Deep Clustering via Probabilistic Ratio-Cut Optimization

University of Colorado Boulder

Ayoub Ghriss[1] Claire Monteleoni[1,2]

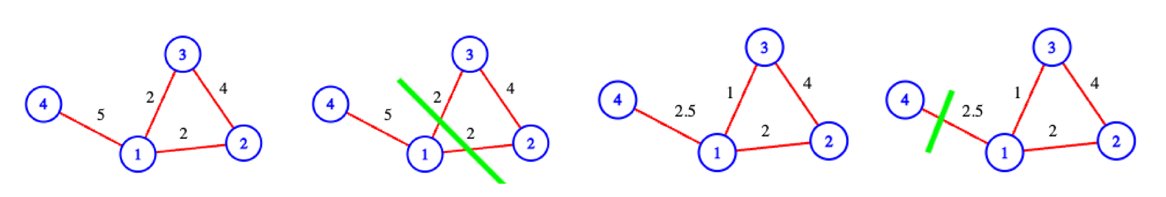
[1]Department of Computer Science, University of Colorado Boulder, [2] INRIA Paris

Abstract

We propose PRCut, optimizing graph ratio-cut by treating assignments as random variables learned online via an unbiased gradient of an expected ratio-cut bound. This novel probabilistic approach surpasses spectral relaxations, aligns clustering strongly with similarity, and effectively utilizes/evaluates self-supervised representations.

Motivation

- Ratio-cut is a fundamental graph partitioning approach.
- Advances in self-supervised learning (SSL) produce powerful embeddings capturing similarity.
- Existing Methods Limitations:
  - Contrastive: Focus on local pairs, miss global structure.
  - Generative (VAE): Prone to prior overfitting.
  - Spectral Clustering Extensions: Struggle with large scale, eigendecomposition sensitivity.
- **Our Contribution:**
  - Treats assignments as random variables.
  - Avoids spectral decomposition and Euclidean relaxation.
  - Enables online learning via stochastic gradient descent.
  - Achieves better ratio-cut and strong empirical performance.



Background: Ratio-Cut

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ a graph with n vertices and adjacency $\boldsymbol{W}$:

- Partition $\mathcal{V}$ into k disjoint clusters $\mathcal{C}_k = \{\mathbb{C}_1, \ldots, \mathbb{C}_k\}$.
- Cluster $\mathbb{C}_\ell$ represented by indicator $\mathbf{1}_{\mathbb{C}_\ell}$ and ratio $f_\ell = \frac{\mathbf{1}_{\mathbb{C}_\ell}}{|\mathbb{C}|}$.
- Ratio-Cut objective:

$$\text{RatioCut}(\mathcal{C}_k) = \frac{1}{2}\sum_{\ell=1}^{k} \sum_{i,j \in \mathbb{C}_\ell \times \overline{\mathbb{C}}_\ell} \frac{W_{ij}}{|\mathbb{C}_\ell|} \propto \sum_{\ell=1}^{k} \sum_{i,j} W_{ij} \left(f_{\ell i} - f_{\ell j}\right)^2$$

- Matrix form using unnormalized Laplacian $\boldsymbol{L}_{un} = \boldsymbol{D} - \boldsymbol{W}$:

$$\text{RatioCut}(\mathcal{C}_k) = \frac{1}{2}\text{Tr}\left[\boldsymbol{F}_{\mathcal{C}_k}^\top \boldsymbol{L}_{un} \boldsymbol{F}_{\mathcal{C}_k}\right]$$

- Optimization is NP-hard. Spectral clustering relaxes this to find eigenvectors of $\boldsymbol{L}_{un}$.

Probabilistic Ratio-Cut (PRCut)

Core Idea: Assignments are independent random variables.

$$\Pr\left(v_i \in \mathbb{C}_\ell\right) = \Pr\left(\mathbf{a}_i^{(\ell)} = 1\right) \stackrel{\text{def}}{=} P_{i,\ell}, \quad \text{with} \sum_\ell P_{i,\ell} = 1.$$

Using random ratio-assignment $\mathbf{f}^{(\ell)} = \mathbf{a}^{(\ell)}/\sqrt{\sum_j \mathbf{a}_j^{(\ell)}}$, minimize the expected ratio-cut $\text{RC}(\mathcal{C}_k) = \mathbb{E}\left[\widehat{\text{RatioCut}}(\mathcal{C}_k)\right]$.

Exact expectation involves complex terms:

$$\mathbb{E}\left[(\mathbf{f}_i^{(\ell)} - \mathbf{f}_j^{(\ell)})^2\right] = (P_{i\ell} + P_{j\ell} - 2P_{i\ell}P_{j\ell})\, \underbrace{\mathbb{E}\left[\frac{1}{1 + \sum_{m \neq i,j} \mathbf{a}_m^{(\ell)}}\right]}_{\text{Hard to compute / optimize}}$$

We derive a tractable upper bound on $\text{RC}(\mathcal{C}_k)$:

$$\text{RC}(\mathcal{C}_k) \leq \mathcal{L}_{rc}(\boldsymbol{W}, \boldsymbol{P}) = \sum_{\ell=1}^{k} \frac{1}{\overline{\boldsymbol{P}}_{:,\ell}} \sum_{i,j=1}^{n} W_{ij}(P_{i\ell} + P_{j\ell} - 2P_{i\ell}P_{j\ell})$$

where $\overline{\boldsymbol{P}}_{:,\ell} = \frac{1}{n}\sum_i P_{i\ell}$ (likelihood of cluster ℓ).

Online Gradient Estimation: We derive an unbiased gradient estimator by using moving averages for $\overline{\boldsymbol{P}}$:

- Compute batch gradient $\dot{\boldsymbol{P}}$ based on $\frac{d\mathcal{L}_{rc}}{d\boldsymbol{P}}$.
- Backpropagate loss term $\propto \text{Tr}(\boldsymbol{P}^\top \text{sg}(\dot{\boldsymbol{P}}))$.

Regularization: Add KL-divergence term to prevent cluster collapse and encourage balanced clusters:

$$\mathcal{L}_{prcut}(\boldsymbol{W}, \boldsymbol{P}) = \mathcal{L}_{rc}(\boldsymbol{W}, \boldsymbol{P}) + \gamma D_{\text{KL}}(\overline{\boldsymbol{P}} \parallel \frac{1}{k}\mathbf{1}_k)$$

PRCut Algorithm

Require: Similarity kernel $\mathcal{K}$, encoder N_θ, #clusters k.

while not terminated **do**
  $t \leftarrow t + 1$
  Sample batches S_l, S_r size b.
  Compute similarities $\boldsymbol{W} = \mathcal{K}(S_l, S_r)$.
  Compute probabilities $\boldsymbol{P}_\theta^l, \boldsymbol{P}_\theta^r \leftarrow N_\theta(S_l), N_\theta(S_r)$.
  Update moving average $\overline{\boldsymbol{P}}_t$ using $\overline{\boldsymbol{P}_\theta^l}, \overline{\boldsymbol{P}_\theta^r}$.
  Compute gradients $\dot{\boldsymbol{P}}_\theta^l, \dot{\boldsymbol{P}}_\theta^r$ for $\mathcal{L}_{rc}$ using $\overline{\boldsymbol{P}}_t$.
  Backpropagate $\text{Tr}\left[\boldsymbol{P}_\theta^l \,\text{sg}(\dot{\boldsymbol{P}}_\theta^l)^\top + \boldsymbol{P}_\theta^r \,\text{sg}(\dot{\boldsymbol{P}}_\theta^r)^\top\right]$.
  Update parameters θ using accumulated gradients.
end while

Experiments: Setup & Metrics

- **Network:** Simple MLP or Linear layer on embeddings. Softmax output for $\boldsymbol{P}_\theta$.
- **Similarity** $\mathcal{K}$**:**
  - Label-based (Oracle): $W_{ij} = 1$ if $y_i = y_j$.
  - k-Nearest Neighbors (kNN) graph adjacency ($k = 150$).
  - Cosine similarity on SSL embeddings (DINOv2, CLIP-ViT, SimCLR, All4One).
- **Baselines:** Spectral Clustering (SC), VMM, Turtle.

vs Baselines

Finding: Using label-based similarity, PRCut achieves performance comparable to a supervised classifier trained with labels. Shows PRCut accurately translates similarity into clusters.

Dataset	Method	ACC	NMI
MNIST	CE	0.980	**0.943**
	PRCut	**0.987**	0.938
F-MNIST	CE	0.885	**0.803**
	PRCut	**0.887**	0.789
CIFAR10	CE	**0.582**	**0.369**
	PRCut	0.571	0.359

vs Spectral Clustering

Finding: PRCut achieves a better (lower) Ratio Cut objective and better clustering metrics (ACC, NMI) than standard Spectral Clustering using kNN similarity.

Dataset	Method	ACC	NMI	RC
MNIST	SC (Best)	0.70	0.744	170.1
	PRCut	**0.821**	**0.778**	**150.2**
F-MNIST	SC (Best)	0.596	0.593	110.2
	PRCut	**0.658**	**0.620**	**101.5**
CIFAR10	SC (Best)	0.217	0.086	479.3
	PRCut	**0.243**	**0.121**	**440.3**

สแกนที่นี่

vs SOTA Methods

Finding: PRCut is competitive with the latest methods, especially when using powerful SSL embeddings (DINOv2: -D, CLIP ViT-L/14: -V). SOTA on F-MNIST.

Dataset	Method	ACC	NMI
F-MNIST	SC	0.596	0.593
	VMM	0.712	0.688
	Turtle-D	0.764	0.723
	PRCut-D	**0.791**	**0.758**
CIFAR10	SC	0.217	0.121
	Turtle-V	0.972	0.929
	PRCut-V	**0.975**	**0.934**
CIFAR100	Turtle-D	**0.806**	**0.870**
	PRCut-D	0.789	0.856

Evaluating SSL Representations

Finding: PRCut can differentiate the quality of various SSL representations. CLIP ViT-L/14 performs best on CIFAR10, DINOv2 on CIFAR100. SimCLR embeddings are less effective for graph clustering.

Dataset	Rep	ACC	NMI
CIFAR10	Raw	0.243	0.121
	SimCLR	0.721	0.652
	All4One	0.710	0.635
	VitL-14	**0.975**	**0.934**
	DinoV2	0.774	0.797
CIFAR100	Raw	0.054	0.022
	SimCLR	0.362	0.483
	All4One	0.382	0.511
	VitL-14	0.720	0.755
	DinoV2	**0.789**	**0.856**

Future Work

- Iterative approach for a K-Means like algorithm
- Sinkhorn-Knopp for $\text{Tr}(\overline{\boldsymbol{P}}^{-1}(\mathbf{1}_{n,k} - \boldsymbol{P})^\top \boldsymbol{W}\boldsymbol{P})$
- Dynamic determination of the number of clusters k.
- Probabilistic normalized cut.