The Power Mean Laplacian for Multilayer Graph Clustering

Pedro Mercado[1] **Antoine Gautier**[1] **Francesco Tudisco**[2] **Matthias Hein**[1]
[1]Department of Mathematics and Computer Science, Saarland University, Germany
[2]Department of Mathematics and Statistics, University of Strathclyde, G11XH Glasgow, UK

Abstract

Multilayer graphs encode different kind of interactions between the same set of entities. When one wants to cluster such a multilayer graph, the natural question arises how one should merge the information from different layers. We introduce in this paper a one-parameter family of matrix power means for merging the Laplacians from different layers and analyze it in expectation in the stochastic block model. We show that this family allows to recover ground truth clusters under different settings and verify this in real world data. While computing the matrix power mean can be very expensive for large graphs, we introduce a numerical scheme to efficiently compute its eigenvectors for the case of large sparse graphs.

1 Introduction

Multilayer graphs have received an increasing amount of attention due to their capability to encode different kinds of interactions between the same set of entities [6, 27]. This kind of graphs arise naturally in diverse applications such as transportation networks [16], financial-asset markets [4], temporal dynamics [53, 54], semantic world clustering [48], multi-video face analysis [7], mobile phone networks [26], social balance [8], citation analysis [52], and many others. The extension of clustering techniques to multilayer graphs is a challenging task and several approaches have been proposed so far. See [25, 51, 57, 63] for an overview. For instance, [13, 14, 52, 62] rely on matrix factorizations, whereas [11, 39, 41, 46, 47] take a Bayesian inference approach, and [28, 29] enforce consistency among layers in the resulting clustering assignment. In [37, 40, 55] Newman's modularity [38] is extended to multilayer graphs. Recently [12, 50] proposed to compress a multilayer graph by combining sets of similar layers (called 'strata') to later identify the corresponding communities. Of particular interest to our work is the popular approach [1, 9, 23, 53, 64] that first blends the information of a multilayer graph by finding a suitable weighted arithmetic mean of the layers and then apply standard clustering methods to the resulting mono-layer graph.

In this paper we focus on extensions of spectral clustering to multilayer graphs. Spectral clustering is a well established method for one-layer graphs which, based on the first eigenvectors of the graph Laplacian, embeds nodes of the graphs in $\mathbb{R}^k$ and then uses k-means to find the partition. We propose to blend the information of a multilayer graph by taking certain matrix power means of Laplacians of the layers.

The power mean of scalars is a general family of means that includes as special cases, the arithmetic, geometric and harmonic means. The arithmetic mean of Laplacians has been used before in the case of signed networks [30] and thus our family of matrix power means, see Section 2.2, is a natural extension of this approach. One of our main contributions is to show that the arithmetic mean is actually suboptimal to merge information from different layers.

We analyze the family of matrix power means in the Stochastic Block Model (SBM) for multilayer graphs in two settings, see Section 3. In the first one all the layers are informative, whereas in the second setting none of the individual layers contains the full information but only if one considers them all together. We show that as the parameter of the family of Laplacian means tends to $-\infty$, in expectation one can recover perfectly the clusters in both situations. We provide extensive experiments which show that this behavior is stable when one samples sparse graphs from the SBM. Moreover, in Section 5, we provide additional experiments on real world graphs which confirm our finding in the SBM.

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84. Copyright 2018 by the author(s).

A main challenge for our approach is that the matrix power mean of sparse matrices is in general dense and thus does not scale to large sparse networks in a straightforward fashion. Thus a further contribution of this paper in Section 4 is to show that the first few eigenvectors of the matrix power mean can be computed efficiently. Our algorithm combines the power method with a Krylov subspace approximation technique and allows to compute the extremal eigenvalues and eigenvectors of the power mean of matrices without ever computing the matrix itself.

2 Spectral clustering of multilayer graphs using matrix power means of Laplacians

Let $V = \{v_1, \dots, v_n\}$ be a set of nodes and let T the number layers, represented by adjacency matrices $\mathbb{W} = \{W^{(1)}, \dots, W^{(T)}\}$. For each non-negative weight matrix $W^{(t)} \in \mathbb{R}_+^{n\times n}$ we have a graph $G^{(t)} = (V, W^{(t)})$ and a multilayer graph is the set $\mathbb{G} = \{G^{(1)}, \dots, G^{(T)}\}$. In this paper our main focus are assortative graphs. This kind of graphs are the most common in the literature (see f.i. [34]) and are used to model the situation where edges carry *similarity* information of pairs of vertices and thus are indicative for vertices being in the same cluster. For an assortative graph $G = (V, W)$ spectral clustering is typically based on the Laplacian matrix and its normalized version, defined respectively as

$$L = D - W \qquad L_{\text{sym}} = D^{-1/2} L D^{-1/2}$$

where $D_{ii} = \sum_{j=1}^{n} w_{ij}$ is the diagonal matrix of the degrees of G. Both Laplacians are symmetric positive semidefinite and the multiplicity of eigenvalue 0 is equal to the number of connected components in G.

Given a multilayer graph with all assortative layers $G^{(1)}, \dots, G^{(T)}$, our goal is to come up with a clustering of the vertex set V. We point out that in this paper a clustering is a partition of V, that is each vertex is uniquely assigned to one cluster.

2.1 Matrix power mean of Laplacians for multilayer graphs

Let us briefly recall the scalar power mean of a set of non-negative scalars $x_1, \dots, x_T$. This is a general one-parameter family of means defined for $p \in \mathbb{R}$ as $m_p(x_1, \dots, x_T) = (\frac{1}{T}\sum_{i=1}^{T} x_i^p)^{1/p}$. It includes some well-known means as special cases:

$$\begin{aligned}
\lim_{p\to\infty} m_p(x_1, \dots, x_T) &= \max\{x_1, \dots, x_T\} \\
m_1(x_1, \dots, x_T) &= (x_1 + \dots + x_T)/T \\
\lim_{p\to 0} m_p(x_1, \dots, x_T) &= \sqrt[T]{x_1 \cdot \dots \cdot x_T} \\
m_{-1}(x_1, \dots, x_T) &= T\,(\tfrac{1}{x_1} + \dots + \tfrac{1}{x_T})^{-1} \\
\lim_{p\to -\infty} m_p(x_1, \dots, x_T) &= \min\{x_1, \dots, x_T\}
\end{aligned}$$

corresponding to the maximum, arithmetic, geometric, harmonic mean and minimum, respectively.

Since matrices do not commute, the scalar power mean can be extended to positive definite matrices in a number of different ways, all of them coinciding when applied to commuting matrices. In this work we use the following matrix power mean.

Definition 1 ([5])**.** *Let $A_1, \dots, A_T$ be symmetric positive definite matrices, and $p \in \mathbb{R}$. The matrix power mean of $A_1, \dots, A_T$ with exponent p is*

$$M_p(A_1, \dots, A_T) = \left(\tfrac{1}{T}\sum_{i=1}^{T} A_i^p\right)^{1/p} \tag{1}$$

where $A^{1/p}$ is the unique positive definite solution of the matrix Equation $X^p = A$.

The previous definition can be extended to positive semi-definite matrices. For $p > 0$, $M_p(A_1, \dots, A_T)$ exists for positive semi-definite matrices, whereas for $p \leq 0$ it is necessary to add a suitable diagonal shift to $A_1, \dots, A_T$ to enforce them to be positive definite (see [5] for details).

We call the matrix above *matrix power mean* and we recover for $p = 1$ the standard arithmetic mean of the matrices. Note that for $p \to 0$, the power mean (1) converges to the Log-Euclidean matrix mean [3]

$$M_0(A_1, \dots, A_T) = \exp\left(\tfrac{1}{T}\sum_{i=1}^{T} \log A_i\right),$$

which is a popular form of matrix geometric mean used, for instance, in diffusion tensor imaging or quantum information theory (see f.i. [2, 42]).

Based on the Karcher mean, a different one-parameter family of matrix power means has been discussed for instance in [31]. When the parameter goes to zero, the Karcher-based power mean of two matrices A and B converges to the geometric mean $A\#B = A^{1/2}(A^{-1/2}BA^{-1/2})^{1/2}A^{-1/2}$. The mean $A\#B$ has been used for instance in [15, 36] for clustering in signed networks, for metric learning [61] and geometric optimization [49]. However, when more than two matrices are considered, the Karcher-based power mean is defined as the solution of a set of nonlinear matrix equations with no known closed-form solution and thus is not suitable for multilayer graphs.

```
Algorithm 1: Spectral clustering with $L_p$ on multilayer networks
Input: Symmetric matrices $W^{(1)}, \dots, W^{(T)}$,
       number $k$ of clusters to construct.
Output: Clusters $C_1, \dots, C_k$.
Compute eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_k$ corresponding to
  the $k$ smallest eigenvalues of $L_p$.
Set $U = (\mathbf{u}_1, \dots, \mathbf{u}_k)$ and cluster the rows of $U$
  with $k$-means into clusters $C_1, \dots, C_k$.
```

The matrix power mean (1) is symmetric positive definite and is independent of the labeling of the vertices in the sense that the matrix power mean of relabeled matrices is the same as relabeling the matrix power mean of the original matrices. The latter property is a necessary requirement for any clustering method. The following lemma illustrates the relation to the scalar power mean and is frequently used in the proofs.

Lemma 1. *Let* $\mathbf{u}$ *be an eigenvector of* $A_1, \dots, A_T$ *with corresponding eigenvalues* $\lambda_1, \dots, \lambda_T$*. Then* $\mathbf{u}$ *is an eigenvector of* $M_p(A_1, \dots, A_T)$ *with eigenvalue* $m_p(\lambda_1, \dots, \lambda_T)$*.*

2.2 Matrix power means for multilayer spectral clustering

We consider the multilayer graph $\mathbb{G} = (G^{(1)}, \dots, G^{(T)})$ and define the **power mean Laplacian** L_p of $\mathbb{G}$ as

$$L_p = M_p(L_{\mathrm{sym}}^{(1)}, \dots, L_{\mathrm{sym}}^{(T)}) \qquad (2)$$

where $L_{\mathrm{sym}}^{(t)}$ is the normalized Laplacian of the graph $G^{(t)}$. Note that Definition 1 of the matrix power mean $M_p(A_1, \dots, A_T)$ requires $A_1, \dots, A_T$ to be positive definite. As the normalized Laplacian is positive semi-definite, in the following, for $p \leq 0$ we add to $L_{\mathrm{sym}}^{(t)}$ in Equation (2) a small diagonal shift which ensures positive definiteness, that is we consider $L_{\mathrm{sym}}^{(t)} + \varepsilon I$ throughout the paper. For all numerical experiments we set $\epsilon = \log(1 + |p|)$ for $p < 0$ and $\epsilon = 10^{-6}$ for $p = 0$. Abusing notation slightly, we always mean the shifted versions in the following, unless the shift is explicitly stated.

Similar to spectral clustering for a single graph, we propose Alg. 1 for the spectral clustering of multilayer graphs based on the matrix power mean of Laplacians. As in standard spectral clustering, see [34], our Algorithm 1 uses the eigenvectors corresponding to the k smallest eigenvalues of the power mean Laplacian L_p. Thus the relative ordering of the eigenvalues of L_p is of utmost importance. By Lemma 1 we know that if $A_i \mathbf{u} = \lambda(A_i)\mathbf{u}$, for $i = 1, \dots, n$, then the corresponding eigenvalue of the matrix power mean is $m_p\left(\lambda(A_1), \dots, \lambda(A_T)\right)$. Hence, the ordering of eigenvalues strongly depends on the choice of the parameter p. In the next Section we study the effect of the parameter p on the ordering of the eigenvectors of L_p for multilayer graphs following the stochastic block model.

3 Stochastic block model on multilayer graphs

In this Section we present an analysis of the eigenvectors and eigenvalues of the power mean Laplacian under the Stochastic Block Model (SBM) for multilayer graphs. The SBM is a widespread random graph model for single-layer networks having a prescribed clustering structure [44]. Studies of community detection for multilayer networks following the SBM can be found in [19, 21, 24, 58, 59, 60].

In order to grasp how different methods identify communities in multilayer graphs following the SBM we will analyze three different settings. In the first setting all layers follow the same node partition (see f.i. [19]) and we study the robustness of the spectrum of the power mean Laplacian when the first layer is informative and the other layers are noise or even contain contradicting information. In the second setting we consider the particularly interesting situation where multilayer-clustering is superior over each individual clustering. More specifically, we consider the case where we are searching for three clusters but each layer contains only information about one of them and only considering all of the layers together reveals the information about the underlying cluster structure. In a third setting we go beyond the standard SBM and consider the case where we have a graph partition for each layer, but this partition changes from layer to layer according to a generative model (see f.i.[4]). However, for the last setting we only provide an empirical study, whereas for the first two settings we analyze the spectrum also analytically. For brevity, all the proofs are moved to the supplementary material.

In the following we denote by $\mathcal{C}_1, \dots, \mathcal{C}_k$ the ground truth clusters that we aim to recover. All the $\mathcal{C}_i$ are assumed to have the same size $|\mathcal{C}|$. Calligraphic letters are used for the expected matrices in the SBM. In particular, for a layer $G^{(t)}$ we denote by $\mathcal{W}^{(t)}$ its expected adjacency matrix, by $\mathcal{D}^{(t)} = \mathrm{diag}(\mathcal{W}^{(t)}\mathbf{1})$ the exptected degree matrix and by $\mathcal{L}_{\mathrm{sym}}^{(t)} = I - (\mathcal{D}^{(t)})^{-1/2}\mathcal{W}^{(t)}(\mathcal{D}^{(t)})^{-1/2}$ the expected normalized Laplacian.

3.1 Case 1: Robustness to noise where all layers have the same cluster structure

The case where all layers follow a given node partition is a natural extension of the mono-layer SBM to the multilayer setting. This is done by having different edge probabilities for each layer [19], while fixing the same node partition in all layers. We denote by $p_{\text{in}}^{(t)}$ (resp. $p_{\text{out}}^{(t)}$) the probability that there exists an edge in layer $G^{(t)}$ between nodes that belong to the same (resp. different) clusters. Then $\mathcal{W}_{ij}^{(t)} = p_{\text{in}}^{(t)}$ if v_i, v_j belong to the same cluster and $\mathcal{W}_{ij}^{(t)} = p_{\text{out}}^{(t)}$ if v_i, v_j belong to different clusters. Consider the following k vectors:

$$\boldsymbol{\chi}_1 = \mathbf{1}, \qquad \boldsymbol{\chi}_i = (k-1)\mathbf{1}_{\mathcal{C}_i} - \mathbf{1}_{\overline{\mathcal{C}_i}} \ .$$

The use of k-means on the embedding induced by the vectors $\{\boldsymbol{\chi}_i\}_{i=1}^k$ identifies the ground truth communities $\{\mathcal{C}_i\}_{i=1}^k$. It turns out that in expectation $\{\boldsymbol{\chi}_i\}_{i=1}^k$ are eigenvectors of the power mean Laplacian L_p. We look for conditions so that they correspond to the k smallest eigenvalues as this implies that our spectral clustering Algorithm 1 recovers the ground truth.

Before addressing the general case, we discuss the case of two layers. For this case we want to illustrate the effect of the power mean by simply studying the extreme limit cases

$$\mathcal{L}_\infty := \lim_{p\to\infty} \mathcal{L}_p \quad \text{and} \quad \mathcal{L}_{-\infty} := \lim_{p\to-\infty} \mathcal{L}_p \, .$$

where $\mathcal{L}_p = M_p(\mathcal{L}_{\text{sym}}^{(1)}, \mathcal{L}_{\text{sym}}^{(2)})$. The next Lemma shows that $\mathcal{L}_\infty$ and $\mathcal{L}_{-\infty}$ are related to the logical operators AND and OR, respectively, in the sense that in expectation $\mathcal{L}_\infty$ recovers the clusters if and only if $G^{(1)}$ **and** $G^{(2)}$ have both clustering structure, whereas in expectation $\mathcal{L}_{-\infty}$ recovers the clusters if and only if $G^{(1)}$ **or** $G^{(2)}$ has clustering structure.

Lemma 2. *Let $\mathcal{L}_p = M_p(\mathcal{L}_{\text{sym}}^{(1)}, \mathcal{L}_{\text{sym}}^{(2)})$.*

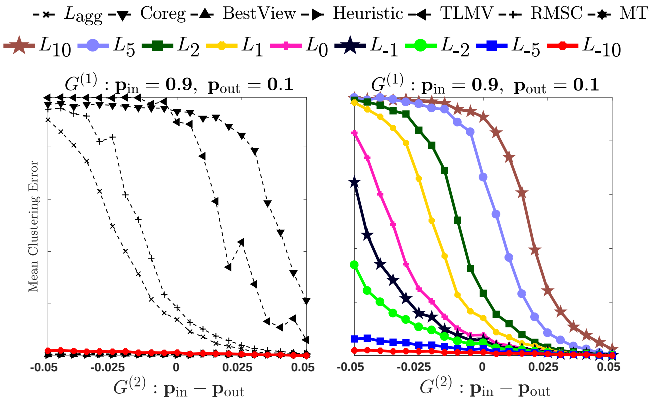
- *$\{\boldsymbol{\chi}_i\}_{i=1}^k$ correspond to the k smallest eigenvalues of $\mathcal{L}_\infty$ if and only if $p_{\text{in}}^{(1)} > p_{\text{out}}^{(1)}$* ***and*** *$p_{\text{in}}^{(2)} > p_{\text{out}}^{(2)}$.*
- *$\{\boldsymbol{\chi}_i\}_{i=1}^k$ correspond to the k smallest eigenvalues of $\mathcal{L}_{-\infty}$ if and only if $p_{\text{in}}^{(1)} > p_{\text{out}}^{(1)}$* ***or*** *$p_{\text{in}}^{(2)} > p_{\text{out}}^{(2)}$.*

The following theorem gives general conditions on the recovery of the ground truth clusters in dependency on p and the size of the shift in $\mathcal{L}_p$, see Section 2.2. Note that, in analogy with Lemma 2, as $p \to -\infty$ the recovery of the ground truth clusters is achieved if at least one of the layers is informative, whereas if $p \to \infty$ all of them have to be informative in order to recover the ground truth.

Theorem 1. *Let $p \in [-\infty, \infty]$, then $\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_k$ correspond to the k-smallest eigenvalues of $\mathcal{L}_p$ if and only if $m_p(\boldsymbol{\mu} + \epsilon\mathbf{1}) < 1 + \epsilon$, where $\boldsymbol{\mu} = (1-\rho_1, \ldots, 1-\rho_T)$, and $\rho_t = (p_{\text{in}}^{(t)} - p_{\text{out}}^{(t)})/(p_{\text{in}}^{(t)} + (k-1)p_{\text{out}}^{(t)})$.*

In particular, for $p \to \pm\infty$, we have

1. *$\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_k$ correspond to the k-smallest eigenvalues of $\mathcal{L}_\infty$ if and only if all layers are informative, i.e. $p_{\text{in}}^{(t)} > p_{\text{out}}^{(t)}$ holds for all $t \in \{1, \ldots, T\}$.*
2. *$\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_k$ correspond to the k-smallest eigenvalues of $\mathcal{L}_{-\infty}$ if and only if there is at least one informative layer, i.e. there exists a $t \in \{1, \ldots, T\}$ such that $p_{\text{in}}^{(t)} > p_{\text{out}}^{(t)}$.*



(a) $L_p = M_p(L_{\text{sym}}^{(1)}, L_{\text{sym}}^{(2)})$ (b) $L_p = M_p(L_{\text{sym}}^{(1)}, L_{\text{sym}}^{(2)})$

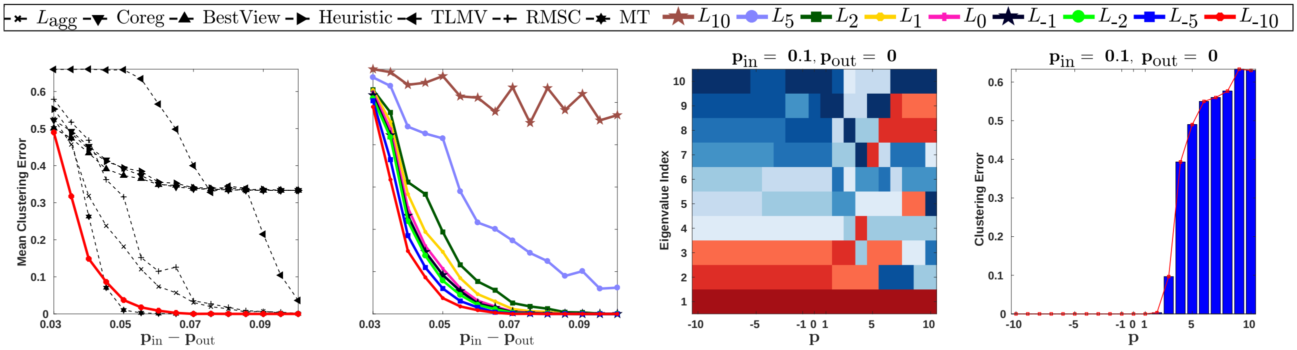
Figure 1: Mean Clustering Error under the SBM with two clusters. First layer $G^{(1)}$ is *assortative* and $L_p = M_p(L_{\text{sym}}^{(1)}, L_{\text{sym}}^{(2)})$. Second layer $G^{(2)}$ transitions from disassortative to assortative. Fig. 1a: Comparison of L_{-10} with state of art. Fig. 1b: Performance of L_p with $p \in \{0, \pm1, \pm2, \pm5, \pm10\}$.

Theorem 1 shows that the informative eigenvectors of $\mathcal{L}_p$ are at the bottom of the spectrum if and only if the scalar power mean of the corresponding eigenvalues is small enough. Since the scalar power mean is monotonically decreasing with respect to p, this explains why the limit case $p \to \infty$ is more restrictive than $p \to -\infty$. The corollary below shows that the coverage of parameter settings in the SBM for which one recovers the ground truth becomes smaller as p grows.

Corollary 1. *Let $q \leq p$. If $\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_k$ correspond to the k-smallest eigenvalues of $\mathcal{L}_p$, then $\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_k$ correspond to the k-smallest eigenvalues of $\mathcal{L}_q$.*

The previous results hold in expectation. The following experiments show that these findings generalize to the case where one samples from the SBM. In Fig. 1 we present experiments on sparse sampled multilayer graphs from the SBM. We consider two clusters of size $|\mathcal{C}| = 100$ and show the mean of clustering error of 50 runs. We evaluate the power mean Laplacian L_p with $p \in \{0, \pm1, \pm2, \pm5, \pm10\}$ and compare with other methods described in Section 5.

In Fig. 1 we fix the first layer $G^{(1)}$ to be strongly assortative and let the second layer $G^{(2)}$ run from a disassortative to an assortative configuration. In Fig.1a we



(a) (b) (c) (d)

Figure 2: SBM experiments with three layers. Each layer is informative with respect to one cluster. 2a: Comparison of L_{-10} with state of art. 2b: Performance of L_p with $p \in \{0, \pm1, \pm2, \pm5, \pm10\}$. 2c: Eigenvalue ordering of power mean Laplacian L_p across different powers. The ordering clearly changes for powers $p \geq 2$, inducing non-informative eigenvectors to the bottom of the spectrum. 2d: Clustering error of the power mean Laplacian L_p. Clustering error increases with $p \geq 2$, as suggested by ordering changes depicted in 2c.

can see that the power mean Laplacian L_{-10} returns the smallest clustering error, together with the multi-tensor method, the best single view and the heuristic approach across all parameter settings. The latter two work well by construction in this setting. However, we will see that they fail for the second setting we consider next. All the other competing methods fail as the second graph $G^{(2)}$ becomes non-informative resp. even violates the assumption to be assortative. In Fig. 1b we can see that the smaller the value of p, the smaller the clustering error of the power mean Laplacian L_p, as stated in Corollary 1.

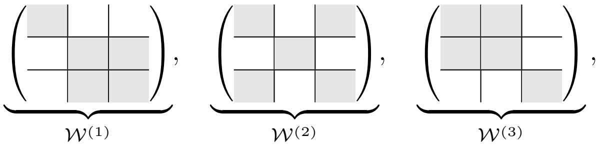
3.2 Case 2: No layer contains full information on the clustering structure

We consider a multilayer SBM setting where each individual layer contains only information about one of the clusters and only considering all the layers together reveals the complete cluster structure. For this particular instance, all power mean Laplacians $\mathcal{L}_p$ allow to recover the ground truth for any non-zero integer p.

For the sake of simplicity, we limit ourselves to the case of three layers and three clusters, showing an assortative behavior in expectation. Let the expected adjacency matrix $\mathcal{W}^{(t)}$ of layer $G^{(t)}$ be defined by

$$\mathcal{W}^{(t)}_{i,j} = \begin{cases} p_{\text{in}}, & v_i, v_j \in \mathcal{C}_t \text{ or } v_i, v_j \in \overline{\mathcal{C}_t} \\ p_{\text{out}}, & \text{else} \end{cases} \tag{3}$$

for $t = 1, 2, 3$. Note that, up to a node relabeling, the three expected adjacency matrices have the form



where each (block) row and column corresponds to a cluster $\mathcal{C}_i$ and gray blocks correspond to nodes whose probability of connections is p_{in}, whereas white blocks correspond to nodes whose probability of connections is p_{out}. Let us assume an assortative behavior on all the layers, that is $p_{\text{in}} > p_{\text{out}}$. In this case spectral clustering applied on a single layer $\mathcal{W}^{(t)}$ would return cluster $\mathcal{C}_t$ and a random partition of the complement, failing to recover the ground truth clustering $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$. This is shown in the following Theorem.

Theorem 2. *If $p_{\text{in}} > p_{\text{out}}$, then for any $t = 1, 2, 3$, there exist scalars $\alpha > 0$ and $\beta > 0$ such that the eigenvectors of $\mathcal{L}^{(t)}_{\text{sym}}$ corresponding to the two smallest eigenvalues are*

$$\boldsymbol{\chi}_1 = \alpha \mathbf{1}_{\mathcal{C}_t} + \mathbf{1}_{\overline{\mathcal{C}_t}} \quad \textit{and} \quad \boldsymbol{\chi}_2 = -\beta \mathbf{1}_{\mathcal{C}_t} + \mathbf{1}_{\overline{\mathcal{C}_t}}$$

whereas any vector orthogonal to both $\boldsymbol{\chi}_1$ and $\boldsymbol{\chi}_2$ is an eigenvector for the third smallest eigenvalue.

On the other hand, it turns out that the power mean Laplacian L_p is able to merge the information of each layer, obtaining the ground truth clustering, for all integer powers different from zero. This is formally stated in the following.

Theorem 3. *Let $p_{\text{in}} > p_{\text{out}}$ and for $\varepsilon > 0$ define*

$$\tilde{\mathcal{L}}^{(t)}_{\text{sym}} = \mathcal{L}^{(t)}_{\text{sym}} + \varepsilon I, \quad t = 1, 2, 3.$$

Then the eigenvectors of $\mathcal{L}_p = M_p(\tilde{\mathcal{L}}^{(1)}_{\text{sym}}, \tilde{\mathcal{L}}^{(2)}_{\text{sym}}, \tilde{\mathcal{L}}^{(3)}_{\text{sym}})$ corresponding to its three smallest eigenvalues are

$$\boldsymbol{\chi}_1 = \mathbf{1}, \quad \boldsymbol{\chi}_2 = \mathbf{1}_{\mathcal{C}_2} - \mathbf{1}_{\mathcal{C}_1}, \quad \textit{and} \quad \boldsymbol{\chi}_3 = \mathbf{1}_{\mathcal{C}_3} - \mathbf{1}_{\mathcal{C}_1}$$

for any nonzero integer p.

The proof of Theorem 3 is more delicate than the one of Theorem 1, as it involves the addition of powers of matrices that do not have the same eigenvectors.

	$\tilde{\boldsymbol{p}}$					
	0.5	0.6	0.7	0.8	0.9	1.0
L_{agg}	0.3	1.3	3.0	8.0	22.3	100.0
Coreg	0.3	0.0	0.3	0.0	0.0	64.7
BestView	9.7	1.0	0.3	0.0	0.7	77.3
Heuristic	0.0	0.0	0.0	0.0	0.3	59.3
TLMV	0.7	0.7	4.0	6.0	24.7	100.0
RMSC	1.0	1.7	4.0	7.0	19.7	100.0
MT	1.3	0.3	0.7	3.0	17.0	100.0
L_{10}	0.0	0.0	0.0	0.0	1.0	100.0
L_{5}	0.0	0.0	0.0	0.0	5.0	100.0
L_{2}	0.0	0.0	0.3	2.3	18.3	100.0
L_{1}	1.0	1.0	3.0	7.0	30.3	100.0
L_{0}	4.3	4.3	9.7	15.3	38.3	100.0
L_{-1}	6.7	7.7	15.7	16.3	42.3	100.0
L_{-2}	8.0	13.0	20.3	20.7	42.7	100.0
L_{-5}	22.3	23.0	36.3	37.7	50.0	100.0
L_{-10}	**69.0**	**76.3**	**68.0**	**67.3**	**59.7**	100.0

	$\boldsymbol{\mu}$					
	0.0	0.1	0.2	0.3	0.4	0.5
L_{agg}	24.7	21.7	21.3	21.7	24.3	21.3
Coreg	16.7	16.7	13.3	11.7	6.0	1.0
BestView	16.7	17.0	17.0	17.7	11.7	9.0
Heuristic	16.7	16.3	15.0	9.0	2.0	0.7
TLMV	25.7	24.3	21.7	23.3	21.0	20.0
RMSC	26.3	22.0	23.0	21.7	20.3	20.0
MT	19.7	19.7	21.0	20.7	20.7	20.7
L_{10}	16.7	17.3	17.0	16.7	16.7	16.7
L_{5}	17.0	18.0	17.3	17.7	18.0	17.0
L_{2}	23.0	21.3	19.3	19.0	20.3	18.0
L_{1}	26.3	25.3	24.0	23.0	22.3	21.3
L_{0}	33.3	30.3	28.7	28.0	28.0	23.7
L_{-1}	36.3	33.0	33.3	32.0	29.0	25.0
L_{-2}	37.3	36.3	36.7	34.0	31.3	29.0
L_{-5}	48.0	45.0	49.0	44.3	43.0	40.0
L_{-10}	**71.7**	**72.3**	**72.7**	**74.7**	**76.3**	**72.7**

Table 1: Percentage of cases where the minimum clustering error is achieved by different methods. Left: Columns correspond to a fixed value of $\tilde{p}$ and we aggregate over $\mu \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. Left: Columns correspond to a fixed value of μ and we aggregate over $\tilde{p} \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$.

Note that Theorem 3 does not distinguish the behavior for distinct values of p. In expectation all nonzero integer values of p work the same. This is different to Theorem 1, where the choice of p had a relevant influence on the eigenvector embedding even in expectation. However, we see in the experiments on graphs sampled from the SBM (Figure 2) that the choice of p has indeed a significant influence on the performance even though they are the same in expectation. This suggests that the smaller p, the smaller the variance in the difference to the expected behavior in the SBM. We leave this as an open problem if such a dependency can be shown analytically.

In Figs. 2a and 2b we present the mean clustering error out of ten runs. In Fig. 2a one can see that BestView and Heuristic, which rely on clusterings determined by single views, return high clustering errors which correspond to the identification of only a single cluster. The result of Theorem 3 explains this failure. The reason for the increasing clustering error with p can be seen in Fig. 2c where we analyze how the ordering of eigenvectors changes for different values of p. We can see that for negative powers, the informative eigenvectors belong to the bottom three eigenvalues (denoted in red). For the cases where $p \geq 2$ the ordering changes, pushing non-informative eigenvectors to the bottom of the spectrum and thus resulting into a high clustering error, as seen in Fig. 2d. However, we conclude that also for this second case a strongly negative power mean Laplacian as L_{-10} works best.

3.3 Case 3: Non-consistent partitions between layers

We now consider the case where all the layers follow the same node partition (as in Section 3.1), but the partitions may fluctuate from layer to layer with a certain probability. We use the multilayer network model introduced in [4]. This generative model considers a graph partition for each layer, allowing the partitions to change from layer to layer according to an interlayer dependency tensor. For the sake of clarity we consider a one-parameter interlayer dependency tensor with parameter $\tilde{p} \in [0, 1]$ (i.e. a uniform multiplex network according to the notation used in Section 3.B in [4]), where for $\tilde{p} = 0$ the partitions between layers are independent, and for $\tilde{p} = 1$ the partitions between layers are identical. Once the partitions are obtained, edges are generated following a multilayer degree-corrected SBM (DCSBM in Section 4 of [4]), according to a one-parameter affinity matrix with parameter $\mu \in [0, 1]$, where for $\mu = 0$ all edges are within communities whereas for $\mu = 1$ edges are assigned ignoring the community structure.

We choose $\tilde{p} \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ and $\mu \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ and consider all possible combinations of $(\tilde{p}, \mu)$. For each pair we count how many times, out of 50 runs, each method achieves the smallest clustering error. The remaining parameters of the DCSBM are set as follows: exponent $\gamma = -3$, minimum degree and maximum degree $k_{min} = k_{max} = 10$, $|V| = 100$ nodes, $T = 10$ layers and $K = 2$ communities. As partitions between layers are not necessarily the same, we take the most frequent node assignment among all 10 layers as ground truth clustering.

In Table 1, left side, we show the result for fixed values of $\tilde{p}$ and average over all values of μ. On the right table we show the corresponding results for fixed values of μ and average over all values of $\tilde{p}$. On the left table we can see that for $\tilde{p} = 1$, where the partition is the same in all layers, all methods recover the clustering, while, as one would expect, the performance decreases with smaller values of $\tilde{p}$. Further, we note that the per-

Algorithm 2: PM applied to M_p.

```
Input: $\mathbf{x}_0$, $p < 0$
Output: Eigenpair $(\lambda, \mathbf{x})$ of $M_p$
repeat
    $\mathbf{u}_k^{(1)} \leftarrow (A_1)^p \mathbf{x}_k$
    $\vdots$
    $\mathbf{u}_k^{(T)} \leftarrow (A_T)^p \mathbf{x}_k$
    $\mathbf{y}_{k+1} \leftarrow \frac{1}{T} \sum_{i=1}^{T} \mathbf{u}_k^{(i)}$
    $\mathbf{x}_{k+1} \leftarrow \mathbf{y}_{k+1} / \|\mathbf{y}_{k+1}\|_2$
until tolerance reached
$\lambda \leftarrow (\mathbf{x}_{k+1}^T \mathbf{x}_k)^{1/p}$,  $\mathbf{x} \leftarrow \mathbf{x}_{k+1}$
```

Algorithm 3: PKSM for the computation of $A^p\mathbf{y}$

```
Input: $\mathbf{u}_0 = \mathbf{y}$, $V_0 = [\,\cdot\,]$, $p < 0$
Output: $\mathbf{x} = A^p\mathbf{y}$
$\mathbf{v}_0 \leftarrow \mathbf{y} / \|\mathbf{y}\|_2$
for $s = 0, 1, 2, \ldots, n$ do
    $\tilde{V}_{s+1} \leftarrow [V_s, \mathbf{v}_s]$
    $V_{s+1} \leftarrow$ Orthogonalize columns of $\tilde{V}_{s+1}$
    $H_{s+1} \leftarrow V_{s+1}^T A V_{s+1}$
    $\mathbf{x}_{s+1} \leftarrow V_{s+1} (H_{s+1})^p \mathbf{e}_1 \|\mathbf{y}\|_2$
    if tolerance reached then break
    $\mathbf{v}_{s+1} \leftarrow A\mathbf{v}_s$
end
$\mathbf{x} \leftarrow \mathbf{x}_{s+1}$
```

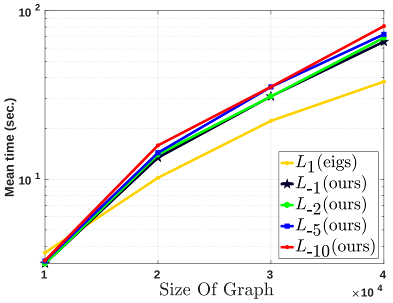


Figure 3: Mean execution time of 10 runs for the power mean Laplacian L_p. L_{-1}(ours), L_{-2}(ours), L_{-5}(ours), L_{-10}(ours) stands for the power mean Laplacian together with our proposed Power Method (Alg. 2) based on the Polynomial Krylov Approximation Method (Alg. 3). L_1(eigs) stands for the arithmetic mean Laplacian together with Matlab's `eigs` function. Experiments are performed using one thread. We generate multilayer graphs with two layers, each with two clusters of same size with parameters $p_{\text{in}} = 0.05$ and $p_{\text{in}} = 0.025$ and graphs of size $|V| \in \{10000, 20000, 30000, 40000\}$.

formance of the power mean Laplacian improves as $\tilde{p}$ decreases and L_{-10} again achieves the best result. On the right table we see that performance is degrading with larger values of μ. This is expected as for larger values of μ the edges inside the clusters are less concentrated. Again the performance of the power mean Laplacian improves as p decreases and L_{-10} performs best.

4 Computing the smallest eigenvalues and eigenvectors of $M_p(A_1, \ldots, A_T)$

We present an efficient method for the computation of the smallest eigenvalues of $M_p(A_1, \ldots, A_T)$ which does not require the computation of the matrix $M_p(A_1, \ldots, A_T)$. This is particularly important when dealing with large-scale problems as $M_p(A_1, \ldots, A_T)$ is typically dense even though each A_i is a sparse matrix. We restrict our attention to the case $p < 0$ which is the most interesting one in practice. The positive case $p > 0$ as well as the limit case $p \to 0$ deserve a different analysis and are not considered here.

Let $A_1, \ldots, A_T$ be positive definite matrices. If $\lambda_1 \le \cdots \le \lambda_n$ are the eigenvalues of $M_p(A_1, \ldots, A_T)$ corresponding to the eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$, then $\mu_i = (\lambda_i)^p$, $i = 1, \ldots, n$, are the eigenvalues of $M_p(A_1, \ldots, A_T)^p$ corresponding to the eigenvectors $\mathbf{u}_i$. However, the function $f(x) = x^p$ is order reversing for $p < 0$. Thus, the relative ordering of the μ_i's changes into $\mu_1 \ge \cdots \ge \mu_n$. Thus, the smallest eigenvalues and eigenvectors of $M_p(A_1, \ldots, A_T)$ can be computed by addressing the largest ones of $M_p(A_1, \ldots, A_T)^p$. To this end we propose a power method type outer-scheme, combined with a Krylov subspace approximation inner-method. The pseudo code is presented in Algs. 2 and 3. Each step of the outer iteration in Alg. 2 requires to compute the pth power of T matrices times a vector. Computing $A^p \times$ *vector*, reduces to the problem of computing the product of a matrix function times a vector. Krylov methods are among the most efficient and most studied strategies to address such a computational issue. As A^p is a polynomial in A, we apply a Polynomial Krylov Subspace Method (PKSM), whose pseudo code is presented in Alg. 3 and which we briefly describe in the following. For further details we refer to [22] and the references therein. For the sake of generality, below we describe the method for a general positive definite matrix A.

The general idea of PKSM s-th iteration is to project A onto the subspace $\mathbb{K}^s(A, \mathbf{y}) = \text{span}\{\mathbf{y}, A\mathbf{y}, \ldots, A^{s-1}\mathbf{y}\}$ and solve the problem there. The projection onto $\mathbb{K}^s(A, \mathbf{y})$ is realized by means of the Lanczos process, producing a sequence of matrices V_s with orthogonal columns, where the first column of V_s is $\mathbf{y}/\|\mathbf{y}\|_2$ and $\text{range}(V_s) = \mathbb{K}^s(A, \mathbf{y})$. Moreover at each step we have $AV_s = V_s H_s + \mathbf{v}_{s+1} \mathbf{e}_s^T$ where H_s is $s \times s$ symmetric tridiagonal, and $\mathbf{e}_i$ is the i-th canonical vector. The matrix vector product $\mathbf{x} = A^p \mathbf{y}$ is then approximated by $\mathbf{x}_s = V_s (H_s)^p \mathbf{e}_1 \|\mathbf{y}\| \approx A^p \mathbf{y}$.

Clearly, if operations are done with infinite precision,

	3Sources	**BBC**	**BBCS**	**Wiki**	**UCI**	**Citeseer**	**Cora**	**WebKB**
# vertices	169	685	544	693	2000	3312	2708	187
# layers	3	4	2	2	6	2	2	2
# classes	6	5	5	10	10	6	7	5
L_{agg}	0.194	0.156	0.152	0.371	0.162	0.373	0.452	**0.277**
Coreg	0.215	0.196	0.164	0.784	0.248	0.395	0.659	0.444
Heuristic	**0.192**	0.218	0.198	0.697	0.280	0.474	0.515	0.400
TLMV	0.284	0.259	0.317	0.412	0.154	0.363	0.533	0.430
RMSC	0.254	0.255	0.194	0.407	0.173	0.422	0.507	0.279
MT	0.249	**0.133**	0.158	0.544	0.103	0.371	0.436	0.298
L_1	0.194	0.154	0.148	0.373	0.163	0.285	**0.367**	0.440
$\boldsymbol{L_{-10}}$ **(ours)**	0.200	0.159	**0.144**	**0.368**	**0.095**	**0.283**	0.374	0.439

Table 2: Average Clustering Error

the exact $\mathbf{x}$ is obtained after n steps. However, in practice, the error $\|\mathbf{x}_s - \mathbf{x}\|$ decreases very fast with s and often very few steps are enough to reach a desirable tolerance. Two relevant observations are in order: first, the matrix $H_s = V_s^T A V_s$ can be computed iteratively alongside the Lanczos method, thus it does not require any additional matrix multiplication; second, the p power of the matrix H_s can be computed directly without any notable increment in the algorithm cost, since H_s is tridiagonal of size $s \times s$.

Several eigenvectors can be simultaneously computed with Algs. 2 and 3 by orthonormalizing the current eigenvector approximation at every step of the power method (Alg. 2) (see f.i. algorithm 5.1 Subspace iteration in [45]). Moreover, the outer iteration in Alg. 2 can be easily run in parallel as the vectors $\mathbf{u}_k^{(i)}$, $i = 1, \dots, T$ can be built independently of each other.

A numerical evaluation of Algs. 2 and 3 is presented in Fig. 3. We consider graphs of sizes $|V| \in \{1\times10^4, 2\times10^4, 3\times10^4, 4\times10^4\}$. Further, for each multilayer graph we generate two assortative graphs with parameters $p_{\text{in}} = 0.05$ and $p_{\text{in}} = 0.025$, following the SBM. Moreover, we consider the power mean Laplacian $L_p = M_p(L_{\text{sym}}^{(1)}, L_{\text{sym}}^{(2)})$ with parameter $p \in \{-1, -2, -5, -10\}$. As a baseline we take the arithmetic mean Laplacian $L_1 = M_1(L_{\text{sym}}^{(1)}, L_{\text{sym}}^{(2)})$ and use Matlab's `eigs` function. For all cases, we compute the two eigenvectors corresponding to the smallest eigenvalues. We present the mean execution time of 10 runs. Experiments are performed using one thread.

5 Experiments

For the sake of comparison we consider the following baseline approaches of spectral clustering applied to: the average adjacency matrix (**$\mathbf{L_{agg}}$**), the arithmetic mean Laplacian (**$\mathbf{L_1}$**), the layer with the largest spectral gap (**Heuristic**), and to the layer with the smallest clustering error (**BestView**). Further, we consider the following methods: Pairwise Co-Regularized Spectral Clustering [29], with parameter $\lambda = 0.01$ (**Coreg**), which proposes a spectral embedding that generates a clustering consistent among all graph layers, Robust Multi-View Spectral Clustering [56], with parameter $\lambda = 0.005$ (**RMSC**), which obtains a robust consensus representation by fusing noiseless information present among layers and samples, spectral clustering applied to a suitable convex combination of normalized adjacency matrices [64] (**TLMV**) that corresponds to a multilayer extension of normalized cuts, and a tensor factorization method [11] (**MT**), which considers a multi-layer mixed membership stochastic block model.

We take popular datasets used for evaluation in multilayer graph clustering: *3-sources* consists of news articles that were covered by news sources [32]; *BBC* [17] and *BBC Sports* [18] news articles, a dataset of Wikipedia articles with ten different classes [43], the hand written *UCI* digits dataset with six different features and citations datasets *CiteSeer* [33], *Cora* [35] and *WebKB* [10], (from WebKB we only take the subset Texas). For each layer we build the corresponding adjacency matrix from the k-nearest neighbour graph based on the Pearson linear correlation between nodes, i.e. the higher the correlation the nearer the nodes are. For each dataset we test all clustering methods over all choices of $k \in \{20, 40, 60, 80, 100\}$, and present the average clustering error in Table 2. Citation datasets CiteSeer, Cora and WebKB have two layers: one is a fixed citation network, whereas the second one is the k-nearest neighbour graph built on documents features. We can see that in four out of eight datasets the power mean Laplacian L_{-10} gets the smallest clustering error. In particular, the largest difference in clustering error is present in the UCI dataset, where the second best is MT. Further, L_1 presents the smallest clustering error in Cora, being L_{-10} close to it. The smallest clustering error in WebKB is achieved by the baseline method L_{agg}. This dataset is particularly challenging, due to conflictive layers, as noted in [20].

References

[1] A. Argyriou, M. Herbster, and M. Pontil. Combining graph Laplacians for semi–supervised learning. In *NIPS*. 2006.

[2] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic resonance in medicine*, 56:411–421, 2006.

[3] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM J. Matrix Anal. Appl.*, 29:328–347, 2007.

[4] M. Bazzi, M. A. Porter, S. Williams, M. McDonald, D. J. Fenn, and S. D. Howison. Community detection in temporal multilayer networks, with an application to correlation networks. *Multiscale Model. Simul.*, 14(1):1–41, 2016.

[5] K. V. Bhagwat and R. Subramanian. Inequalities between means of positive operators. *Mathematical Proceedings of the Cambridge Philosophical Society*, 83(3):393401, 1978.

[6] S. Boccaletti, G. Bianconi, R. Criado, C. del Genio, J. Gmez-Gardees, M. Romance, I. Sendia-Nadal, Z. Wang, and M. Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1 – 122, 2014. The structure and dynamics of multilayer networks.

[7] X. Cao, C. Zhang, C. Zhou, H. Fu, and H. Foroosh. Constrained multi-view video face clustering. *IEEE Transactions on Image Processing*, 24(11):4381–4393, Nov 2015.

[8] D. Cartwright and F. Harary. Structural balance: a generalization of Heider's theory. *Psychological Review*, 63(5):277–293, 1956.

[9] P. Y. Chen and A. O. Hero. Multilayer spectral graph clustering via convex layer aggregation: Theory and algorithms. *IEEE Transactions on Signal and Information Processing over Networks*, 3(3):553–567, Sept 2017.

[10] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the world wide web. AAAI, 2011.

[11] C. De Bacco, E. A. Power, D. B. Larremore, and C. Moore. Community detection, link prediction, and layer interdependence in multilayer networks. *Phys. Rev. E*, 95:042317, Apr 2017.

[12] M. De Domenico, V. Nicosia, A. Arenas, and V. Latora. Structural reducibility of multilayer networks. *Nature Communications*, 6:6864, 2015.

[13] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov. Clustering with multi-layer graphs: A spectral perspective. *IEEE Transactions on Signal Processing*, 60(11):5820–5831, Nov 2012.

[14] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov. Clustering on multi-layer graphs via subspace analysis on grassmann manifolds. *IEEE Transactions on Signal Processing*, 62(4):905–918, Feb 2014.

[15] M. Fasi and B. Iannazzo. Computing the weighted geometric mean of two large-scale matrices and its inverse times a vector. *MIMS EPrint: 2016.29.*

[16] R. Gallotti and M. Barthelemy. The multilayer temporal network of public transport in great britain. *Scientific Data*, 2, 2015.

[17] D. Greene and P. Cunningham. Producing accurate interpretable clusters from high-dimensional data. In A. M. Jorge, L. Torgo, P. Brazdil, R. Camacho, and J. Gama, editors, *Knowledge Discovery in Databases: PKDD 2005*, pages 486–494, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[18] D. Greene and P. Cunningham. A matrix factorization approach for integrating multiple data views. *Machine Learning and Knowledge Discovery in Databases*, pages 423–438, 2009.

[19] Q. Han, K. S. Xu, and E. M. Airoldi. Consistent estimation of dynamic and multi-layer block models. In *ICML*, 2015.

[20] X. He, L. Li, D. Roqueiro, and K. Borgwardt. Multi-view spectral clustering onconflictingviews. In *ECML PKDD*, pages 826–842, 2017.

[21] S. Heimlicher, M. Lelarge, and L. Massoulié. Community detection in the labelled stochastic block model. *arXiv:1209.2910*, 2012.

[22] N. J. Higham. *Functions of Matrices: Theory and Computation.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.

[23] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen. Affinity aggregation for spectral clustering. In *CVPR*, 2012.

[24] V. Jog and P.-L. Loh. Information-theoretic bounds for exact recovery in weighted stochastic block models using the Renyi divergence. *arXiv:1509.06418*, 2015.

[25] J. Kim and J.-G. Lee. Community detection in multi-layer graphs: A survey. *SIGMOD Rec.*, 2015.

[26] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, and J. Laurila. Towards rich mobile phone datasets: Lausanne data collection campaign. *Proc. ICPS, Berlin*, 2010.

[27] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.

[28] A. Kumar and H. D. III. A co-training approach for multi-view spectral clustering. In *ICML*, 2011.

[29] A. Kumar, P. Rai, and H. Daume. Co-regularized multi-view spectral clustering. In *NIPS*, 2011.

[30] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. Luca, and S. Albayrak. Spectral analysis of signed graphs for clustering, prediction and visualization. In *ICDM*, pages 559–570, 2010.

[31] Y. Lim and M. Pálfia. Matrix power means and the Karcher mean. *Journal of Functional Analysis*, 262:1498–1514, 2012.

[32] J. Liu, C. Wang, J. Gao, and J. Han. Multi-view clustering via joint nonnegative matrix factorization. In *SDM*, 2013.

[33] Q. Lu and L. Getoor. Link-based classification. In *ICML*, 2003.

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

[35] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

[36] P. Mercado, F. Tudisco, and M. Hein. Clustering signed networks with the geometric mean of Laplacians. In *NIPS*. 2016.

[37] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010.

[38] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[39] S. Paul and Y. Chen. Consistent community detection in multi-relational data through restricted multi-layer stochastic blockmodel. *Electron. J. Statist.*, 10(2):3807–3870, 2016.

[40] S. Paul and Y. Chen. Null models and modularity based community detection in multi-layer networks. *arXiv:1608.00623*, 2016.

[41] T. P. Peixoto. Inferring the mesoscale structure of layered, edge-valued, and time-varying networks. *Phys. Rev. E*, 92:042807, 2015.

[42] D. Petz. *Quantum Information Theory and Quantum Statistics*. Springer Berlin Heidelberg, 2007.

[43] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos. A new approach to cross-modal multimedia retrieval. In *ACM Multimedia*, 2010.

[44] K. Rohe, S. Chatterjee, B. Yu, et al. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 2011.

[45] Y. Saad. *Numerical Methods for Large Eigenvalue Problems*. SIAM, 2011.

[46] A. Schein, J. Paisley, D. M. Blei, and H. Wallach. Bayesian Poisson Tensor factorization for inferring multilateral relations from sparse dyadic event counts. In *KDD*, 2015.

[47] A. Schein, M. Zhou, D. Blei, and H. Wallach. Bayesian Poisson Tucker decomposition for learning the structure of international relations. In *ICML*, 2016.

[48] J. Sedoc, J. Gallier, D. Foster, and L. Ungar. Semantic word clusters using signed spectral clustering. In *ACL*, 2017.

[49] S. Sra and R. Hosseini. Geometric optimization in machine learning. In *Algorithmic Advances in Riemannian Geometry and Applications*, pages 73–91. Springer, 2016.

[50] N. Stanley, S. Shai, D. Taylor, and P. J. Mucha. Clustering network layers with the strata multilayer stochastic block model. *IEEE Transactions on Network Science and Engineering*, 3(2):95–105, 2016.

[51] S. Sun. A survey of multi-view machine learning. *Neural Computing and Applications*, 23(7):2031–2038, 2013.

[52] W. Tang, Z. Lu, and I. S. Dhillon. Clustering with multiple graphs. In *ICDM*, 2009.

[53] D. Taylor, R. S. Caceres, and P. J. Mucha. Super-resolution community detection for layer-aggregated multilayer networks. *Phys. Rev. X*, 7:031056, 2017.

[54] D. Taylor, S. Shai, N. Stanley, and P. J. Mucha. Enhanced detectability of community structure in multilayer networks through layer aggregation. *Phys. Rev. Lett.*, 116:228301, 2016.

[55] J. D. Wilson, J. Palowitch, S. Bhamidi, and A. B. Nobel. Community extraction in multilayer networks with heterogeneous community structure. *Journal of Machine Learning Research*, 18(149):1–49, 2017.

[56] R. Xia, Y. Pan, L. Du, and J. Yin. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *AAAI*, 2014.

[57] C. Xu, D. Tao, and C. Xu. A survey on multi-view learning. *arXiv:1304.5634*, 2013.

[58] J. Xu, L. Massouli, and M. Lelarge. Edge label inference in generalized stochastic block models: from spectral theory to impossibility results. In *COLT*, 2014.

[59] M. Xu, V. Jog, and P.-L. Loh. Optimal rates for community estimation in the weighted stochastic block model. *arXiv:1706.01175*, 2017.

[60] S.-Y. Yun and A. Proutiere. Optimal cluster recovery in the labeled stochastic block model. In *NIPS*. 2016.

[61] P. Zadeh, R. Hosseini, and S. Sra. Geometric mean metric learning. In *ICML*, 2016.

[62] H. Zhao, Z. Ding, and Y. Fu. Multi-view clustering via deep matrix factorization. In *AAAI*, 2017.

[63] J. Zhao, X. Xie, X. Xu, and S. Sun. Multi-view learning overview: Recent progress and new challenges. *Information Fusion*, 38:43 – 54, 2017.

[64] D. Zhou and C. J. Burges. Spectral clustering and transductive learning with multiple views. In *ICML*, 2007.