, Xuelong Li

Northwestern Polytechnical University, P. R. China

{j.lee9383, feipingnie}@gmail.com, xuelong_li@opt.ac.cn

Content

- Brief review of the original RatioCut problem
- Motivation of this work
- Directly solving RatioCut (DRC)
- Optimization
- Experiments
- Summary

Brief review of RatioCut problem

Given a similarity graph $W \in \mathbb{R}^{n\times n}$, denoting by $Y = \left[y^1, y^2, \ldots, y^c \right] \in \mathbb{R}^{n\times c}$ the indicator matrix, we can write the RatioCut problem as

$$\max_{Y} \frac{1}{c} \sum_{j=1}^{c} \frac{{y^j}^T W y^j}{{y^j}^T y^j} \quad s.t. \, Y \in \{0,1\}^{n\times c}, Y 1_c = 1_n$$

By defining a scaled cluster assignment matrix $F \in \mathbb{R}^{n\times c}$ as $F = Y\left(Y^T Y\right)^{-\frac{1}{2}}$, this problem is equivalent to

$$\min_{F \in Disc} Tr\left(F^T L F\right)$$

where L is Laplacian matrix. F is discrete and should satisfy $F^T F = I_c$,

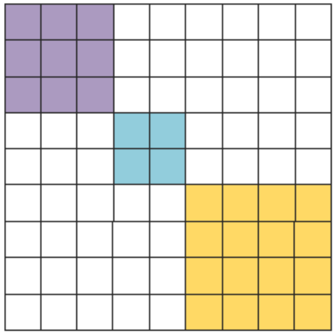
NP hard $\xrightarrow{\text{relax}}$ $\min\limits_{F^T F = I_c} Tr\left(F^T L F\right)$ $\xrightarrow{\text{optimize}}$ S1. Eigen decomposition
S2. K-means/spectral rotation

Unsteady!

Motivation of this work

- Two-stage solving procedure is not desired.
- Unsteady clustering performance makes it not practical to use.

Idea: We obtain the final clustering results as soon as the objective is solved.



Block-diagonal graph

Nie et al., [1] *KDD 2014*;
Feng et al., [2] *CVPR 2014*;
Chen and Dy, [3] *UAI 2016;*

[1] F. Nie, X. Wang, and H. Huang. Clustering and projected clustering with adaptive neighbors. *In SIGKDD*, pp. 977–986, 2014.
[2] J. Feng, Z. Lin, H. Xu, and S. Yan. Robust subspace segmentation with block-diagonal prior. *In CVPR*, pp. 3818–3825, 2014.
[3] J. Chen and J. Dy. A generative block-diagonal model for clustering. *In UAI*, pp. 25-29, 2016.

Directly solving RatioCut (DRC)

From matrix theory, we know that

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, according to the Ky Fans Theorem, we have

$$\sum_{i=1}^{c} \sigma_i \left(L_s \right) = \min_{F^T F = I_c} Tr(F^T L_S F)$$

Adding this formula to original objective, we come to

$$\boxed{\begin{aligned} &\min_{F,S} Tr\left(F^T L F \right) + \lambda Tr(F^T L_S F) + \alpha \left\| S \right\|_F^2 \\ &s.t.\ F^T F = I_c, s_{ij} \ge 0, \sum\nolimits_j s_{ij} = 1 \end{aligned}}$$

where $\left\| S \right\|_F^2$ is to avert that S is too sparse.

When λ is large enough, S will have c connected components.

Optimization

We solve the following problem by optimizing S and F alternatively:

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

When S is fixed, we need solve the subproblem

$$\boxed{\min_{F^T F = I} Tr\left(F^T\left(L + \lambda L_s\right) F\right)}$$

Standard SC

When F is fixed, we obtain the following subproblem

$$\min_{s_i \geq 0, s_i 1_n = 1} \left\| s_i + \frac{\lambda}{4\alpha} v_i \right\|_2^2$$
$$\text{where } v_{ij} = \left\| f_i - f_j \right\|_2^2$$

Alg. in [4]

[4] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the L1 ball for learning in high dimensions," *In ICML*, pp. 272–279, 2008.

Experiments

Table 1. Clustering comparison

		Vehicle	Yeast	Abalone	Dermato	COIL20	USPS	Umist
ACC	K-means	0.4421	0.3753	0.1391	0.7514	0.6944	0.6521	0.4452
	RRC	0.4456	0.4111	0.1386	0.9536	0.7813	0.6312	0.4313
	RNC	0.4456	0.3753	0.1515	0.9536	0.7708	0.6358	0.4661
	NMF	0.3995	0.3740	0.1585	0.9508	0.7833	0.6746	**0.4887**
	DRC	**0.4492**	**0.4683**	**0.1764**	**0.9563**	**0.8271**	**0.7165**	0.4870
NMI		vehicle	Yeast	Abalone	Dermato	Coil20	USPS	Umist
	K-means	0.1800	0.2425	0.1542	0.8616	0.7937	0.6299	0.6735
	RRC	0.2131	0.2652	0.1470	0.9051	0.8326	0.7355	0.6766
	RNC	0.2131	0.2401	0.1447	0.9037	0.8387	0.7330	0.6974
	NMF	0.1676	0.2655	0.1460	0.9010	0.8540	0.7568	0.6979
	DRC	**0.2168**	**0.2994**	**0.1565**	**0.9098**	**0.8841**	**0.7718**	**0.7092**

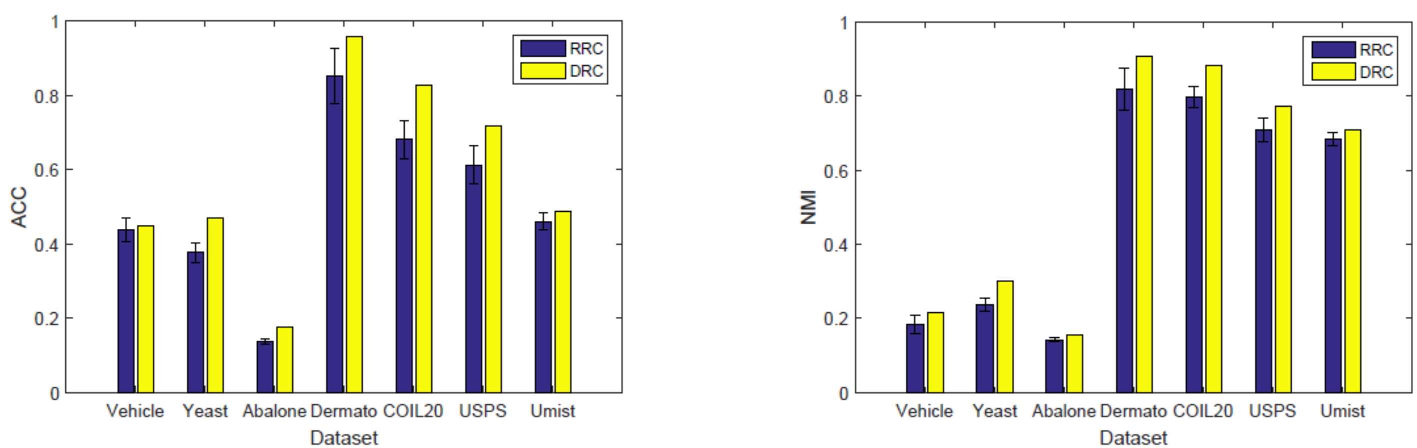


Figure 1. DRC VS. RRC

Summary

- DRC is a method which performs better at the expense of additive computation compared with Relaxed RatioCut.
- Starting with a fixed initialization, DRC obtains steady clustering results without any postprocessing.
- This work presents an example of leveraging block-diagonal similarity matrix, and this can be extended to other graph-based clustering models.

Thank you