Graph Cuts with Arbitrary Size Constraints Through Optimal Transport

Chakib Fettal, Lazhar Labiod, Mohamed Nadif

GRAPH CUTS WITH ARBITRARY SIZE CONSTRAINTS THROUGH OPTIMAL TRANSPORT

Chakib Fettal
Centre Borelli UMR 9010
Université Paris Cité
Informatique CDC

Lazhar Labiod
Centre Borelli UMR 9010
Université Paris Cité

Mohamed Nadif
Centre Borelli UMR 9010
Université Paris Cité

ABSTRACT

A common way of partitioning graphs is through minimum cuts. One drawback of classical minimum cut methods is that they tend to produce small groups, which is why more balanced variants such as normalized and ratio cuts have seen more success. However, we believe that with these variants, the balance constraints can be too restrictive for some applications like for clustering of imbalanced datasets, while not being restrictive enough for when searching for perfectly balanced partitions. Here, we propose a new graph cut algorithm for partitioning graphs under arbitrary size constraints. We formulate the graph cut problem as a regularized Gromov-Wasserstein problem. We then propose to solve it using accelerated proximal GD algorithm which has global convergence guarantees, results in sparse solutions and only incurs an additional ratio of $\mathcal{O}(\log(n))$ compared to the classical spectral clustering algorithm but was seen to be more efficient.

1 Introduction

Clustering is an important task in the field of unsupervised machine learning. For example, in the context of computer vision, image clustering consists in grouping images into clusters such that the images within the same clusters are similar to each other, while those in different clusters are dissimilar. Applications are diverse and wide ranging, including, for example, content-based image retrieval [3], image annotation [9, 5], and image indexing [6]. A popular way of clustering an image dataset is through creating a graph from input images and partitioning it using techniques such as spectral clustering which solves the minimum cut (`min-cut`) problem. This is notably the case in subspace clustering where an self-representation matrix is learned according to the subspaces in which images lie and a graph is built from this matrix [33, 16, 4, 26, 49].

However, in practice, algorithms associated with the `min-cut` problem suffer from the formation of some small groups which leads to bad performance. As a result, other versions of `min-cut` were proposed that take into account the size of the resulting groups, in order to make resulting partitions more balanced. This notion of size is variable, for example, in the Normalized Cut (`ncut`) [41], size refers to the total volume of a cluster, while in the Ratio Cut (`rcut`) problem [22], it refers to the cardinality of a cluster. A common method for solving the `ncut` and `rcut` problems is the spectral clustering approach [44, 35] which is popular due to often showing good empirical performance and being somewhat efficient.

However, there some weaknesses that apply to the spectral clustering algorithms and to most approaches tackling the `ncut` and `rcut` problems. A first one is that, for some applications, the cluster balancing is not strict enough, meaning that even if we include the size regularization into the `min-cut` problem, the groups are still not necessarily of similar size, which is why several truly balanced clustering algorithm have been proposed in the literature [8, 7, 31]. Another problem is that the balance constraint is too restrictive for many real world datasets, for example, a recent trend in computer vision is to propose approaches dealing with long-tailed datasets which are datasets that contain head classes that represent most of the overall dataset and then have tail classes that represent a small fraction of the overall dataset [47, 51]. Some approaches propose integrating generic size constraints to the objective like in [21, 24, 50], however these approaches directly deal with euclidean data instead of graphs.

In this paper, we propose a novel framework that can incorporate generic size constraints in a strict manner into `min-cut` problem using Optimal Transport. We sum up our contributions in this work as follows:

- We formulate a problem for obtaining graph cuts that are balanced for an arbitrarily defined notion of size instead of specifically the volume or cardinality as is traditionally done in spectral clustering. We also propose a more general formulation of graph cuts with cluster size constraints through optimal transport. This can help with dealing with balanced and imbalanced graphs.
- We then propose a new way to solve this constrained graph cut problem using a nonconvex accelerated proximal gradient scheme which has global convergence guarantees for specific step sizes.
- Comprehensive experiments on real-life graphs and graphs built from image datasets using subspace clustering are performed. Results showcase the effectiveness of the proposed method in terms of obtaining the desired cluster sizes, clustering performance and computational efficiency.

The rest of this paper is organized as follows: Preliminaries are presented in Section 2. Some related work is discussed in section 3. The `OT-cut` problem and algorithm along with their analysis and links to prior research are given in section 4. We present experimental results and analysis in section 5. Conclusions are then given in section 6.

2 Related Work

Our work is related with balanced clustering, as the latter is a special case of it, as well as with the more generic problem of constrained clustering, and GW based graph partitioning.

2.1 Balanced Clustering.

A common class of constrained clustering problems is balanced clustering where we wish to obtain a partition with clusters of the same size. For example, [15] introduced a conscience mechanism which penalizes clusters relative to their size, [1], then employed it to develop the Frequency Sensitive Competitive Learning (FSCL) algorithm. In [31], authors proposed to leverage the exclusive lasso on the k-means and `min-cut` problems to regulate the balance degree of the clustering results. [32] proposed a simplex algorithm to solve a minimum cost flow problem similar to k-means. In [8], authors proposed a self-balanced `min-cut` algorithm for image clustering implicitly using exclusive lasso as a balance regularizer in order to produce balanced partitions.

2.2 Constrained Clustering.

Some clustering approaches with generic size constraints, which can be seen as an extension of balanced clustering, also exist. In [50], a heuristic algorithm to transform size constrained clustering problems into integer linear programming problems was proposed. Authors in [20] introduced a modified k-means algorithm which can be used to obtain clusters of preferred sizes. Clustering paradigms based on OT generally offer the possibility to set a target distribution for resulting partitions. [21] proposed a deep clustering algorithm through optimal transport with entropic regularization. In [28, 42, 18], authors proposed to tackle co-clustering and biclustering problems using OT demonstrating good empirical performance.

2.3 Gomov-Wasserstein Graph Clustering.

The Gromov-Wasserstein (GW) partitioning paradigm S-GWL [46] supposes that the Gromov-Wasserstein discrepancy can uncover the clustering structure of the observed source graph $\mathcal{G}$ when the target graph $\mathcal{G}_{dc}$ only contains weighted self-connected isolated nodes, this means that its adjacency matrix is diagonal. The weights of this diagonal matrix as well as the source and target distribution are special functions of the node degrees. Their approach uses a regularized proximal gradient method as well as a recursive partitioning scheme and can be used in a multi-view clustering setting. The problem with this approach is its sensitivity to the hyperparameter setting which is problematic since it is an unsupervised method. Another approach, SpecGWL [10] generalizes spectral clustering using Gromov-Wasserstein discrepancy and heat kernels but suffers from high computational complexity. Given a graph with n node, its optimization procedure involves the computation of a gradient which is in $O(n^3 \log(n))$ and an eigendecompostion $O(n^3)$ and therefore is not usable for large scale graphs.

3 Preliminaries

In what follows, $\Delta^n = \{\mathbf{p} \in \mathbb{R}^n_+ | \sum_{i=1}^n p_i = 1\}$ denotes the n-dimensional standard simplex. $\Pi(\mathbf{w}, \mathbf{v}) = \{\mathbf{Z} \in \mathbb{R}^{n\times k}_+ | \mathbf{Z}\mathbb{1} = \mathbf{w}, \mathbf{Z}^\top \mathbb{1} = \mathbf{v}\}$ denotes the transportation polytope, where $\mathbf{w} \in \Delta^n$ and $\mathbf{v} \in \Delta^k$ are the marginals of the joint distribution $\mathbf{Z}$ and $\mathbb{1}$ is a vector of ones, its size can be inferred from the context. Matrices are denoted with uppercase boldface letters, and vectors with lowercase boldface letters. For a matrix $\mathbf{M}$, its i-th row is $\mathbf{m}_i$ and m_{ij} is the j-th entry of row i. Tr refers to the trace of a square matrix.

3.1 Graph Cuts and Spectral Clustering

Minimum-cut Problem. Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with an weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ with $n = |\mathcal{V}|$, a cut is a partition of its vertices $\mathcal{V}$ into two disjoint subsets $\mathcal{A}$ and $\bar{\mathcal{A}}$. The value of a cut is given by

$$\text{cut}(\mathcal{A}) = \sum_{v_i \in \mathcal{A}, v_j \in \bar{\mathcal{A}}} w_{ij}. \tag{1}$$

The goal of the minimum k-cut problem is to find a partition $(\mathcal{A}_1, ..., \mathcal{A}_k)$ of the set of vertices $\mathcal{V}$ into k different groups that is minimal in some metric. Intuitively, we wish for the edges between different subsets to have small weights, and for the edges within a subset have large weights. Formally, it is defined as

$$\texttt{min-cut}(\mathbf{W}, k) = \min_{\mathcal{A}_1,...,\mathcal{A}_k} \sum_{i=1}^k \text{cut}(\mathcal{A}_i). \tag{2}$$

This problem can also be stated as a trace minimization problem by representing the resulting partition $\mathcal{A}_1, \ldots, \mathcal{A}_k$ using an assignment matrix $\mathbf{X}$ such that for each row i, we have that

$$x_{ij} = \begin{cases} 1 & \text{if vertex } i \text{ is in } \mathcal{A}_j, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

This condition is equivalent to introducing two constraints which are $\mathbf{X} \in \{0, 1\}^{n\times k}$ and $\mathbf{X}\mathbb{1} = \mathbb{1}$. The minimum k-cut problem can then be formulated as

$$\texttt{min-cut}(\mathbf{W}, k) = \min_{\substack{\mathbf{X}\in\{0,1\}^{n\times k} \\ \mathbf{X}\mathbb{1}=\mathbb{1}}} \text{Tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X}), \tag{4}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ refers to the graph Laplacian of the graph $\mathcal{G}$ and $\mathbf{D}$ is the diagonal matrix of degree of $\mathbf{W}$, i.e., $d_{ii} = \sum_j w_{ij}$.

Normalized k-Cut Problem. In practice, solutions to the minimum k-cut problem do not yield satisfactory partitions due to the formation of small groups of vertices. Consequently, versions of the problem that take into account some notion of "size" for these groups have been proposed. The most commonly used one is normalized cut [41]:

$$\texttt{ncut}(\mathbf{W}, k) = \min_{\mathcal{A}_1,...,\mathcal{A}_k} \sum_{i=1}^k \frac{\text{cut}(\mathcal{A}_i)}{\text{vol}(\mathcal{A}_i)}, \tag{5}$$

where the volume can be conveniently written as $\text{vol}(\mathcal{A}_i) = \mathbf{x}_i^T \mathbf{D}\mathbf{x}_i$. Another variant which is referred to as the ratio cut problem due to the different groups being normalized by their cardinality instead of their volumes:

$$\texttt{rcut}(W, k) = \min_{\mathcal{A}_1,...,\mathcal{A}_k} \sum_{i=1}^k \frac{\text{cut}(\mathcal{A}_i)}{|\mathcal{A}_i|}, \tag{6}$$

where $|\mathcal{A}_i| = \mathbf{x}_i^T \mathbf{x}_i$. This variant can be recovered from the normalized graph cut problem by replacing $\mathbf{D}$ with $\mathbf{I}$ in the computation of the volume.

Spectral Clustering. A common approach to solving the normalized graph cut problems, spectral clustering, relaxes the partition constraints on $\mathbf{X}$ and instead considers a form of semi-orthogonality constraints. In the case of `rcut`, we have `rcut` written as a trace optimization problem:

$$\texttt{rcut}(\mathbf{W}, k) = \min_{\substack{\mathbf{X}\in\mathbb{R}^{n\times k} \\ \mathbf{X}^\top\mathbf{X}=\mathbf{I}}} \text{Tr}\left(\mathbf{X}^\top \mathbf{L}\mathbf{X}\right). \tag{7}$$

On the other hand for ncut, the partition matrix $\mathbf{X}$ is substituted with $\mathbf{H} = \mathbf{D}^{1/2}\mathbf{X}$ and a semi-orthogonality constraint is placed on this $\mathbf{H}$, i.e.,

$$\texttt{ncut}(\mathbf{W}, k) = \min_{\substack{\mathbf{H}\in\mathbb{R}^{n\times k} \\ \mathbf{H}^\top\mathbf{H}=\mathbf{I}}} \quad \mathrm{Tr}\left(\mathbf{H}^\top\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}\mathbf{H}\right). \tag{8}$$

A solution $\mathbf{H}$ for the ncut problem is formed by stacking the first k-eigenvectors of the symmetrically normalized Laplacian $\mathbf{L}_s = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$ as its columns, and then applying a clustering algorithm such as k-means on its rows and assign the original data points accordingly [35]. The principle is the same for solving rcut but instead using the unnormalized Laplacian.

3.2 Optimal Transport

Discrete optimal transport. The goal of the optimal transport problem is to find a minimal cost transport plan $\mathbf{X}$ between a source probability distribution of $\mathbf{w}$ and a target probability distribution $\mathbf{v}$. Here we are interested in the discrete Kantorovich formulation of OT. When dealing with discrete probability distributions, said formulation is

$$\texttt{OT}(\mathbf{M}, \mathbf{w}, \mathbf{v}) \triangleq \min_{\mathbf{X}\in\Pi(\mathbf{w},\mathbf{v})} \langle \mathbf{M}, \mathbf{X}\rangle, \tag{9}$$

where $\mathbf{M} \in \mathbb{R}^{n\times k}$ is the cost matrix, and c_{ij} quantifies the effort needed to transport a probability mass from $\mathbf{w}_i$ to $\mathbf{v}_j$. Regularization can be introduced to further speed up computation of OT. Examples include entropic regularization [12] and low-rank regularization [40].

Discrete Gromov-Wasserstein Discrepancy. The discrete Gromov-Wasserstein (GW) discrepancy [39] is an extension of optimal transport to the case where the source and target distributions are defined on different metric spaces:

$$\mathrm{GW}(\mathbf{M}, \mathbf{M}', \mathbf{w}, \mathbf{v}) \triangleq \min_{\mathbf{X}\in\Pi(\mathbf{w},\mathbf{v})} \langle L(\mathbf{M}, \mathbf{M}') \otimes \mathbf{X}, \mathbf{X}\rangle \tag{10}$$

where $\mathbf{M} \in \mathbb{R}^{n\times n}$ and $\mathbf{M}' \in \mathbb{R}^{k\times k}$ are similarity matrices defined on the source space and target space respectively, and $L : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ is a divergence measure between scalars, $L(\mathbf{M}, \mathbf{M}')$ is the $n \times n \times k \times k$ tensor of all pairwise divergences between the elements of $\mathbf{M}$ and $\mathbf{M}'$. $\otimes$ denotes tensor-matrix product.

4 Proposed Methodology

In this section, we derive our OT-based constrained graph cut problem and propose a nonconvex proximal GD algorithm with global convergence guarantees for its resolution.

4.1 Normalized Cuts via Optimal Transport

As already mentioned, the good performance of the normalized cut algorithm comes from the normalization by the volume of each group in the cut. However, the size constraint is not a hard one, meaning that obtained groups are not of exactly the same volume. This leads us to propose to replace the volume normalization by a strict balancing constraint as follows:

$$\min_{\mathcal{A}_1,\ldots,\mathcal{A}_k} \sum_{i=1}^{k} \mathrm{cut}(\mathcal{A}_i) \;\text{ s.t. }\; \mathrm{vol}(\mathcal{A}_1) = \ldots = \mathrm{vol}(\mathcal{A}_k). \tag{11}$$

this problem can be rewritten as the following trace minimization problem:

$$\begin{aligned} &\min \quad \mathrm{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) \\ &\text{subject to:} \\ &\begin{cases} \mathbf{X}\mathbb{1} = \mathbf{D}\mathbb{1}, & (\mathbf{x}_i \text{ sums to the degree of node } i) \\ \mathbf{X}^\top\mathbb{1} = \dfrac{\sum_i d_{ii}}{k}\mathbb{1}, & (\text{clusters are balanced w.r.t degrees}) \\ \forall_i \|\mathbf{x}_i\|_0 = 1 & (\text{a node belongs to a unique cluster.}) \end{cases} \end{aligned} \tag{12}$$

Here, $\|.\|_0$ is the zero norm that returns the number of nonzero elements in its argument. This problem may not have feasible solutions. However, by dropping the third constraint, this problem becomes an instance of the Gromov-Wasserstein problem which is always feasible. Specifically, the first and second constraints are equivalent to defining $\mathbf{X}$ to be an element of the transportation polytope with a uniform target distribution and a source distribution consisting of

the degrees of the nodes. These degrees can be represented as proportions instead of absolute quantities by dividing them over their sum, yielding the following problem:

$$\min_{\mathbf{X}\in\Pi\left(\frac{1}{\sum_i d_{ii}}\mathbf{D}\mathbb{1},\frac{1}{k}\mathbb{1}\right)} \quad \mathrm{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) \tag{13}$$

This formulation is a special case of the Gromov-Wasserstein problem for a source space whose similarity matrix in the initial space is $\mathbf{M} = -\mathbf{L}$ and whose similarity matrix in the destination space is $\mathbf{M}' = -\mathbf{I}$. Note that a ratio cut version can be obtained by replacing the volume constraint with

$$|\mathcal{A}_1| = \ldots = |\mathcal{A}_k| \tag{14}$$

in problem 11, and similarly in problem 13, by substituting the identity matrix$\mathbf{I}$ for the degree matrix $\mathbf{D}$, giving rise to:

$$\min_{\mathbf{X}\in\Pi\left(\frac{1}{n}\mathbb{1},\frac{1}{k}\mathbb{1}\right)} \quad \mathrm{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) \tag{15}$$

4.2 Graph Cuts with Arbitrary Size Constraints

From the previous problem, it is easy to see that target distribution does not need to be uniform, and as such, any distribution can be considered, leading to further applications like imbalanced dataset clustering. Another observation is that any notion of size can be considered and not only the volume or cardinality of the formed node groups. We formulate an initial version of the generic optimal transport graph cut problem as:

$$\min_{\mathbf{X}\in\Pi(\boldsymbol{\pi}^s,\boldsymbol{\pi}^t)} \quad \mathrm{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) \;\equiv\; \min_{\mathbf{X}\in\Pi(\boldsymbol{\pi}^s,\boldsymbol{\pi}^t)} \quad \langle\mathbf{L}\mathbf{X},\mathbf{X}\rangle\,, \tag{16}$$

where $\boldsymbol{\pi}^s_i$ is the relative 'size' of the element i and $\boldsymbol{\pi}^t_j$ is the desired relative 'size' of the group j. Through the form that uses the Frobenius product, it is easy to see how our problem is related to the Gromov-Wasserstein problem. These size parameters can either be set using domain knowledge by the expert using our algorithm or by trying mutliple random guesses and then selecting the best one via internal clustering quality metrics such as Davies-Bouldin index [13]. Another way would be to consider the cluster distribution generated by another algorithm such as spectral clustering.

4.3 Regularization for Sparse Solutions

We wish to obtain sparse solutions in order to easily interpret them as partition matrices of the input graph. We do so by aiming to find solutions over the extreme points of the transportation polytope which are matrices that have at most $n + k - 1$ non-zero entries [38]. We do so by introducing a regularization term to problem 16. Consequently, we consider the following problem which we coin `OT-cut`:

$$\texttt{OT-cut}(\mathbf{X},\pi_s,\pi_t) \equiv \min_{\mathbf{X}\in\Pi(\boldsymbol{\pi}^s,\boldsymbol{\pi}^t)} \mathrm{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) \;-\; \lambda\|\mathbf{X}\|^2 \tag{17}$$

where $\lambda \in \mathbb{R}^+$ is the regularization trade-off parameter. It should be noted that our regularizer is concave. We also define two special cases of this problem, which are based on the `ncut` and `rcut` problems. The first one which we call `OT-ncut` is obtained by fixing the hyper-parameter $\pi_s = \frac{1}{\sum_i d_{ii}}\mathbf{D}\mathbb{1}$ while the second one `OT-rcut` is obtained by substituting the $\mathbf{D}$ in the previous formula with $\mathbf{I}$ and forcing the target to be uniform. Figure 1 shows the evolution of the objective on different datasets.

4.4 Optimization, Convergence and Complexity

We wish to solve problem 17 which is nonconvex, but algorithms with convergence guarantees exist for problems of this form. Specifically, we will be using a nonconvex proximal gradient descent based on [30]. The pseudocode is given in algorithm 1.

Proposition 1. *For step size $\alpha = \frac{1}{2\lambda}$, the iterates $\mathbf{X}^{(t)}$ generated by the nonconvex PGD algorithm for our problem are all extreme points of the transporation polytope, and as such, have at most $n + k - 1$ nonzero entries.*

Proof. Problem 17 can be equivalently stated by writing the constraint as a term in the loss function:

$$\min_{\mathbf{X}} \quad \underbrace{\mathrm{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X})}_{f(\mathbf{X})} + \underbrace{I_{\Pi(\boldsymbol{\pi}^s,\boldsymbol{\pi}^t)}(\mathbf{X}) - \lambda\|\mathbf{X}\|^2}_{g(\mathbf{X})} \tag{18}$$

where $I_{\mathcal{C}}$ is the characteristic function of set $\mathcal{C}$ i.e.

$$I_{\mathcal{C}} = \begin{cases} 0, & \text{if } \mathbf{X} \in \mathcal{C}, \\ +\infty, & \text{if } \mathbf{X} \notin \mathcal{C}. \end{cases}$$

Since we use a proximal descent scheme, we show how to compute the proximal operator for our loss function:

$$\begin{aligned} &\texttt{prox}_{\alpha g}\left(\mathbf{X}^{(t)} - \alpha \nabla f(\mathbf{X}^{(t)})\right) = \texttt{prox}_{\alpha g}\left(\mathbf{X}^{(t)} - \alpha \nabla \operatorname{Tr}\left(\mathbf{X}^{(t)}\mathbf{L}\mathbf{X}^{(t)}\right)\right) \\ &= \texttt{prox}_{\alpha(I_{\Pi(\boldsymbol{\pi}^s,\boldsymbol{\pi}^t)} - \alpha\|.\|^2)}\left((\mathbf{I} - 2\alpha\mathbf{L})\mathbf{X}^{(t)}\right) = \underset{\mathbf{Z}\in\Pi(\boldsymbol{\pi}^s,\boldsymbol{\pi}^t)}{\arg\min} \frac{1}{2\alpha}\left\|\mathbf{Z} - (\mathbf{I} - 2\alpha\mathbf{L})\mathbf{X}^{(t)}\right\|^2 - \lambda\|\mathbf{Z}\|^2 \\ &= \underset{\mathbf{Z}\in\Pi(\boldsymbol{\pi}^s,\boldsymbol{\pi}^t)}{\arg\min} \frac{1}{2\alpha}\|\mathbf{Z}\|^2 + \frac{1}{2\alpha}\left\|(\mathbf{I} - 2\alpha\mathbf{L})\mathbf{X}^{(t)}\right\|^2 - \frac{1}{\alpha}\operatorname{Tr}\left(\mathbf{Z}^\top(\mathbf{I} - 2\alpha\mathbf{L})\mathbf{X}^{(t)}\right) - \lambda\|\mathbf{Z}\|^2. \end{aligned}$$

We assumed that $\alpha = \frac{1}{2\lambda}$, by substituting for λ into the previous formula and dropping the constant term, we obtain:

$$\underset{\mathbf{Z}\in\Pi(\boldsymbol{\pi}^s,\boldsymbol{\pi}^t)}{\arg\min} \operatorname{Tr}\left(\mathbf{Z}^\top(2\alpha\mathbf{L} - \mathbf{I})\mathbf{X}^{(t)}\right) = \underset{\mathbf{Z}\in\Pi(\boldsymbol{\pi}^s,\boldsymbol{\pi}^t)}{\arg\min} \left\langle \mathbf{Z},\ (2\alpha\mathbf{L} - \mathbf{I})\mathbf{X}^{(t)}\right\rangle.$$

This is the classical OT problem. Its resolution is possible by stating it as the earth-mover's distance (EMD) linear program [23] and using the network simplex algorithm. □

Proposition 2. *Algorithm 1 globally converges for step size $\alpha < \frac{1}{s}$ where s is the smoothness constant of Tr$(\mathbf{X}^\top\mathbf{L}\mathbf{X})$.*

Proof. Here, we have that f is proper and s-smooth i.e. ∇f is s-Lipschitz. g is proper and lower semi-continuous. Additionally, $f + g$ is coercive. Then, according to [30], nonconvex accelerated proximal GD globally converges for $\alpha < \frac{1}{s}$. □

Proposition 3. *For a graph with n nodes, the complexity of an iteration of the proposed algorithm is $\mathcal{O}\left(kn^2\log n\right)$.*

Proof. We note that in practice $n >> k$ and that the complexity of the network simplex algorithm for some graph $\mathcal{G}_{EMD} = (\mathcal{V}_{EMD}, \mathcal{V}_{EMD})$ is in $\mathcal{O}(|\mathcal{V}_{EMD}||\mathcal{E}_{EMD}|\log|\mathcal{E}_{EMD}|)$ [36]. In our case, this graph has $|\mathcal{V}_{EMD}| = n + k$ (since $n >> k$, we can drop the k) and $|\mathcal{E}_{EMD}| = nk$. The other operation that is performed during each iteration is the matrix multiplication whose complexity is in $\mathcal{O}(k|\mathcal{E}|)$ where $|\mathcal{E}|$ is the number of edges in the original graph. In the worst case when matrix $\mathbf{L}$ is fully dense, we have that $|\mathcal{E}| = n^2$. □

Algorithm 1: Nonconvex Accelerated PGD for `OT-cut`

Data: $\mathbf{A}$ Adjacency matrix, $\boldsymbol{\pi}^s$ node size distribution, $\boldsymbol{\pi}^t$ cluster size distribution, $\mathbf{G}_{init}$ initial partition matrix, $\alpha = \frac{1}{2\lambda} < \frac{1}{s}$ step size, *maxIter* maximum number of iterations.
Result: $\mathbf{G}$ partition of the graph.
Construct Laplacian matrix $\mathbf{L}$ from the adjacency matrix $\mathbf{A}$;
$\mathbf{X}^{(0)} \leftarrow \arg\texttt{OT}\left(\mathbf{G}_{init}, \boldsymbol{\pi}^s, \boldsymbol{\pi}^t\right)$;
$\mathbf{Z}^{(1)} \leftarrow \mathbf{X}^{(0)}, \mathbf{X}^{(1)} \leftarrow \mathbf{X}^{(0)}$;
$c_0 \leftarrow 0, c_1 \leftarrow 1$;
while *maxIter not reached* **do**
  $\mathbf{Y}^{(t)} = \mathbf{X}^{(t)} + \frac{c_{t-1}}{c_t}(\mathbf{Z}^{(t)} - \mathbf{X}^{(t)}) + \frac{c_{t-1}-1}{c_t}(\mathbf{X}^{(t)} - \mathbf{X}^{(t-1)})$;
  $\mathbf{Z}^{(t+1)} := \arg\texttt{OT}\left((2\alpha\mathbf{L} - \mathbf{I})\mathbf{Y}^{(t)}, \boldsymbol{\pi}^s, \boldsymbol{\pi}^t\right)$;
  $\mathbf{V}^{(t+1)} := \arg\texttt{OT}\left((2\alpha\mathbf{L} - \mathbf{I})\mathbf{X}^{(t)}, \boldsymbol{\pi}^s, \boldsymbol{\pi}^t\right)$;
  $c_{t+1} = (\sqrt{4c_t^2 + 1} + 1)/2$;
  $\mathbf{X}^{(t+1)} = \begin{cases} \mathbf{Z}^{(t+1)}, & \text{if loss}\left(\mathbf{Z}^{(t+1)}\right) < \text{loss}\left(\mathbf{V}^{(t+1)}\right) \\ \mathbf{V}^{(t+1)}, & \text{otherwise.} \end{cases}$;
Generate partition matrix $\mathbf{G}$ by assigning each node i to the $(\arg\max_i \mathbf{x}_i)$-th partition.;

5 Experiments

We evaluated the clustering performance of our two variants `OT-ncut` and `OT-rcut` algorithms against the spectral clustering algorithm and state-of-the-art OT-based graph clustering approaches.

Table 1: Dataset Statistics. The balance ratio is the ratio of the most frequent class over the least frequent one.

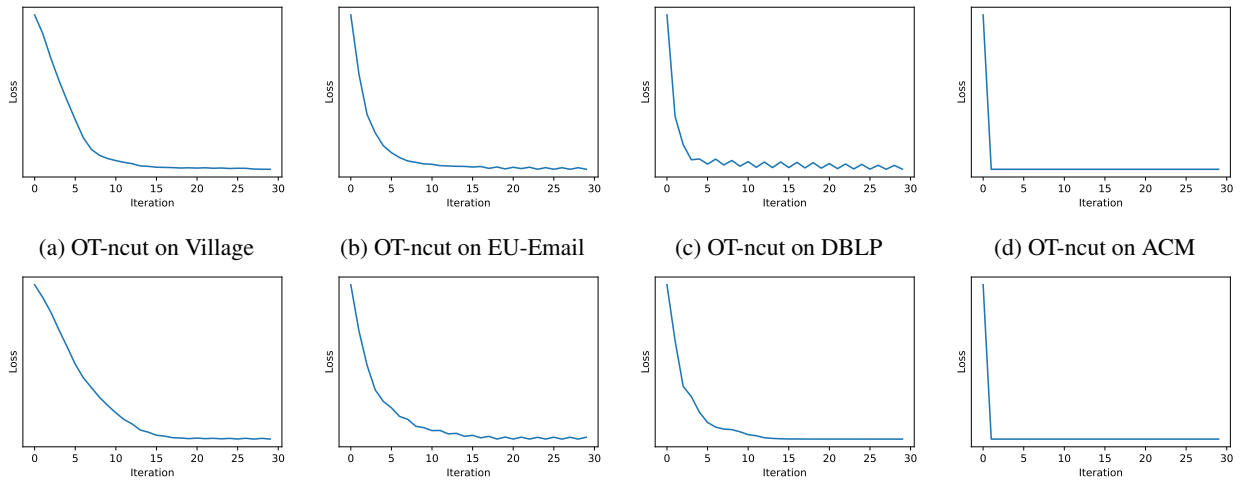
Type	Dataset	Nodes	Graph & Edges	Sparsity	Clusters	Balance Ratio
Graphs Built From Images	MNIST	10,000	LRSC (100,000,000)	0.0%	10	1.272
			LSR (100,000,000)	0.0%		
			ENSC (785,744)	99.2%		
	Fashion-MNIST	10,000	LRSC (100,000,000)	0.0%	10	1.0
			LSR (100,000,000)	0.0%		
			ENSC (458,390)	99.5%		
	KMNIST	10,000	LSR (100,000,000)	0.0%	10	1.0
			LRSC (100,000,000)	0.0%		
			ENSC (817,124)	99.2%		
Naturally Occuring Graphs	ACM	3,025	2,210,761	75.8%	3	1.099
	DBLP	4,057	6,772,278	58.9%	4	1.607
	Village	1,991	16,800	99.6%	12	3.792
	EU-Email	1,005	32,770	96.8%	42	109.0

Table 2: Average (±sd) clustering performance and running times on the graph built from images. The best performance is highlighted in bold, the lowest (highest) runtime is highlighted in blue (red).

		MNIST		Fashion-MNIST		KMNIST	
Graph	Method	ARI	Time	ARI	Time	ARI	Time
LRSC	Spectral	0.4134 ±0.0003	10.28	0.1742 ±0.0003	10.51	0.4067 ±0.0	9.83
	S-GWL	0.0488 ±0.0	7.88	0.0188 ±0.0	7.84	0.0560 ±0.0	7.98
	SpecGWL	0.0248 ±0.0	453.19	0.0111 ±0.0	397.19	0.0145 ±0.0	383.23
	OT-rcut	0.4516 ±0.0273	5.58	0.2231 ±0.0051	5.82	**0.4157** ±0.0154	6.15
	OT-ncut	**0.4751** ±0.0383	6.12	**0.2291** ±0.0148	6.11	0.3832 ±0.0279	5.88
LSR	Spectral	0.311 ±0.0002	8.82	0.1486 ±0.0001	10.19	0.3631 ±0.0001	9.72
	S-GWL	0.0628 ±0.0	8.2	0.0357 ±0.0	7.93	0.0593 ±0.0	8.01
	SpecGWL	0.1127 ±0.0	454.06	0.0341 ±0.0	454.26	0.0267 ±0.0	407.55
	OT-rcut	**0.3723** ±0.0377	6.23	0.1771 ±0.0164	6.61	**0.4335** ±0.0105	6.01
	OT-ncut	0.3458 ±0.0267	5.77	**0.1915** ±0.0131	5.72	0.4301 ±0.0075	5.87
ENSC	Spectral	0.1206 ±0.0001	13.28	0.1164 ±0.0	12.62	**0.4321** ±0.0007	13.6
	S-GWL	0.0798 ±0.0	8.05	0.0362 ±0.0	7.85	0.0422 ±0.0	7.96
	SpecGWL	**0.5444** ±0.0	268.12	0.1082 ±0.0	288.75	0.4020 ±0.0	287.06
	OT-rcut	0.4228 ±0.0694	5.47	0.2113 ±0.0257	6.18	0.2924 ±0.0589	6.27
	OT-ncut	0.3882 ±0.0718	5.68	**0.2251** ±0.0191	5.77	0.2771 ±0.0226	5.83

Table 3: Average (±sd) clustering performance and running times on the graph datasets. Same legend as for Table 2.

	EU-Email		Village		ACM		DBLP	
Method	ARI	Time	ARI	Time	ARI	Time	ARI	Time
Spectral	0.2445 ±0.0133	1.13	0.3892 ±0.1934	0.76	0.1599 ±0.003	1.83	0.0039 ±0.0053	16.49
S-GWL	0.2684 ±0.0	2.11	0.5333 ±0.0	4.26	0.1873 ±0.0	2.09	0.0 ±0.0	18.42
SpecGWL	0.1125 ±0.0	0.59	0.5887 ±0.0	0.89	0.008 ±0.0	4.65	0.2891 ±0.0	13.66
OT-rcut	0.2629 ±0.0096	0.22	**0.5969** ±0.0505	0.27	**0.2643** ±0.0249	0.69	**0.3119** ±0.0279	1.45
OT-ncut	**0.2687** ±0.0094	0.20	0.4819 ±0.0369	0.30	0.2167 ±0.045	0.77	0.1721 ±0.0674	1.33



(a) OT-ncut on Village (b) OT-ncut on EU-Email (c) OT-ncut on DBLP (d) OT-ncut on ACM

(e) OT-rcut on Village (f) OT-rcut on EU-Email (g) OT-rcut on DBLP (h) OT-rcut on ACM

Figure 1: Evolution of the objective as function of the number of iterations.

Table 4: Faithfulness to the constraints: KL divergence between the desired and the resulting cluster distributions. A value of zero reflects a perfect match between the constraint and the result.

Dataset	Graph	OT-rcut	OT-ncut
MNIST	LRSC	0.0 ±0.0	0.0001 ±0.0001
	LSR	0.0 ±0.0	0.0001 ±0.0001
	ENSC	0.0 ±0.0	0.0 ±0.0
Fashion-MNIST	LRSC	0.0 ±0.0	0.0001 ±0.0001
	LSR	0.0 ±0.0	0.0001 ±0.0001
	ENSC	0.0 ±0.0	0.0 ±0.0

Dataset	Graph	OT-rcut	OT-ncut
KMNIST	LRSC	0.0 ±0.0	0.0 ±0.0
	LSR	0.0 ±0.0	0.0001 ±0.0001
	ENSC	0.0 ±0.0	0.0 ±0.0
ACM		0.0 ±0.0	0.0 ±0.0
DBLP		0.0 ±0.0	0.00.0 ±0.0021
Village		0.0 ±0.0	0.0011 ±0.0027
EU-Email		0.0 ±0.0	0.0004 ±0.0007

5.1 Datasets

We perform experiments on graphs constructed from image datasets, namely, MNIST [14], Fashion-MNIST [45] and KMNIST [11]. We generate these graphs using three subspace clustering approaches: low-rank subspace clustering (LRSC) [43], least-square regression subspace clustering (LSR) [33] and elastic net subspace clustering (ENSC) [48]. We also consider four graph datasets: DBLP, a co-term citation network; and ACM, a co-author citation networks [17]. EU-Email an email network from a large European research institution [29]. Indian-Village describes interactions among villagers in Indian villages [2]. The statistical summaries of these datasets are available in Table 1.

5.2 Performance Metrics

We adopt Adjusted Rand Index (ARI) [25] to evaluate clustering performance. It takes values between 1 and -0.5; larger values signify better performance. To evaluate the concordance of the desired and the obtained cluster distributions, we use the Kullback-Leibler (KL) divergence [27]. The KL divergence between two perfectly matching distributions will be equal to zero. Otherwise, it would be greater than zero. Smaller KL values signify better concordance.

5.3 Experimental settings

Our two variants, `OT-ncut` and `OT-rcut` are implemented via the Python optimal transport package (POT) [19]. We use random initialization and use uniform target distributions unless explicitly stated otherwise. We also set $\alpha = 2$ and the maximum number of iterations to 30. We also use normalized laplacian matrices. For the comparative approaches, we use the Scikit-Learn[37] implementation of spectral clustering. For OT-based methods, S-GWL [46] and SpecGWL

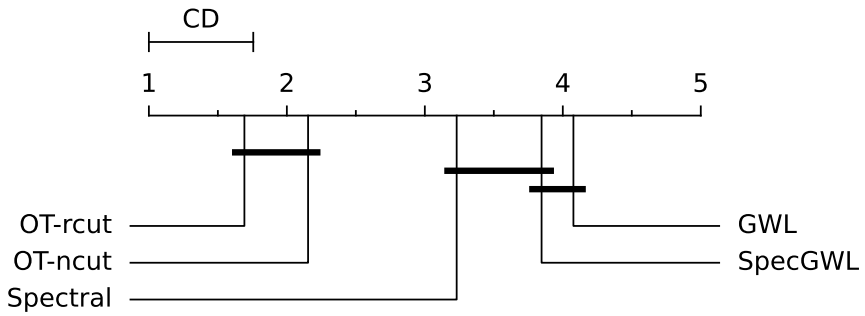
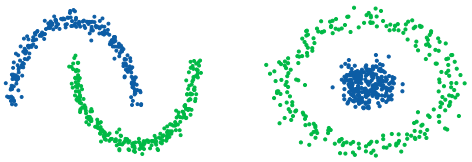


Figure 3: Neményi post-hoc rank test results. OT-rcut and OT-ncut outperform the comparative approaches in a statistically significant manner while having comparable performance.

[10], we use their official implementations as they are open source. All experiments were run five times and were performed on a 64gb RAM machine with a 12th Gen Intel(R) Core(TM) i9-12950HX (24 CPUs) processor with a frequency of 2.3GHz.

5.4 Experimental Results

Toy Examples. Our algorithm deals with a graph cut-like criterion which means that it should partition a dataset according to its connectivity. This means that it should work on datasets on which metric clustering approaches such as k-means fail. Two toy examples are given in Figure 2a and Figure 2b.



(a) OT-ncut on a a graph built using a k-nn graph. (b) OT-rcut on a graph built using RBF kernel.

Figure 2: OT-ncut and OT-rcut results on toy datasets.

Clustering Performance. Table 3 shows the clustering performance on the graph datasets. In all cases, one of our two variants has the best results in terms of ARI. Table 3 presents the clustering performance on the graph datasets. In all cases, one of our two variants has the best results in terms of ARI. Table 2 describes results obtained on image graph datasets. One of our two variants gives the best results on all three datasets with the graphs generated by LRSC and LSR. On the graphs generated by ENSC, the best result is obtained only on Fashion-MNIST while SpecGWL has the best results on MNIST. Spectral clustering gives the best performance on KMNIST. Note that better results can also be obtained with our variants by trading-off some computational efficiency. Specifically, this can be done by using several different initializations and taking the one that leads to minimizing the objective the most.

Statistical Significance Testing Figure 3 shows the performance ranks of the different methods averaged over all the runs on the datasets we considered in terms of ARI. The Neményi post-hoc rank test [34] shows that OT-rcut and OT-ncut perform similarly and outperform the other approaches for a confidence level of 95%. Other approaches perform similarly.

Concordance of the Desired & Resulting Cluster Sizes. To evaluate our algorithm's ability to produce a partition with the desired group size distribution, we use the KL divergence metric. Specifically, we compare the distribution obtained by our OT-rcut and OT-ncut variants against the target distribution specified as a hyperparameter ($\boldsymbol{\pi}^t$). Table 4 presents the KL divergences for both variants on various datasets. Predictably, our approaches achieve near-perfect performance on most datasets. Notably, OT-rcut is always able to perfectly recover the desired group sizes. This has to do with the fact that, up to a constant, all the entries in the solutions to the `rcut` problem are integers. This is not necessarily the case for `ncut` but the KL divergence is still very small due to the sparsity of the solutions.

Running Times. As shown in Table 3 and Table 2, OT-ncut and OT-rcut are the fastest in terms of execution times compared to other approaches on all datasets. As the graphs got larger, SpecGWL consistently had the largest runtimes. The difference is greater on larger datasets especially when they are represented as sparse matrices due to the fact that matrix multiplication is faster with sparse representations. Smaller execution times can also be obtained with our approaches by using sparse representations of the solutions.

6 Conclusion

In this paper we proposed a new graph cut algorithm for partitioning with arbitrary size constraints through optimal transport. This approach generalizes the concept of the normalized and ratio cut to arbitrary size distributions and this for any notion of size. We derived an algorithm that results in sparse solutions and has global convergence guarantees. Experiments on balanced and imbalanced datasets showed the superiority of our approach both in terms of clustering performance and empirical execution times compared to spectral clustering and other OT-based graph clustering approaches. They also demonstrated our approach's ability to recover partitions that match the desired ones which is valuable for practical problems where we wish to obtain balanced or constrained partitions.

References

[1] S. C. Ahalt, A. K. Krishnamurthy, P. Chen, and D. E. Melton. Competitive learning algorithms for vector quantization. *Neural networks*, 3(3):277–290, 1990.

[2] A. Banerjee, A. G. Chandrasekhar, E. Duflo, and M. O. Jackson. The diffusion of microfinance. *Science*, 341(6144):1236498, 2013.

[3] A. K. Bhunia, Y. Yang, T. M. Hospedales, T. Xiang, and Y.-Z. Song. Sketch less for more: On-the-fly fine-grained sketch-based image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9779–9788, 2020.

[4] J. Cai, J. Fan, W. Guo, S. Wang, Y. Zhang, and Z. Zhang. Efficient deep embedded subspace clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2022.

[5] X. Cai, F. Nie, W. Cai, and H. Huang. New graph structured sparsity model for multi-label image annotations. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 801–808, 2013.

[6] J. Cao, Z. Wu, J. Wu, and W. Liu. Towards information-theoretic k-means clustering for image indexing. *Signal Processing*, 93(7):2026–2037, 2013.

[7] X. Chen, R. Chen, Q. Wu, Y. Fang, F. Nie, and J. Z. Huang. Labin: balanced min cut for large-scale data. *IEEE transactions on neural networks and learning systems*, 31(3):725–736, 2019.

[8] X. Chen, J. Zhexue Haung, F. Nie, R. Chen, and Q. Wu. A self-balanced min-cut algorithm for image clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2061–2069, 2017.

[9] Q. Cheng, Q. Zhang, P. Fu, C. Tu, and S. Li. A survey and analysis on automatic image annotation. *Pattern Recognition*, 79:242–259, 2018.

[10] S. Chowdhury and T. Needham. Generalized spectral clustering via gromov-wasserstein learning. In *International Conference on Artificial Intelligence and Statistics*, pages 712–720. PMLR, 2021.

[11] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

[12] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[13] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

[14] L. Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[15] D. DeSieno. Adding a conscience to competitive learning. In *ICNN*, volume 1, 1988.

[16] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781, 2013.

[17] S. Fan, X. Wang, C. Shi, E. Lu, K. Lin, and B. Wang. One2multi graph autoencoder for multi-view graph clustering. In *proceedings of the web conference 2020*, pages 3070–3076, 2020.

[18] C. Fettal, L. Labiod, and M. Nadif. Efficient and effective optimal transport-based biclustering. *Advances in Neural Information Processing Systems*, 35:32989–33000, 2022.

[19] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, L. Gautheron, N. T. Gayraud, H. Janati, A. Rakotomamonjy, I. Redko, A. Rolet, A. Schutz, V. Seguy, D. J. Sutherland, R. Tavenard, A. Tong, and T. Vayer. Pot: Python optimal transport. *The Journal of Machine Learning Research*, 22(1), jan 2021.

[20] N. Ganganath, C.-T. Cheng, and K. T. Chi. Data clustering with cluster size constraints using a modified k-means algorithm. In *2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pages 158–161. IEEE, 2014.

[21] A. Genevay, G. Dulac-Arnold, and J.-P. Vert. Differentiable deep clustering with cluster size constraints. *arXiv preprint arXiv:1910.09036*, 2019.

[22] L. Hagen and A. B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE transactions on computer-aided design of integrated circuits and systems*, 11(9):1074–1085, 1992.

[23] F. L. Hitchcock. The distribution of a product from several sources to numerous localities. *Journal of mathematics and physics*, 20(1-4):224–230, 1941.

[24] F. Höppner and F. Klawonn. Clustering with size constraints. In *Computational Intelligence Paradigms*, pages 167–180. Springer, 2008.

[25] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[26] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid. Deep subspace clustering networks. *Advances in neural information processing systems*, 30, 2017.

[27] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[28] C. Laclau, I. Redko, B. Matei, Y. Bennani, and V. Brault. Co-clustering through optimal transport. In *International conference on machine learning*, pages 1955–1964. PMLR, 2017.

[29] J. Leskovec and A. Krevl. Snap datasets: Stanford large network dataset collection, 2014.

[30] H. Li and Z. Lin. Accelerated proximal gradient methods for nonconvex programming. *Advances in neural information processing systems*, 28, 2015.

[31] Z. Li, F. Nie, X. Chang, Z. Ma, and Y. Yang. Balanced clustering via exclusive lasso: A pragmatic approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[32] W. Lin, Z. He, and M. Xiao. Balanced clustering: A uniform model and fast algorithm. In *IJCAI*, pages 2987–2993, 2019.

[33] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan. Robust and efficient subspace segmentation via least squares regression. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII 12*, pages 347–360. Springer, 2012.

[34] P. B. Nemenyi. *Distribution-free multiple comparisons.* Princeton University, 1963.

[35] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001.

[36] J. B. Orlin. A polynomial time primal network simplex algorithm for minimum cost flows. *Mathematical Programming*, 78(2):109–129, 1997.

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[38] G. Peyré, M. Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[39] G. Peyré, M. Cuturi, and J. Solomon. Gromov-wasserstein averaging of kernel and distance matrices. In *International conference on machine learning*, pages 2664–2672. PMLR, 2016.

[40] M. Scetbon and marco cuturi. Low-rank optimal transport: Approximation, statistics and debiasing. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[41] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[42] V. Titouan, I. Redko, R. Flamary, and N. Courty. Co-optimal transport. *Advances in neural information processing systems*, 33:17559–17570, 2020.

[43] R. Vidal and P. Favaro. Low rank subspace clustering (lrsc). *Pattern Recognition Letters*, 43:47–61, 2014.

[44] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[45] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[46] H. Xu, D. Luo, and L. Carin. Scalable gromov-wasserstein learning for graph partitioning and matching. *Advances in neural information processing systems*, 32, 2019.

[47] Y. Xu, Y.-L. Li, J. Li, and C. Lu. Constructing balance from imbalance for long-tailed image recognition. In *European Conference on Computer Vision*, pages 38–56. Springer, 2022.

[48] C. You, C.-G. Li, D. P. Robinson, and R. Vidal. Oracle based active set algorithm for scalable elastic net subspace clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3928–3937, 2016.

[49] P. Zhou, Y. Hou, and J. Feng. Deep adversarial subspace clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1596–1604, 2018.

[50] S. Zhu, D. Wang, and T. Li. Data clustering with size constraints. *Knowledge-Based Systems*, 23(8):883–889, 2010.

[51] X. Zhu, D. Anguelov, and D. Ramanan. Capturing long-tail distributions of object subcategories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 915–922, 2014.