Clustering on Multi-Layer Graphs via Subspace Analysis on Grassmann Manifolds

Xiaowen Dong, Pascal Frossard, *Senior Member, IEEE*, Pierre Vandergheynst, and Nikolai Nefedov

***Abstract*—Relationships between entities in datasets are often of multiple nature, like geographical distance, social relationships, or common interests among people in a social network, for example. This information can naturally be modeled by a set of weighted and undirected graphs that form a global multi-layer graph, where the common vertex set represents the entities and the edges on different layers capture the similarities of the entities in term of the different modalities. In this paper, we address the problem of analyzing multi-layer graphs and propose methods for clustering the vertices by efficiently merging the information provided by the multiple modalities. To this end, we propose to combine the characteristics of individual graph layers using tools from subspace analysis on a Grassmann manifold. The resulting combination can then be viewed as a low dimensional representation of the original data which preserves the most important information from diverse relationships between entities. As an illustrative application of our framework, we use our algorithm in clustering methods and test its performance on several synthetic and real world datasets where it is shown to be superior to baseline schemes and competitive to state-of-the-art techniques. Our generic framework further extends to numerous analysis and learning problems that involve different types of information on graphs.**

***Index Terms*—Multi-layer graphs, subspace representation, Grassmann manifold, clustering.**

I. Introduction

GRAPHS are powerful mathematical tools for modeling pairwise relationships among sets of entities; they can be used for various analysis tasks such as classification or clustering. Traditionally, a graph captures a single form of relationships between entities and data are analyzed in light of this one-layer graph. However, numerous emerging applications rely on different forms of information to characterize relationships between entities. Diverse examples include human interactions in a social network or similarities between images or videos in multimedia applications. The multimodal nature of the relationships can naturally be represented by a set of weighted and undirected graphs that share a common set of vertices but with different edge weights depending on the type of information in each graph. This can then be represented by a multi-layer or multi-view graph which gathers all sources of information in a unique representation. Assuming that all the graph layers are informative, they are likely to provide complementary information and thus to offer richer information than any single layer taken in isolation. We thus expect that a proper combination of the information contained in the different layers leads to an improved understanding of the structure of the data and the relationships between entities in the dataset.

In this paper, we consider a M-layer graph G with individual graph layers $G_i = \{V, E_i, \omega_i\}$, $i = 1, \ldots, M$, where V represents the common vertex set and E_i represents the edge set in the i-th individual graph G_i with associated edge weights ω_i. An example of a three-layer graph is shown in Fig. 1(a), where the three graph layers share the same set of 12 vertices but with different edges (we assume unit edge weights for the sake of simplicity). Clearly, different graph layers capture different types of relationships between the vertices, and our objective is to find a method that properly combines the information in these different layers. We first adopt a subspace representation for the information provided by the individual graph layers, which is inspired by the spectral clustering algorithms [1]–[3]. We then propose a novel method for combining the multiple subspace representations into one representative subspace. Specifically, we model each graph layer as a subspace on a Grassmann manifold. The problem of combining multiple graph layers is then transformed into the problem of efficiently merging different subspaces on a Grassmann manifold. To this end, we study the distances between the subspaces and develop a new framework to merge the subspaces where the overall distance between the representative subspace and the individual subspaces is minimized. We further show that our framework is well justified by results from statistical learning theory [4], [5]. The proposed method is a dimensionality reduction algorithm for the original data; it leads to a summarization of the information contained in the multiple graph layers, which reveals the intrinsic relationships between the vertices in the multi-layer graph.

Various learning problems can then be solved using these relationships, such as classification or clustering. Specifically, we focus in this paper on the clustering problem: we want to find a unified clustering of the vertices (as illustrated in Fig. 1(b)) by utilizing the representative subspace, such that it is better than clustering achieved on any of the graph layers G_i independently. To address this problem, we first apply our

Manuscript received September 10, 2013; revised December 09, 2013; accepted December 12, 2013. Date of publication December 18, 2013; date of current version January 20, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ignacio Santamaria. This work has been partly supported by Nokia Research Center (NRC) Lausanne, and the EDGAR project funded by Hasler Foundation, Switzerland.

X. Dong, P. Frossard, and P. Vandergheynst are with Signal Processing Laboratories (LTS4/LTS2), École Polytechnique Fédérale de Lausanne (EPFL), Lausanne 1015, Switzerland (e-mail: xiaowen.dong@epfl.ch; pascal.frossard@epfl.ch; pierre.vandergheynst@epfl.ch).

N. Nefedov is with Signal and Information Processing Laboratory, Eidgenössische Technische Hochschule Zürich (ETH Zürich), Zurich 8092, Switzerland (e-mail: nefedov@isi.ee.ethz.ch).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2013.2295553

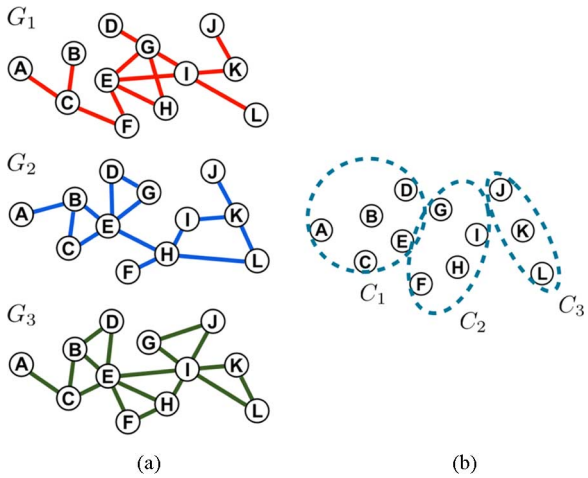


Fig. 1. (a) An illustration for a three-layer graph G, whose three layers $\{G_i\}_{i=1}^3$ share the same set of vertices but with different edges. (b) A potential unified clustering $\{C_k\}_{k=1}^3$ of the vertices based on the information provided by the three layers.

generic framework of subspace analysis on the Grassmann manifold to compute a meaningful summarization (as a representative subspace) of information contained in the individual graph layers. We then implement a spectral clustering algorithm based on the representative subspace. Experiments on synthetic and real world datasets demonstrate the advantages of our approach compared to baseline algorithms, like the summation of individual graphs [6], as well as state-of-the-art techniques, such as co-regularization [7]. Finally, we believe that our framework is beneficial not only to clustering, but also to many other data processing tasks based on multi-layer graphs or multi-view data in general.

This paper is organized as follows. We first review the related work and summarize the contribution of the paper in Section II. In Section III, we describe the subspace representation inspired by spectral clustering, which captures the characteristics of a single graph. In Section IV, we review the main ingredients of Grassmann manifold theory, and propose a new framework for combining information from multiple graph layers. We then propose our novel algorithm for clustering on multi-layer graphs in Section V, and compare its performance with other clustering methods on multiple graphs in Section VI. Finally, we conclude in Section VII.

II. Related Work

In this section we review the related work in the literature. First, we describe briefly graph-based clustering algorithms, with a particular focus on the methods that have subspace interpretations. Second, we summarize the previous works built upon subspace analysis and the Grassmann manifold theory. Finally, we report the recent progresses in the field of analysis of multi-layer graphs or multi-view data.

Clustering on graphs has been studied extensively due to its numerous applications in different domains. The works in [8], [9] have given comprehensive overviews of the advancements in this field over the last few decades. The algorithms based on spectral techniques on graphs are of particular interest, typical examples being spectral clustering [1]–[3], [10] and modularity maximization via spectral method [11], [12]. Specifically, these approaches propose to embed the vertices of the original graph into a low dimensional space, usually called the spectral embedding, which consists of the top eigenvectors of a special matrix (graph Laplacian matrix for spectral clustering and modularity matrix for modularity maximization). Due to the special properties of these matrices, clustering in such low dimensional spaces usually becomes trivial. Therefore, the corresponding clustering approaches can be interpreted as transforming the information on the original graph into a meaningful subspace representation. Another example is the Principal Component Analysis (PCA) interpretation on graphs described in [13]. These works have inspired us to consider the subspace representation in Section III.

In the past few decades, subspace-based methods have been widely used in classification and clustering problems, most notably in image processing and computer vision. In [14], [15], the authors have discovered that human faces can be characterized by low-dimensional subspaces. In [16], the authors have proposed to use the so-called "eigenfaces" for recognition. Inspired by these works, researchers have been particularly interested in data where data points of the same pattern can be represented by a subspace. Due to the growing interests in this field, there is an increasingly large number of works that use tools from the Grassmann manifold theory, which provides a natural tool for subspace analysis. In [17], the authors have given a detailed overview of the basics of the Grassmann manifold theory, and developed new optimization techniques on the Grassmann manifold. In [18], the author has presented statistical analysis on the Grassmann manifold. Both works study the distances on the Grassmann manifold. In [19], [4], the authors have proposed learning frameworks based on distance analysis and positive semidefinite kernels defined on the Grassmann manifold. Other recent representative works include [20]–[24], however, none of the above works considers datasets represented by multi-layer graphs.

At the same time, multi-view data have attracted a large amount of interest in the learning research communities. These data form multi-layer graph representations (or multi-view representations), which generally refer to data that can be analyzed from different viewpoints. In this setting, the key challenge is to combine efficiently the information from multiple graphs (or multiple views) for learning purposes. The existing techniques can be roughly grouped into the following categories. First, the most straightforward way is to form a convex combination of the information from the individual graphs. For example, in [25], the authors have developed a method to learn an optimal convex combination of Laplacian kernels from different graphs. In [26], the authors have proposed a Markov mixture model, which corresponds to a convex combination of the normalized adjacency matrices of the individual graphs, for supervised and unsupervised learning. In [27], the authors have presented several averaging techniques for combining information from the individual graphs for clustering. Finally, in [28], the authors have proposed to combine multiple kernels by forming summations of weighted or projected kernels. Second, following the intuitive approaches in the first category of convex combination of information layers, many existing works aim at finding

a unified representation of the multiple graphs (or multiple views), but using more sophisticated methods. For instances, the authors in [6], [29]–[33] have developed several joint matrix factorization approaches to combine different views of data through a unified optimization framework, where the authors in [34] have proposed to find a unified spectral embedding of the original data by integrating information from different views. Similarly, clustering algorithms based on Canonical Correlation Analysis (CCA) first project the data from different views into a unified low dimensional subspace, and then apply simple algorithms like single linkage or k-means to achieve the final clustering [35], [36]. Third, unlike the previous methods that try to find a unified representation before applying learning techniques, another strategy in the literature is to integrate the information from individual graphs (views) directly into the optimization problems for the learning purposes. Examples include the co-EM clustering algorithm proposed in [37], and the clustering approaches proposed in [38], [7] based on the frameworks of co-training [39] and co-regularization [40]. Fourth, particularly in the analysis of multiple graphs, regularization frameworks on graphs have also been applied. In [41], the authors have presented a regularization framework over edge weights of multiple graphs to compute an improved similarity graph of the vertices (entities). In [42], [31], the authors have proposed graph regularization frameworks in both vertex and graph spectral domain to combine individual graph layers. Finally, other representative approaches include introducing additional graph representations in the learning processes [43], [41] and ensemble clustering [44]–[47]. From this categorization, the proposed approach belongs to the second category mentioned above, where we first find a representative subspace for the information provided by the multi-layer graph and then implement the clustering step, or other learning tasks. We believe that this type of approaches is intuitive and easily understandable, yet still flexible and generic enough to be applied to different types of data.

To summarize, the main differences between the related work and the contributions proposed in this paper are the following. First, the research work on Grassmann manifold theory has been mainly focused on subspace analysis. The subspaces usually come directly from the data but are not linked to graph-based learning problems. Our paper makes the explicit link between subspaces and graphs, and presents a fundamental and intuitive way of approaching the learning problems on multi-layer graphs, with help of subspace analysis on the Grassmann manifold. Second, we show the link between the projection distance on the Grassmann manifold [17], [19] and the empirical estimate of the Hilbert-Schmidt Independence Criterion (HSIC) [5]. Therefore, together with the results in [4], we offer a unified view of concepts from three different perspectives, namely, the projection distance on the Grassmann manifold, the Kullback-Leibler (K-L) divergence [48] and the HSIC [5]. This helps to understand better the key concept of distance measure in subspace analysis. Finally, using our novel layer merging framework, we provide a simple yet competitive solution to the problem of clustering on multi-layer graphs. We also discuss the influence of the relationships between the individual graph layers on the performance of the proposed clustering algorithm. We believe that this is helpful towards the design of efficient and adaptive learning algorithms.

III. Subspace Representation for Graphs

In this section, we describe a subspace representation for the information provided by a single graph, which is inspired by the spectral clustering algorithms. Let us consider a weighted and undirected graph[1] $G = (V, E, \omega)$, where $V = \{v_i\}_{i=1}^n$ represents the vertex set and E represents the edge set with associated edge weights ω, respectively. Let W and D denote the adjacency and degree matrix of G. The normalized graph Laplacian matrix L is then defined as: $L = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$. The graph Laplacian is of broad interests in the studies of spectral graph theory [49]. Among several variants, we use the normalized graph Laplacian defined above, since its spectrum (i.e., its eigenvalues) always lie between 0 and 2, a property favorable in comparing different graph layers in the following sections. We consider now the problem of clustering the vertices $V = \{v_i\}_{i=1}^n$ of G into k distinct subsets such that the vertices in the same subset are similar, i.e., they are connected by edges of large weights. This problem can be efficiently solved by the spectral clustering algorithms. Specifically, we focus on the algorithm proposed in [2], which solves the following trace minimization problem:

$$\min_{U \in \mathbb{R}^{n \times k}} \operatorname{tr}(U'LU), \quad \text{s.t.} \quad U'U = I, \tag{1}$$

where n is the number of vertices in the graph, k is the target number of clusters, and $(\cdot)'$ denotes the matrix transpose operator. It can be shown by a version of the Rayleigh-Ritz theorem [3] that the solution U to the problem of (1) contains the first k eigenvectors (which correspond to the k smallest eigenvalues) of L as columns. The clustering of the vertices in G is then achieved by applying the k-means algorithm [50] to the normalized row vectors of the matrix[2] U. This algorithm is summarized in Algorithm 1.

Algorithm 1: Normalized Spectral Clustering [2]

1: **Input:**
W: the $n \times n$ weighted adjacency matrix of graph G k: target number of clusters
2: Compute the degree matrix D and the normalized graph Laplacian matrix $L = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$.
3: Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the first k eigenvectors $u_1, \ldots, u_k$ of L (solution of (1)). Normalize each row of U to get U_{norm}.
4: Let $y_j \in \mathbb{R}^k (j = 1, \ldots, n)$ be the transpose of the j-th row of U_{norm}.
5: Cluster y_j in $\mathbb{R}^k$ into k clusters $C_1, \ldots, C_k$ using the k-means algorithm.
6: **Output:**
$C_1, \ldots, C_k$: the cluster assignment

[1] We use the notation G for a single graph exclusively in this section.

[2] The necessity for row normalization is discussed in [3] and we omit this discussion here. However, the normalization does not change the nature of spectral embedding, hence, it does not affect our derivation later.

Inspired by the spectral clustering theory, one can define a meaningful subspace representation of the vertex connectivity in a graph using its k-dimensional spectral embedding, which is driven by the matrix U built on the first k eigenvectors of the graph Laplacian L. Each row being the coordinates of the corresponding vertex in the low dimensional subspace, this representation contains the information on the connectivity of the vertices in the original graph. Such information can be used for finding clusters of the vertices, as shown above, but it is also useful for other analysis tasks on graphs. By adopting this subspace representation that "summarizes" the graph information, multiple graph layers can naturally be represented by multiple such subspaces (whose geometrical relationships can be quite flexible). The task of multi-layer graph analysis can then be transformed into the problem of effective combination of the multiple subspaces. This is the focus of the next section.

IV. Merging Subspaces via Analysis on the Grassmann Manifold

We now discuss the problem of effectively combining multiple graph layers by merging multiple subspaces. The theory of Grassmann manifold provides a natural framework for such a problem. In this section, we first review the main ingredients of the Grassmann manifold theory, and then move onto our generic framework for merging subspaces.

A. Ingredients of Grassmann Manifold Theory

By definition, a Grassmann manifold $\mathcal{G}(k,n)$ is the set of k-dimensional linear subspaces in $\mathbb{R}^n$, where each unique subspace is mapped to a unique point on the manifold. The advantage of using tools from Grassmann manifold theory is thus two-fold: (i) it provides a natural and intuitive representation for our problem: the subspaces representing the individual graph layers can be considered as individual points on the Grassmann manifold[3]; (ii) the analysis on the Grassmann manifold permits to use efficient tools to study the distances between points on the manifold, namely, distances between different subspaces. Such distances play an important role in the problem of merging the information from multiple graph layers. Mathematically speaking, each point on $\mathcal{G}(k,n)$ can be represented by an orthonormal matrix $Y \in \mathbb{R}^{n\times k}$ whose columns span the corresponding k-dimensional subspace in $\mathbb{R}^n$; it is thus denoted as $\mathrm{span}(Y)$. The distance between two points on the manifold, or between two subspaces $\mathrm{span}(Y_1)$ and $\mathrm{span}(Y_2)$, is then defined based on a set of principal angles $\{\theta_i\}_{i=1}^k$ between these subspaces [51]. Specifically, θ_1 is defined as the smallest possible angle between all pairs of unit vectors z_i and z_j that come from $\mathrm{span}(Y_1)$ and $\mathrm{span}(Y_2)$ respectively. The other principal angles are defined recursively as the smallest possible angle between all pairs of unit vectors that (i) come from $\mathrm{span}(Y_1)$ and $\mathrm{span}(Y_2)$ and (ii) are orthogonal to all the previously selected z_i and z_j. By definition, the principal angles measure how the subspaces are geometrically close to each other, and are the fundamental measures used to define various distances on the Grassmann manifold, such as the Riemannian (geodesic) distance or the projection distance [17], [19]. In this paper, we use the projection distance, which is defined as:

$$d_{\mathrm{proj}}(Y_1,Y_2)=\left(\sum_{i=1}^{k}\sin^2\theta_i\right)^{\frac{1}{2}}, \tag{2}$$

where Y_1 and Y_2 are the orthonormal matrices representing the two subspaces under comparison[4]. The reason for choosing the projection distance is two-fold: (i) the projection distance is defined as the ℓ^2-norm of the vector of sines of the principal angles. Since it uses all the principal angles, it is therefore an unbiased definition. This is favorable as we do not assume prior knowledge on the importance of specific principal angles, and we consider that all of them carry meaningful information; (ii) the projection distance can be interpreted using a one-to-one mapping that preserves distinctness: $\mathrm{span}(Y) \rightarrow YY' \in \mathbb{R}^{n\times n}$. Note that the squared projection distance can be rewritten as:

$$\begin{aligned} d^2_{\mathrm{proj}}(Y_1,Y_2) &= \sum_{i=1}^{k}\sin^2\theta_i = k-\sum_{i=1}^{k}\cos^2\theta_i \\ &= k-\mathrm{tr}(Y_1Y_1'Y_2Y_2') \qquad (3)\\ &= \frac{1}{2}[2k-2\mathrm{tr}(Y_1Y_1'Y_2Y_2')] \\ &= \frac{1}{2}[\mathrm{tr}(Y_1'Y_1)+\mathrm{tr}(Y_2'Y_2)-2\mathrm{tr}(Y_1Y_1'Y_2Y_2')] \\ &= \frac{1}{2}\|Y_1Y_1'-Y_2Y_2'\|_F^2, \qquad (4)\end{aligned}$$

where the third equality comes from the definition of the principal angles and the fifth equality uses the fact that Y_1 and Y_2 are orthonormal matrices. It can be seen from (4) that the projection distance can be related to the Frobenius norm of the difference between the mappings of the two subspaces $\mathrm{span}(Y_1)$ and $\mathrm{span}(Y_2)$ in $\mathbb{R}^{n\times n}$. Because the mapping preserves distinctness, it is natural to take the projection distance as a proper distance measure between subspaces. Moreover, (3) provides an explicit way of computing the projection distance between two subspaces from their matrix representations Y_1 and Y_2. We are going to use it in developing the generic merging framework in the following section.

B. Layer Merging Framework

Equipped with the subspace representation for individual graphs and with a distance measure to compare different subspaces, we are now ready to present our generic framework for merging the information from multiple graph layers. Given a multi-layer graph G with M individual layers $\{G_i\}_{i=1}^M$, we first compute the graph Laplacian matrix L_i for each G_i and then represent each G_i by the spectral embedding matrix $U_i \in \mathbb{R}^{n\times k}$ from the first k eigenvectors of L_i, where n is the number of vertices and k is the target number of clusters. Recall that each of the matrices $\{U_i\}_{i=1}^M$ defines a k-dimensional subspace in $\mathbb{R}^n$, which can be denoted as $\mathrm{span}(U_i)$. The goal is to merge these multiple subspaces in a meaningful and efficient way. To this end, our philosophy is to find a representative

[3]Different graph layers naturally lead to different points on the manifold; However, we do not specifically exclude the case where there exist two graph layers that are exactly the same.

[4]In the special case where Y_1 and Y_2 represent the same subspace, we have $d_{\mathrm{proj}}(Y_1,Y_2)=0$.

subspace $\mathrm{span}(U)$ that is close to all the individual subspaces $\mathrm{span}(U_i)$, and at the same time the representation U preserves the information about vertex connectivity in each graph layer. For notational convenience, in the rest of the paper we simply refer to the representations U and U_i as the corresponding subspaces, unless indicated specifically.

The squared projection distance between subspaces defined in (4) can be naturally generalized for analysis of multiple subspaces. More specifically, we can define the squared projection distance between the target representative subspace U and the M individual subspaces $\{U_i\}_{i=1}^M$ as the sum of squared projection distances between U and each individual subspace given by U_i:

$$\begin{aligned} d^2_{\mathrm{proj}}\left(U, \{U_i\}_{i=1}^M\right) &= \sum_{i=1}^M d^2_{\mathrm{proj}}(U, U_i) \\ &= \sum_{i=1}^M [k - \mathrm{tr}(UU'U_iU_i')] \\ &= kM - \sum_{i=1}^M \mathrm{tr}(UU'U_iU_i'). \end{aligned} \quad (5)$$

The minimization of the distance measure in (5) enforces the representative subspace U to be close to all the individual subspaces $\{U_i\}_{i=1}^M$ in terms of the projection distance on the Grassmann manifold. At the same time, we want U to preserve the information about vertex connectivity in each graph layer. This can be achieved by minimizing the Laplacian quadratic form evaluated on the columns of U, as also indicated by the objective function in (1) for spectral clustering. Therefore, we finally propose to merge multiple subspaces by solving the following optimization problem that integrates (1) and (5):

$$\min_{U\in\mathbb{R}^{n\times k}} \sum_{i=1}^M \mathrm{tr}(U'L_iU) + \alpha[kM - \sum_{i=1}^M tr(UU'U_iU_i')], \quad \text{s.t.} \quad U'U = I, \quad (6)$$

where L_i and U_i are the graph Laplacian and the subspace representation for G_i, respectively. The regularization parameter α balances the trade-off between the two terms in the objective function.

The problem of (6) can be solved in a similar manner as (1). Specifically, by ignoring constant terms and rearranging the trace form in the second term of the objective function, (6) can be rewritten as

$$\min_{U\in\mathbb{R}^{n\times k}} \mathrm{tr}\left[U'\left(\sum_{i=1}^M L_i - \alpha\sum_{i=1}^M U_iU_i'\right)U\right], \quad \text{s.t.} \quad U'U = I. \quad (7)$$

It is interesting to note that this is the same trace minimization problem as in (1), but with a "modified" Laplacian:

$$L_{\mathrm{mod}} = \sum_{i=1}^M L_i - \alpha\sum_{i=1}^M U_iU_i'. \quad (8)$$

Therefore, by the Rayleigh-Ritz theorem, the solution to the problem of (7) is given by the first k eigenvectors of the modified Laplacian L_{mod}, which can be computed using efficient algorithms for eigenvalue problems [52], [53].

In the problem of (6) we try to find a representative subspace U from the multiple subspaces $\{U_i\}_{i=1}^M$. Such a representation not only preserves the structural information contained in the individual graph layers, which is encouraged by the first term of the objective function in (6), but also keeps a minimum distance between itself and the multiple subspaces, which is enforced by the second term. Notice that the minimization of only the first term itself corresponds to simple averaging of the information from different graph layers, which usually leads to suboptimal clustering performance as we shall see in the experimental section. Similarly, imposing only a small projection distance to the individual subspaces $\{U_i\}_{i=1}^M$ does not necessarily guarantee that U is a good solution for merging the subspaces. In fact, for a given k-dimensional subspace, there are infinitely many choices for the matrix representation, and not all of them are considered as meaningful summarizations of the information provided by the multiple graph layers. However, under the additional constraint of minimizing the trace of the quadratic term $U'L_iU$ over all the graphs (which is the first term of the objective function in (6)), the vertex connectivity in the individual graphs tends to be preserved in U. In this case, the smaller the projection distance between U and the individual subspaces, the more representative it is for all graph layers.

Finally, we note that the proposed merging framework can be easily extended to take into account the relative importance of each individual graph layer with respect to the specific learning purpose. For instance, when prior knowledge about the importance of the information in the individual graphs is available, we can adapt the value of the regularization parameter α in (6) to the different layers such that the representative subspace is closer to the most informative subspace representations. Also, we do not incorporate specific prior knowledge about vertex pairs that must, or must not be linked in the design of the merging framework. That could be done by introducing additional graph layers that only consist of such connections, which would certainly be emphasized by the spectral (subspace) representations computed from these graph layers. We can then choose rather large regularization parameters for these layers in the optimization problem to enforce such constraints.

C. Discussion of the Distance Function

Interestingly, the choice of projection distance as a similarity measure between subspaces in the optimization problem of (6) can be well justified from information-theoretic and statistical learning points of view. The first justification is from the work of Hamm *et al.* [4], in which the authors have shown that the Kullback-Leibler (K-L) divergence [48], which is a well-known similarity measure between two probability distributions in information theory, is closely related to the squared projection distance. More specifically, the work in [4] suggests that, under certain conditions, we can consider a linear subspace U_i as the "flattened" limit of a Factor Analyzer distribution p_i [54]:

$$p_i : \mathcal{N}(u_i, C_i), \quad C_i = U_iU_i' + \sigma^2 I_D, \quad (9)$$

where $\mathcal{N}$ stands for the normal distribution, $u_i \in \mathbb{R}^n$ is the mean, $U_i \in \mathbb{R}^{n\times k}$ is a full-rank matrix with $n > k > 0$ (which represents the subspace), σ is the ambient noise level, and I_n is

the identity matrix of dimension n. For two subspaces U_i and U_j, the symmetrized K-L divergence between the two corresponding distributions p_i and p_j can then be rewritten as:

$$d_{\mathrm{KL}}(p_i, p_j) = \frac{1}{2\sigma^2(\sigma^2+1)}(2k - 2\mathrm{tr}(U_i{U_i}'U_j{U_j}')), \quad (10)$$

which is of the same form as the squared projection distance when we ignore the constant factor (see (3)). This shows that, if we take a probabilistic view of the subspace representations $\{U_i\}_{i=1}^M$, then the projection distance between subspaces can be considered consistent with the K-L divergence.

The second justification is from the recently proposed Hilbert-Schmidt Independence Criterion (HSIC) [5], which measures the statistical dependence between two random variables. Given $K_{\mathcal{X}_1}, K_{\mathcal{X}_2} \in \mathbb{R}^{n\times n}$ that are the centered Gram matrices of some kernel functions defined over two random variables $\mathcal{X}_1$ and $\mathcal{X}_2$, the empirical estimate of HSIC is given by

$$d_{\mathrm{HSIC}}(\mathcal{X}_1, \mathcal{X}_2) = \mathrm{tr}\left(K_{\mathcal{X}_1}K_{\mathcal{X}_2}\right). \quad (11)$$

That is, the larger the $d_{\mathrm{HSIC}}(\mathcal{X}_1, \mathcal{X}_2)$, the stronger the statistical dependence between $\mathcal{X}_1$ and $\mathcal{X}_2$. In our case, using the idea of spectral embedding, we can consider the rows of the individual subspace representations U_i and U_j as two particular sets of sample points in $\mathbb{R}^k$, which are drawn from two probability distributions governed by the information on vertex connectivity in G_i and G_j, respectively. In other words, the sets of rows of U_i and U_j can be seen as realizations of two random variables $\mathcal{X}_i$ and $\mathcal{X}_j$. Therefore, we can define the Gram matrices of linear kernels on $\mathcal{X}_i$ and $\mathcal{X}_j$ as:

$$K_{\mathcal{X}_i} = (U_i{}')'(U_i{}') = U_iU_i{}', K_{\mathcal{X}_j} = (U_j{}')'(U_j{}') = U_jU_j{}'. \quad (12)$$

Combining (11) and (12), we can see that:

$$d_{\mathrm{HSIC}}(\mathcal{X}_i, \mathcal{X}_j) = \mathrm{tr}(U_iU_i{}'U_jU_j{}') = k - d^2_{\mathrm{proj}}(U_i, U_j). \quad (13)$$

This shows that the projection distance between subspaces U_i and U_j can be interpreted as the negative dependence between $\mathcal{X}_i$ and $\mathcal{X}_j$, which reflect the information provided by the two individual graph layers G_i and G_j.

Therefore, from both information-theoretic and statistical learning points of view, the smaller the projection distance between two subspace representations U_i and U_j, the more similar the information in the respective graphs that they represent. As a result, the representative subspace (the solution U to the problem of (6)) can be considered as a subspace representation that "summarizes" the information from the individual graph layers, and at the same time captures the intrinsic relationships between the vertices in the graph. As one can imagine, such relationships are of crucial importance in our multi-layer graph analysis.

V. Clustering on Multi-Layer Graphs

In Section IV, we introduced a novel framework for merging subspace representations from the individual layers of a multi-layer graph, which leads to a representative subspace that captures the intrinsic relationships between the vertices of the graph. This representative subspace provides a low dimensional form that can be used in several applications involving multi-layer graph analysis. In particular, we study now one such application, namely the problem of clustering vertices in a multi-layer graph[5]. We further analyze the behavior of the proposed clustering algorithm with respect to the properties of the individual graph layers (subspaces).

A. Clustering Algorithm

As we have already seen in Section III, the success of the spectral clustering algorithm relies on the transformation of the information contained in the graph structure into a spectral embedding computed from the graph Laplacian matrix, where each row of the embedding matrix (after normalization) is treated as the coordinates of the corresponding vertex in a low dimensional subspace. In our problem of clustering on a multi-layer graph, the setting is slightly different, since we aim at finding a unified clustering of the vertices that takes into account information contained in all the individual layers of the multi-layer graph. However, the merging framework proposed in the previous section can naturally be applied in this context. In fact, it leads to a natural solution to the clustering problem on multi-layer graphs. In more details, similarly to the spectral embedding matrix in the spectral cluttering algorithm, which is a subspace representation for one individual graph, our merging framework provides a representative subspace that contains the information from the multiple graph layers. Using this representation, we can then follow the same steps of spectral clustering to achieve the final clustering of the vertices with a k-means algorithm. The proposed clustering algorithm is summarized in Algorithm 2.

Algorithm 2: Spectral Clustering on Multi-Layer graphs (SC-ML)

1: **Input:**
$\{W_i\}_{i=1}^M$: $n \times n$ weighted adjacency matrices of individual graph layers $\{G_i\}_{i=1}^M$
k: target number of clusters
α: regularization parameter
2: Compute the normalized Laplacian matrix L_i and the subspace representation U_i for each G_i.
3: Compute the modified Laplacian matrix $L_{\mathrm{mod}} = \sum_{i=1}^M L_i - \alpha\sum_{i=1}^M U_iU_i{}'$.
4: Compute $U \in \mathbb{R}^{n\times k}$ that is the matrix containing the first k eigenvectors $u_1, \ldots, u_k$ of L_{mod}. Normalize each row of U to get U_{norm}.
5: Let $y_j \in \mathbb{R}^k (j = 1, \ldots, n)$ be the transpose of the j-th row of U_{norm}.
6: Cluster y_j in $\mathbb{R}^k$ into $C_1, \ldots, C_k$ using the k-means algorithm.
7: **Output:**
$C_1, \ldots, C_k$: The cluster assignment

[5]In addition to clustering, which is in an unsupervised fashion, the proposed framework can also be applied in a semi-supervised fashion, to learning problems such as classification. It can also be useful in ranking where an intrinsic relationship between objects, which is summarized from the individual graph layers, would certainly help.

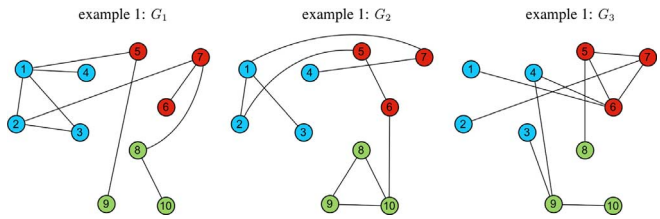


Fig. 2. A 3-layer graph with unit edge weights for toy example 1. The colors indicate the groundtruth clusters.

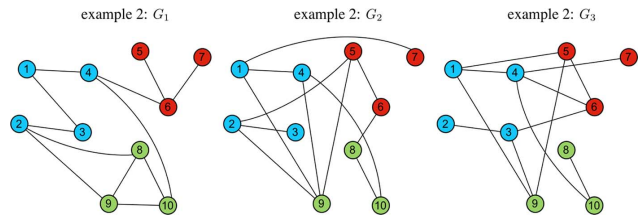


Fig. 3. A 3-layer graph with unit edge weights for toy example 2. The colors indicate the groundtruth clusters.

TABLE I
ANALYSIS OF TOY EXAMPLE 1

	layer G_1	layer G_2	layer G_3	SC-ML
NMI	0.6279	0.6181	0.2673	1.0000

(a)

	layer G_1	layer G_2	layer G_3	subspace computed by SC-ML
layer G_1	0	1.1100	1.3670	0.9456
layer G_2	1.1100	0	1.3354	1.0452
layer G_3	1.3670	1.3354	0	1.0788

(b)

TABLE II
ANALYSIS OF TOY EXAMPLE 2

	layer G_1	layer G_2	layer G_3	SC-ML
NMI	0.7934	0.2673	0.4728	0.5300

(a)

	layer G_1	layer G_2	layer G_3	subspace computed by SC-ML
layer G_1	0	1.3098	1.2296	1.0311
layer G_2	1.3098	0	0.9343	0.8828
layer G_3	1.2296	0.9343	0	0.5058

(b)

It is clear that Algorithm 2 is a direct generalization of Algorithm 1 in the case of multi-layer graphs. The main ingredient of our clustering algorithm is the merging framework proposed in Section IV, in which information from individual graph layers is summarized, prior to the actual clustering process (i.e., the k-means step) is implemented. This provides an example that illustrates how our generic merging framework can be applied to specific learning tasks on multi-layer graphs.

B. Analysis of the Proposed Algorithm

We now analyze the behavior of the proposed clustering algorithm under different conditions. Specifically, we first outline the link between subspace distance and clustering quality, and then compare the clustering performances in two scenarios where the relationships between the individual subspaces $\{U_i\}_{i=1}^M$ are different. Finally, we discuss about the relations between the choice of the number of clusters and the clustering performance.

As we have seen in Section IV, the rows of the subspace representations $\{U_i\}_{i=1}^M$ can be viewed as realizations of random variables $\{\mathcal{X}_i\}_{i=1}^M$ governed by the graph information. At the same time, spectral clustering directly utilizes U_i for the purpose of clustering. Therefore, $\{\mathcal{X}_i\}_{i=1}^M$ can be considered as random variables that control the cluster assignment of the vertices[6]. Since the projection distance can be understood as the negative statistical dependence between such random variables, the minimization of the projection distance in (6) is equivalent to the maximization of the dependence between the random variable from the representative subspace U and the ones from the individual subspaces $\{U_i\}_{i=1}^M$. The optimization in (6) can then be seen as a solution that tends to produce a clustering with the representative subspace that is consistent with those computed from the individual subspace representations.

We now discuss how the relationships between the individual subspaces possibly affect the performance of our clustering algorithm **SC-ML**. Intuitively, since the second term of the objective function in (6) represents the distance between the representative subspace U and all the individual subspaces $\{U_i\}_{i=1}^M$, it tends to drive the solution towards those subspaces that themselves are close to each other on the Grassmann manifold. To show it more clearly, let us consider two toy examples. The first example is illustrated in Fig. 2, where we have a 3-layer graph with the individual layers G_1, G_2 and G_3 sharing the same set of vertices. For the sake of simplicity, all the edge weights are set to one. In addition, three groundtruth clusters are indicated by the colors of the vertices. Table I(a) shows the performances of Algorithm 1 with individual layers as well as Algorithm 2 for the multi-layer graph[7], in terms of *Normalized Mutual Information (NMI)* [55] with respect to the groundtruth clusters. Table I(b) shows the projection distances between various pairs of subspaces. It is clear that the layers G_1 and G_2 produce better clustering quality, and that the distance between the corresponding subspaces is smaller. However, the vertex connectivity in layer G_3 is not very consistent with the groundtruth clusters and the corresponding subspace is further away from the ones from G_1 and G_2. In this case, the solution found by **SC-ML** is enforced to be close to the consistent subspaces from G_1 and G_2, hence provides satisfactory clustering results ($\mathrm{NMI} = 1$ represents perfect recovery of groundtruth clusters). Let us now consider a second toy example, as illustrated in Fig. 3. In this example we have two layers G_2 and G_3 with relatively low quality information with respect to the groundtruth clustering of the vertices. As

[6]The columns of U are a rotation of the columns of a cluster indicator matrix by $D^{1/2}$. It has been discussed in [3] that if there exist vertices of particularly low degrees, this rotation would make the columns of U differ from the indicator vectors. However, according to [3], one can argue that such low-degree vertices can be considered as outliers anyway, which does not affect much the global clustering quality. Therefore, the columns of U are quite informative about the global clustering structure.

[7]We choose the value of the regularization parameter α that leads to the best possible clustering performance. For the results presented in both Tables I and II, the regularization parameter is set to be 0.5. More discussions about the choices of this parameter are presented in Section VI.

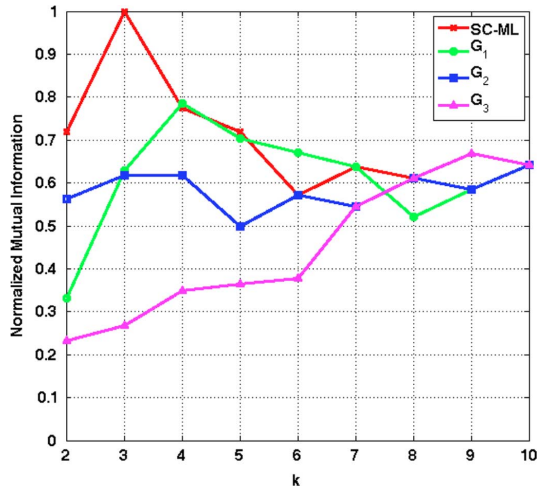


Fig. 4. Clustering performances of Algorithm 1 on the individual graph layers and Algorithm 2 on the multi-layer graph in the first toy example, under different values of k (number of clusters).

we see in Table II(b), their corresponding subspaces are close to each other on the Grassmann manifold. The most informative layer G_1, however, represents a subspace that is quite far away from the ones from G_2 and G_3. At the same time, we see in Table II(a) that the clustering results are better for the first layer than for the other two less informative layers. If the quality of the information in the different layers is not considered in computing the representative subspace, **SC-ML** enforces the solution to be closer to two layers of relatively lower quality, which results in unsatisfactory clustering performance in this case.

The analysis above implies that the proposed clustering algorithm works well under the following assumptions: (i) the majority of the individual subspaces are relatively informative, namely, they are helpful for recovering the groundtruth clustering, and (ii) they are reasonably close to each other on the Grassmann manifold, namely, they are expected to provide *complementary* but not *contradictory* information. When this is the case, the majority of informative views tend to agree with each other, and the information contained in these informative views is likely to be mainly captured by k-dimensional subspaces. The global clustering structure will then mainly be defined by the informative views that admit a value of k for clustering, even though k might not be optimal for each individual view independently (in fact we do not assume that k is optimal for each individual view. Therefore, without loss of generality, it is reasonable for the proposed framework to consider a universal k for the subspace dimension across different views).

In Algorithm 2 we assume that the target number of clusters k is known a priori. Although this is a reasonable assumption done in many popular clustering algorithms, there are practical situations where k is not defined a priori. Traditionally, in spectral methods we could use the eigen-gap of the graph Laplacian matrices as a heuristic to choose the number of clusters [3]. In the case of Algorithm 2, if the majority of the informative views agree with each other, one could estimate the number of clusters k such that the gap between the k-th and the $(k+1)$-th eigenvalues is reasonably large for all these views. We remark, however, that after the merging of multiple layers a particular value of k could emerge as a good choice, which is not necessarily optimal for all the individual views, as in the first toy example illustrated in Fig. 4. A more detailed analysis in this respect remains as our future work.

Finally, we note that we could use the information about the disagreement between views to tune the regularization parameters in the optimization problem to promote better final clustering quality. For example, if one view is significantly different or contradictory from other views, we tend to discard it or choose a rather small regularization parameter to attenuate its influence on the final clustering quality.

VI. Experimental Results

In this section, we evaluate the performance of the **SC-ML** algorithm presented in Section V on one synthetic and two real world datasets. We first describe the datasets that we use for the evaluation, and then explain the various clustering algorithms that we adopt in the performance comparisons. We finally present the results in terms of three evaluation criteria as well as some discussions.

A. Datasets

The first dataset that we use is a synthetic dataset, where we have three point clouds in $\mathbb{R}^2$ forming the English letters "N", "R" and "C" (shown in Fig. 5). Each point cloud is generated from a five-component Gaussian mixture model with different values for the mean and covariance of the Gaussian distributions[8], where each component represents a class of 500 points with specific color. A 5-nearest neighbor graph is then constructed for each point cloud by assigning the weight of the edges connecting two vertices (points) as the reciprocal of the Euclidean distance between them. This gives us a 3-layer graph of 2500 vertices, where each graph layer is from a point cloud forming a particular letter. The goal with this dataset is to recover the five clusters (indicated by five colors) of the 2500 vertices using the three graph layers constructed from the three point clouds.

The second dataset contains data collected during the Lausanne Data Collection Campaign [56] by the Nokia Research Center (NRC) in Lausanne. This dataset contains the mobile phone data of 136 users living and working in the Lake Léman region in Switzerland, recorded over a one-year period. Considering the users as vertices in the graph, we construct three graphs by measuring the proximities between these users in terms of GPS locations, Bluetooth scanning activities and phone communication. More specifically, for GPS locations and bluetooth scans, we measure how many times two users are sufficiently close geographically (within a distance of roughly 1 km), and how many times two users' devices have detected the same bluetooth devices, respectively, within 30-minute time

[8]For letter "N", the mean and covariance of the five components are $[-1\ 0]$, $[1\ 3]$, $[3\ 2]$, $[6\ 3]$, $[4\ 0]$, and $[1\ 0.3;\ 0.3\ 1]$, $[0.6\ 0.1;\ 0.1\ 0.5]$, $[0.5\ -0.1;\ -0.1\ 1.5]$, $[0.8\ 0.3;\ 0.3\ 0.4]$, $[0.5\ 0.2;\ 0.2\ 1.5]$, respectively. For letter "R", the mean and covariance of the five components are $[1\ 0]$, $[0\ 2]$, $[2\ 4]$, $[4\ 3]$, $[4\ 0]$, and $[0.8\ -0.2;\ -0.2\ 0.8]$, $[0.3\ 0.1;\ 0.1\ 0.7]$, $[1\ 0.3;\ 0.3\ 0.2]$, $[0.5\ -0.1;\ -0.1\ 1.5]$, $[1.2\ -0.4;\ -0.4\ 0.6]$, respectively. For letter "C", the mean and covariance of the five components are $[1\ 0]$, $[0\ 1]$, $[2\ 3]$, $[4\ 3]$, $[5\ -1]$, and $[1.2\ -0.6;\ -0.6\ 0.8]$, $[0.6\ 0.1;\ 0.1\ 0.5]$, $[1.2\ 0.3;\ 0.3\ 0.2]$, $[1.5\ -0.3;\ -0.3\ 0.5]$, $[1.6\ 0.3;\ 0.3\ 0.2]$, respectively.

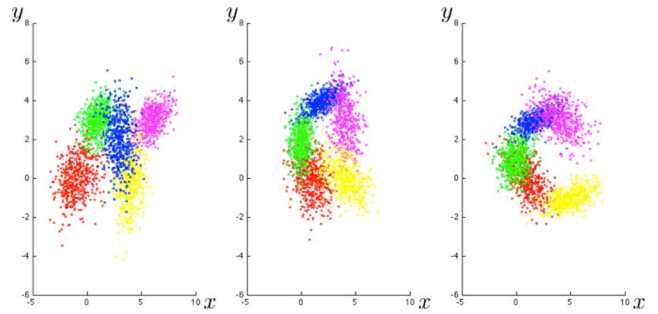


Fig. 5. Three five-class point clouds in $\mathbb{R}^2$ forming English letters "N", "R" and "C".

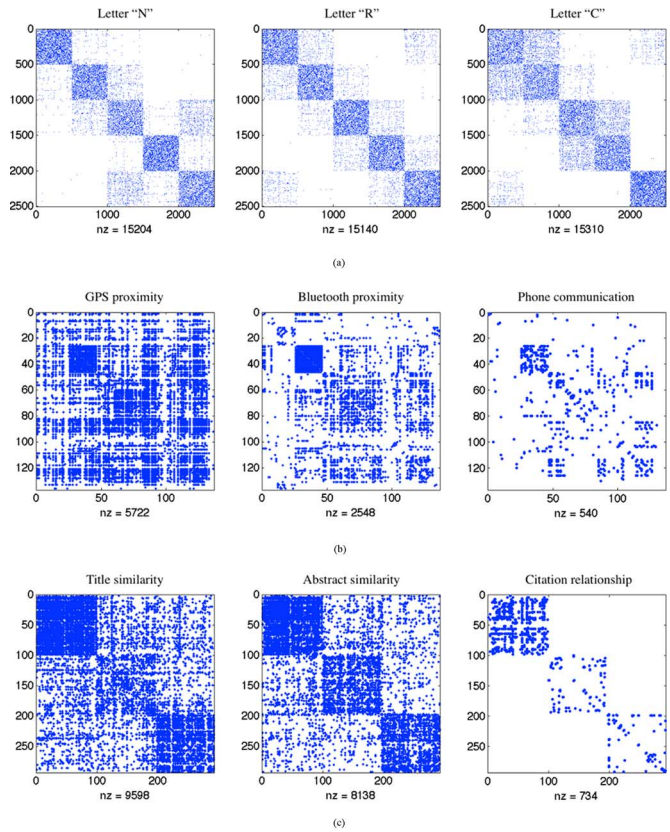


Fig. 6. Spy plots of three adjacency matrices in (a) the synthetic dataset, (b) the NRC dataset, and (c) the Cora dataset.

windows. Aggregating these results for a one-year period leads to two weighted adjacency matrices that represent the physical proximities of the users measured with different modalities. In addition, an adjacency matrix for phone communication is generated by assigning edge weights depending on the number of calls between any pair of two users. These three adjacency matrices form a 3-layer graph of 136 vertices, where the goal is to recover the eight groundtruth clusters that have been constructed from the users' email affiliations.

The third dataset is a subset of the Cora bibliographic dataset[9]. This dataset contains 292 research papers from three different fields, namely, natural language processing, data mining and robotics. Considering papers as vertices in the graph, we construct the first two graphs by measuring the similarities among the title and the abstract of these papers. More clearly, for both title and abstract, we represent each paper by a vector of non-trivial words using the *Term Frequency-Inverse Document Frequency (TF-IDF)* [55] weighting scheme, and compute the cosine similarities between every pair of vectors as the edge weights in the graphs. Moreover, we add a third graph which reflects the citation relationships among the papers, namely, we assign an edge with unit weight between papers A and B if A has cited or been cited by B. This results in a 3-layer graph of 292 vertices, and the goal in this dataset is to recover the three clusters corresponding to the different fields the papers belong to.

The adjacency matrices of the graphs are visualized as the spy plots shown in Fig. 6(a), (b) and (c) for the synthetic, NRC and Cora dataset, respectively, where the orderings of the vertices are made consistent with the groundtruth clusters[10]. A spy plot is a global view of a matrix where every non-zero entry in the matrix is represented by a blue dot. As shown in these figures, we see clearly the clusters in the synthetic and Cora datasets, while the clusters in the NRC dataset are not very clear. The reason for this is that, in the NRC dataset, the email affiliations used to create the groundtruth clusters only provide approximative information.

B. Clustering Algorithms

We now explain briefly the clustering algorithms in our comparative performance analysis along with some implementation details. We adopt three baseline algorithms as well as a state-of-the-art technique, namely the co-regularization approach introduced in [7]. As we shall see, there are interesting connections between the competitor clustering schemes and the proposed algorithm. First of all, we describe some implementation details of the proposed **SC-ML** algorithm and the co-regularization approach in [7]:

- **SC-ML**: Spectral Clustering on Multi-Layer graphs, as presented in Section V. The implementation of **SC-ML** is pretty straightforward, and the only parameter to choose is the regularization parameter α in (6). In our experiments, we choose the value of α through multiple empirical trials and report the peak and average performances of 20 test runs[11]. We will discuss the choice of this parameter later in this section.
- **SC-CoR**: Spectral Clustering with Co-Regularization proposed in [7]. We follow the same practice as in [7] to choose the most informative graph layer to initialize the alternating optimization scheme in **SC-CoR**. The stopping criteria for the optimization process is chosen such that the optimization stops when changes in the objective function are smaller than 10^{-5}. Similarly, we choose the value of the regularization parameter α in **SC-CoR** through multiple empirical trials. As in [7], the parameter α is fixed in the optimization steps for all graph layers.

[9]Available online at "http://people.cs.umass.edu/~mccallum/data.html" under category "Cora Research Paper Classification".

[10]The adjacency matrix for GPS proximity in the NRC dataset is thresholded for better illustration.

[11]The values of α that achieve the peak performances are 0.695, 0.42 and 0.44 for the synthetic, NRC and Cora datasets, respectively.

TABLE III
PEAK PERFORMANCES OF DIFFERENT CLUSTERING ALGORITHMS OUT OF 20 TEST RUNS ON ONE SYNTHETIC AND TWO REAL WORLD DATASETS

Algorithm	Purity (%)			NMI (%)			RI (%)		
	Synthetic	NRC	Cora	Synthetic	NRC	Cora	Synthetic	NRC	Cora
SC-Single	85.8	51.5	95.6	72.7	31.3	83.1	90.2	73.3	94.3
SC-Sum	97.3	54.4	96.9	91.8	36.1	86.6	97.9	77.1	96.0
SC-KSum	97.7	53.7	95.2	92.8	34.1	81.2	98.2	76.7	93.8
SC-CoR	97.8	58.1	**98.3**	92.8	40.6	**93.0**	98.3	78.8	**97.8**
SC-ML	**98.3**	**61.0**	**98.3**	**94.1**	**41.6**	91.8	**98.6**	**79.3**	**97.8**

TABLE IV
AVERAGE PERFORMANCES OF DIFFERENT CLUSTERING ALGORITHMS OUT OF 20 TEST RUNS ON ONE SYNTHETIC AND TWO REAL WORLD DATASETS. THE NUMBERS IN PARENTHESIS ARE THE STANDARD DEVIATIONS

Algorithm	Purity (%)			NMI (%)			RI (%)		
	Synthetic	NRC	Cora	Synthetic	NRC	Cora	Synthetic	NRC	Cora
SC-Single	84.2(5.0)	50.3(2.6)	95.6(0.0)	71.3(4.2)	32.7(2.1)	83.1(0.0)	89.4(2.4)	74.4(1.6)	94.3(0.0)
SC-Sum	96.4(4.2)	52.7(2.7)	93.7(9.8)	91.3(2.3)	34.7(1.9)	83.4(9.8)	97.5(1.9)	76.7(1.0)	93.6(7.5)
SC-KSum	95.8(6.0)	53.1(1.9)	95.2(0.0)	91.7(3.4)	33.9(1.6)	81.2(0.0)	97.3(2.7)	76.8(1.0)	93.8(0.0)
SC-CoR	96.7(4.6)	57.8(1.9)	**98.3(0.0)**	91.9(3.1)	**39.7(1.9)**	**93.0(0.0)**	97.8(2.0)	77.9(1.4)	**97.8(0.0)**
SC-ML	**98.2(0.0)**	**58.0(1.5)**	**98.3(0.0)**	**93.8(0.0)**	38.7(1.4)	91.8(0.0)	**98.6(0.0)**	**78.4(0.6)**	**97.8(0.0)**

Next, we introduce three baseline comparative algorithms that work as follows:

- **SC-Single**: Spectral Clustering (Algorithm 1) applied on a single graph layer, where the graph is chosen to be the one that leads to the best clustering results.
- **SC-Sum**: Spectral clustering applied on a global matrix W that is the summation of the normalized adjacency matrices of the individual layers:

$$W = \sum_{i=1}^{M} D_i^{-\frac{1}{2}} W_i D_i^{-\frac{1}{2}}. \tag{14}$$

- **SC-KSum**: Spectral clustering applied on the summation K of the spectral kernels [6] of the adjacency matrices:

$$K = \sum_{i=1}^{M} K_i \quad \text{with} \quad K_i = \sum_{m=1}^{d} u_{im} u_{im}{}', \tag{15}$$

where n is the number of vertices, $d \ll n$ is the number of eigenvectors used in the definition of the spectral kernels K_i, and u_{im} represents the m-th eigenvector of the Laplacian L_i for graph G_i. To make it more comparable with spectral clustering, we choose d to be the target number of clusters in our experiments.

C. Clustering Results

We evaluate the performance of the different clustering algorithms with three different criteria, namely *Purity*, *Normalized Mutual Information (NMI)* and *Rand Index (RI)* [55]. Specifically, let $\Omega = \{\omega_1, \ldots, \omega_k\}$ be the computed clusters and $C = \{c_1, \ldots, c_k\}$ be the intended groundtruth classes. First, *Purity* is defined as:

$$\text{Purity}(\Omega, C) = \frac{1}{n} \sum_{k} \max_{j} |\omega_k \cap c_j|, \tag{16}$$

where n is the total number of objects, and $|\omega_k \cap c_j|$ denotes the number of objects in the intersection of ω_k and c_j. Next, *Normalized Mutual Information* is defined as:

$$\text{NMI}(\Omega, C) = \frac{I(\Omega; C)}{[H(\Omega) + H(C)]/2}, \tag{17}$$

where I is the mutual information between clusters Ω and classes C, and $H(\Omega)$ and $H(C)$ represent the entropies of the clusters and classes, respectively. Finally, when interpreting clustering as a series of binary decisions on each pair of objects, *Rand Index* is defined as:

$$\text{RI}(\Omega, C) = \frac{TP + TN}{TP + FP + FN + TN}, \tag{18}$$

where TP, TN, FP, FN represent true positive, true negative, false positive and false negative decisions, respectively.

The clustering results are summarized in Tables III and IV for the peak and average performances of all the algorithms out of 20 test runs, respectively. For each scenario, the best result is highlighted in bold font. First, as expected, we see that the clustering performances for the synthetic and Cora datasets are higher than that for the NRC dataset, which indicates that the latter one is indeed more challenging due to the approximative groundtruth information. Second, it is clear that **SC-ML** and **SC-CoR** generally outperform the baseline approaches for the three datasets. More specifically, although both **SC-Sum** and **SC-KSum** indeed improve the clustering quality compared to clustering with individual graph layers, they only provide limited improvement, and the potential drawback for both of the summation methods is that they can be considered as similar to building a simple average graph for representing the different layers of information. Therefore, depending on data characteristics in specific datasets, this might smooth out the particular information provided by individual layers, and thus penalize the clustering performance. In comparison, **SC-ML** and

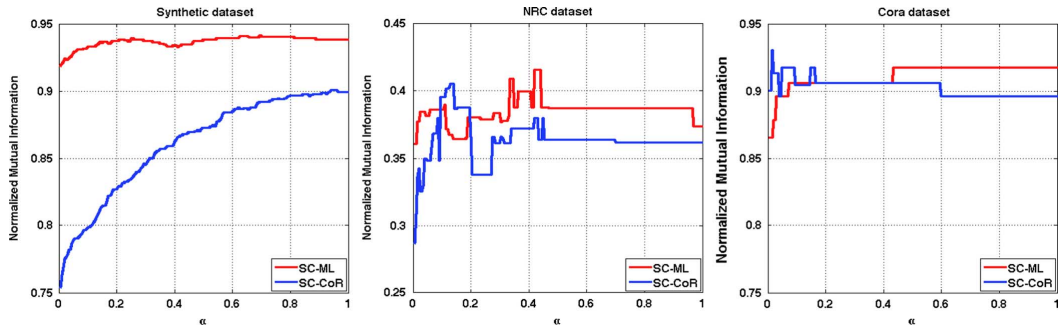


Fig. 7. Comparison of performances (in terms of *NMI*) of **SC-ML** and **SC-CoR** under different values of parameter α in the corresponding implementations.

TABLE V
CONFUSION MATRICES FOR DIFFERENT CLUSTERING ALGORITHMS ON THE NRC DATASET

SC-Single

3	4	0	1	2	0	0	9
1	**3**	0	0	0	0	0	2
1	0	**14**	0	0	0	6	0
0	1	0	**4**	4	0	0	4
0	1	0	1	**17**	3	0	12
0	0	0	0	1	**0**	0	1
0	0	0	0	0	0	**0**	2
1	7	1	5	5	2	3	**15**

diagonal sums up to 56

SC-Sum

8	1	2	0	2	5	0	1
3	**1**	0	0	0	0	1	1
0	0	**21**	0	0	0	0	0
1	1	0	**9**	1	0	0	1
9	0	0	3	**12**	0	1	9
0	1	0	0	1	**0**	0	0
0	0	0	0	1	0	**1**	0
8	6	5	10	4	0	2	**4**

diagonal sums up to 56

SC-KSum

3	3	0	1	4	1	0	7
0	**3**	0	0	0	0	0	3
0	0	**20**	0	1	0	0	0
1	1	0	**9**	1	0	0	1
0	1	0	3	**13**	6	4	7
1	0	0	0	1	**0**	0	0
0	1	0	0	1	0	**0**	0
1	7	4	10	4	3	0	**10**

diagonal sums up to 58

SC-ML

7	4	1	0	1	3	1	2
1	**3**	0	0	0	0	0	2
0	0	**20**	0	0	1	0	0
1	0	0	**10**	1	0	1	0
0	1	0	5	**20**	0	8	0
1	0	0	0	1	**0**	0	0
0	1	0	0	1	0	**0**	0
2	5	5	13	4	0	3	**7**

diagonal sums up to 67

SC-CoR

11	2	2	0	1	2	1	0
3	**1**	0	0	0	1	0	1
0	0	**21**	0	0	0	0	0
1	0	0	**10**	1	0	1	0
0	0	0	3	**12**	1	8	10
1	0	0	0	0	**0**	0	1
0	0	0	0	0	0	**0**	2
6	5	5	10	1	2	3	**7**

diagonal sums up to 62

SC-CoR always achieve significant improvements in the clustering quality compared to clustering using individual graph layers. Third, **SC-ML** achieves very competitive performances compared to **SC-CoR** for all the three evaluation criteria on the three datasets, with a much simpler implementation scheme and lower computational complexity, which we will explain in more details in the following section.

In addition to the clustering benchmarks provided above, we have computed and shown in Table V the confusion matrices based on the outcomes of the five clustering algorithms on the NRC dataset, as an illustrative example of the clustering qualities. The columns of the confusion matrices represent the predicted clusters while the rows represent the intended classes. The diagonal entries represent the numbers of objects that have been correctly identified for each class. By summing up the diagonal entries, it is clear that overall **SC-ML** best reveals the eight classes in the groundtruth data.

Finally, the eigen-gap is considered as a heuristic indicator of the clusterability of the vertices into k subsets. To understand better the benefits of multi-layer graph clustering compared to clustering with individual graph layers, we have computed the gap between the k-th and the $(k+1)$-th eigenvalues of the graph Laplacian matrices corresponding to individual and merged graph layers, where k is the target number of clusters.

TABLE VI
EIGEN-GAP BETWEEN THE k-TH AND THE $(k+1)$-TH EIGENVALUES OF THE GRAPH LAPLACIAN MATRICES FROM THE INDIVIDUAL GRAPH LAYERS AND THE "VIRTUAL" GRAPHS COMPUTED BY **SC-CoR** AND **SC-ML** ON DIFFERENT DATASETS

eigen-gap between the k-th and (k+1)-th eigenvalues	layer G_1	layer G_2	layer G_3	SC-CoR	SC-ML
Synthetic dataset	0.0010	0.0005	0.0008	0.2140	0.2146
NRC dataset	0.0116	0.0084	0.0087	0.0125	0.0250
Cora dataset	0.0384	0.1163	0.0000	0.2792	0.2954

Specifically, although the proposed merging framework does not lead directly to a graph topology, but rather a representative subspace, we created a "virtual" graph by using a Gaussian kernel together with the Euclidean distances between the low dimensional representations of every pair of vertices in the representative subspace. The same method can be used to create a "virtual" graph in the co-regularization approach. We then compared the eigen-gaps of the individual graph Laplacian matrices and the eigen-gaps computed using the "virtual" graphs, and the results for different datasets are shown in Table VI. As we can see, clustering on multi-layer graphs always leads to larger eigen-gaps, which is indicative of better clustering structures.

D. Further Discussions

We now present some discussions on parameter selection in **SC-ML** and its connections to the competitor clustering schemes. First of all, we discuss the influence of the choice of the regularization parameter α on the performance of **SC-ML**. In Fig. 7, we compare the performances of **SC-ML** and **SC-CoR** in terms of *NMI* under different values of parameter α in the corresponding implementations. As we can see, in our experiments, **SC-ML** achieves the best performances when α is chosen between 0.4 and 0.6, and it outperforms **SC-CoR** for a large range of α for the synthetic and NRC datasets. For the Cora dataset, the two algorithms achieve very similar performances at different values of α, but **SC-ML** permits a larger range of parameter selection. Furthermore, it is worth noting that the optimal values for α in **SC-ML** lie in similar ranges across different datasets, thanks to the adoption of the normalized graph Laplacian matrix whose spectral norm is upper bounded by 2. In summary, this shows that the performance of **SC-ML** is reasonably stable with respect to the parameter selection.

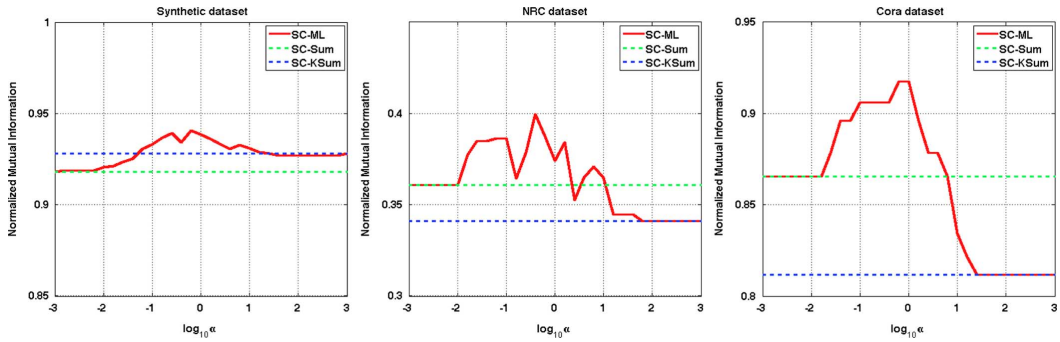


Fig. 8. Comparison between performances (in terms of *NMI*) of **SC-ML**, **SC-Sum** and **SC-KSum** under different values of parameter α.

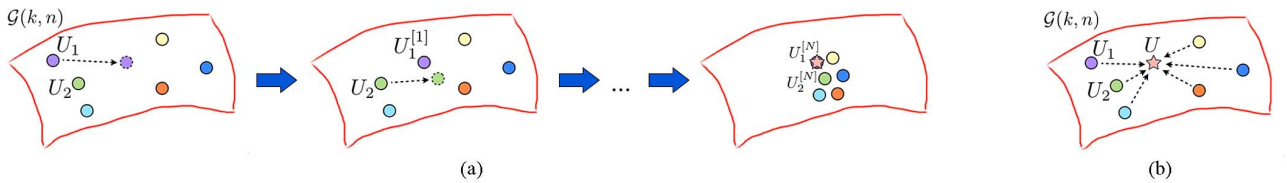


Fig. 9. Illustrations of graph layer merging. (a) **SC-CoR**: iterative update of the individual subspace representations. The superscript $[N]$ represents the number of iterative steps on each individual subspace representation. The final update of the subspace representation for the most informative graph layer ($U_1^{[N]}$, shown as a star) is considered as a good merging solution; (b) **SC-ML**: the representative subspace (U, shown as a star) is found in one step.

The role of the parameter α can also be understood by comparing **SC-ML** with **SC-Sum** and **SC-KSum**. Specifically, **SC-Sum** can be considered as taking an average of the graph Laplacian matrices of the individual graph layers, while **SC-KSum** takes the average of the corresponding low-dimensional spectral representations. These are exactly the two parts of the objective function in the optimization problem of (6), whose relative importance is weighted by α. Theoretically, on the one hand, if we set α to be zero, the solution of the problem becomes equivalent to the one found by **SC-Sum**; on the other hand, if we let α go to infinity, then the solution becomes equivalent to the one found by **SC-KSum**. This intuition is confirmed by the results shown in Fig. 8. As we can see, **SC-ML** achieves the same performances as **SC-Sum** and **SC-KSum** when α is chosen to be 0 and very large, respectively. More importantly, for a wide and stable range of choices of α, it leads to better clustering performances than these two baseline schemes.

Finally, we take a closer look at the comparisons between **SC-ML** and **SC-CoR**. Although the latter is not developed from the viewpoint of subspace analysis on the Grassmann manifold, it can actually be interpreted as a process in which individual subspace representations are updated based on the same distance analysis as in our framework. In this sense, **SC-CoR** uses the same distance as ours to measure similarities between subspaces. The merging solution however leads to a different optimization problem than that of (6), which is based on a slightly different merging philosophy. Specifically, it enforces the information contained in the individual subspace representations to be consistent with each other. An alternating optimization scheme optimizes, at each step, one subspace representation, while fixing the others. This can be interpreted as a process in which one subspace at each step becomes closer to other subspaces in term of the projection distance on the Grassmann manifold. Upon convergence, all initial subspaces are "brought" closer to each other and the final subspace representation from the most informative graph layer is considered as the one that combines information from all the graph layers efficiently. Two illustrations of **SC-CoR** and **SC-ML** are shown in Fig. 9(a) and (b), respectively. Therefore, on the one hand, results for both approaches demonstrate the benefit of using our distance analysis on the Grassmann manifold for merging information in multi-layer graphs. Indeed, for both approaches, since the distances between the solutions and the individual subspaces are minimized without sacrificing too much of the information from individual graph layers, the resulting combinations can be considered as good summarizations of the multiple graph layers. On the other hand, however, **SC-ML** differs from **SC-CoR** mainly in the following aspects. First, the alternating optimization scheme in **SC-CoR** focuses only on optimizing one subspace representation at each step, and it requires a sensible initialization to guarantee that the algorithm ends up at a good local minimum for the optimization problem; it also does not guarantee that all the subspace representations converge to one point on the Grassmann manifold (it uses the final update of the most informative layer for clustering)[12]. In contrast, **SC-ML** directly finds a single representation through a unique optimization of the representative subspace with respect to all graph layers jointly, which does not need alternating optimization steps and careful initializations. These are the possible reasons why **SC-ML** performs slightly better than **SC-CoR** for the synthetic and NRC datasets in our experiments. Second, it is worth noting from a computational point of view that, the performance improvements are achieved with lower computational complexity, since the optimization process involved

[12]In [7], the authors have also proposed a "centroid-based co-regularization approach" that introduces a consensus representation. However, such a representation is still computed via an alternating optimization scheme, which needs a sensible initialization and keeps the same iterative nature.

in **SC-ML** is much simpler than that in **SC-CoR**. Specifically, the iterative nature of **SC-CoR** requires solving an eigenvalue problem for MN times, where M and N are the number of individual graphs and the number of iterations needed for the algorithm to converge, respectively. In contrast, since **SC-ML** aims at finding a globally representative subspace without modifying the individual ones, it needs to solve an eigenvalue problem only once.

VII. Conclusion

In this paper, we provide a framework for analyzing information provided by multi-layer graphs and for clustering vertices of graphs in rich datasets. Our generic approach is based on the transformation of information contained in the individual graph layers into subspaces on the Grassmann manifold. The estimation of a representative subspace can then be essentially considered as the problem of finding a good summarization of multiple subspaces using distance analysis on the Grassmann manifold. The proposed framework can be applied to various learning tasks where multiple subspace representations are involved. Under appropriate and realistic assumptions, we show that it leads to a novel clustering algorithm on multi-layer graphs that is competitive to the state-of-the-art techniques. Finally, we mention the following research directions as interesting and open problems. First, as the subspace representation inspired by spectral clustering is not the only valid representation for the graph information (alternatively we can consider eigenvectors of the modularity matrix of the graph as suggested in [11], [12]), an interesting problem is to find the most appropriate subspace representation for the data available, either they are in the forms of graphs or they are of more general forms. Second, we believe that better clustering performance can be achieved if specific prior knowledge on the data is available, in particular the consistency of the information in the different graph layers and their relative importances. These problems are however left for future studies.

References

[1] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[2] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2001, vol. 14, pp. 849–856.

[3] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, Dec. 2007.

[4] J. Hamm and D. Lee, "Extended Grassmann kernels for subspace-based learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2008, vol. 21, pp. 601–608.

[5] A. Gretton, O. Bousquet, A. Smola, and B. Scholkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," in *Proc. Int. Conf. Algorithmic Learn. Theory*, 2005, pp. 63–77.

[6] W. Tang, Z. Lu, and I. Dhillon, "Clustering with multiple graphs," in *Proc. IEEE Int. Conf. Data Min. (ICDM)*, 2009, pp. 1016–1021.

[7] A. Kumar, P. Rai, and H. Daumé, III, "Co-regularized multi-view spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2011, vol. 24, pp. 1413–1421.

[8] E. Schaeffer, "Survey: Graph clustering," *Comput. Sci. Rev.*, vol. 1, no. 1, pp. 27–64, Aug. 2007.

[9] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, no. 3–5, pp. 75–174, Feb. 2010.

[10] Z. Zhang and M. Jordan, "Multiway spectral clustering: A margin based perspective," *Statist. Sci.*, vol. 23, no. 3, pp. 383–403, 2008.

[11] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E*, vol. 74, no. 3, Sep. 2006.

[12] M. E. J. Newman, "Modularity and community structure in networks," *Proc. Natl. Acad. Sci. USA*, vol. 103, no. 23, pp. 8577–8582, Jun. 2006.

[13] M. Saerens, F. Fouss, L. Yen, and P. Dupont, "The principal components analysis of a graph, and its relationships to spectral clustering," in *Proc. Eur. Conf. Mach. Learn. (ECML)*, 2004, pp. 371–383.

[14] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *J. Opt. Soc. Amer. A*, vol. 4, no. 3, pp. 519–524, Mar. 1987.

[15] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve procedure for the characterization of human faces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 1, pp. 103–108, Jan. 1990.

[16] M. Turk and A. P. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 71–86, Winter 1991.

[17] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal. Appl.*, vol. 20, no. 2, pp. 303–353, 1998.

[18] Y. Chikuse, "Statistics on special manifolds," in *Lecture Notes in Statistics*. New York, NY, USA: Springer, 2003, vol. 174.

[19] J. Hamm and D. Lee, "Grassmann discriminant analysis: A unifying view on subspace-based learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 376–383.

[20] X. Liu, A. Srivastava, and K. Gallivan, "Optimal linear representations of images for object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 662–666, May 2004.

[21] D. Lin, S. Yan, and X. Tang, "Pursuing informative projection on Grassmann manifold," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR)*, 2006, vol. 2, pp. 1727–1734.

[22] P. Turaga, A. Veeraraghavan, and R. Chellappa, "Statistical analysis on Stiefel and Grassmann manifolds with applications in computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR)*, 2008.

[23] M. T. Harandi, C. Sanderson, S. Shirazi, and B. C. Lovell, "Graph embedding discriminant analysis on Grassmannian manifolds for improved image set matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR)*, 2011, pp. 2705–2712.

[24] X. Wang, D. Tao, and Z. Li, "Subspaces indexing model on Grassmann manifold for image search," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2627–2635, Sep. 2011.

[25] A. Argyriou, M. Herbster, and M. Pontil, "Combining graph Laplacians for semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2005, vol. 18, pp. 67–74.

[26] D. Zhou and C. Burges, "Spectral clustering and transductive learning with multiple views," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 1159–1166.

[27] L. Tang, X. Wang, and H. Liu, "Community detection via heterogeneous interaction analysis," *Data Min. Knowl. Discov.*, vol. 25, no. 1, pp. 1–33, Jul. 2012.

[28] B. McFee and G. Lanckriet, "Learning multimodal similarity," *J. Mach. Learn. Res.*, vol. 12, pp. 491–523, 2011.

[29] Z. Akata, C. Thurau, and C. Bauckhage, "Non-negative matrix factorization in multimodality data for segmentation and label prediction," presented at the Comput. Vis. Winter Workshop, Mitterberg, Styria, Austria, Feb. 2011.

[30] X. Cai, F. Nie, H. Huang, and F. Kamangar, "Heterogeneous image feature integration via multi-modal spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR)*, 2011, pp. 1977–1984.

[31] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "Clustering with multi-layer graphs: A spectral perspective," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5820–5831, Nov. 2012.

[32] D. Eynard, K. Glashoff, M. M. Bronstein, and A. M. Bronstein, "Multimodal diffusion geometry by joint diagonalization of Laplacians," 2012 [Online]. Available: http://arxiv.org/abs/1209.2295, arXiv:1209.2295

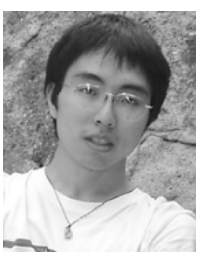
[33] J. Liu, C. Wang, J. Gao, and J. Han, "Multi-view clustering via joint nonnegative matrix factorization," in *Proc. SIAM Int. Conf. Data Min.*, 2013, pp. 252–260.

[34] T. Xia, D. Tao, T. Mei, and Y. Zhang, "Multiview spectral embedding," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 6, pp. 1438–1446, Dec. 2010.

[35] M. B. Blaschko and C. H. Lampert, "Correlational spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR)*, 2008.

[36] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via canonical correlation analysis," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 129–136.

[37] S. Bickel and T. Scheffer, "Multi-view clustering," in *Proc. IEEE Int. Conf. Data Min. (ICDM)*, 2004, pp. 19–26.
[38] A. Kumar and H. Dauḿ, III, "A co-training approach for multi-view spectral clustering," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 393–400.
[39] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th Annu. Conf. Comput. Learn. Theory*, 1998, pp. 92–100.
[40] V. Sindhwani and P. Niyogi, "A co-regularization approach to semisupervised learning with multiple views," in *Proc. ICML Workshop on Learn. With Multiple Views*, 2005.
[41] P. Muthukrishnan, D. Radev, and Q. Mei, "Edge weight regularization over multiple graphs for similarity learning," in *Proc. IEEE Int. Conf. Data Min. (ICDM)*, 2010, pp. 374–383.
[42] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "A regularization framework for mobile social network analysis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2011, pp. 2140–2143.
[43] V. R. de Sa, "Spectral clustering with two views," in *Proc. ICML Workshop Learn. With Multiple Views*, 2005.
[44] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Dec. 2002.
[45] E. Bruno and S. Marchand-Maillet, "Multiview clustering: A late fusion approach using latent models," in *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2009, pp. 736–737.
[46] D. Greene and P. Cunningham, "A matrix factorization approach for integrating multiple data views," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discov. Databases: Pt. I*, 2009, pp. 423–438.
[47] Y. Cheng and R. Zhao, "Multiview spectral clustering via ensemble," in *Proc. IEEE Int. Conf. Granular Comput.*, 2009, pp. 101–106.
[48] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, Mar. 1951.
[49] F. R. K. Chung, "Spectral graph theory," in *CBMS Regional Conf. Series in Math., Amer. Math. Soc.*, 1997.
[50] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probabil.*, 1967, vol. 1, pp. 281–297.
[51] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 1996.
[52] D. C. Sorensen, "Implicit application of polynomial filters in a k-step Arnoldi method," *SIAM J. Matrix Anal. Appl.*, vol. 13, no. 1, pp. 357–385, Jan. 1992.
[53] R. B. Lehoucq and D. C. Sorensen, "Deflation techniques for an implicitly restarted Arnoldi iteration," *SIAM J. Matrix Anal. Appl.*, vol. 17, no. 4, pp. 789–821, Oct. 1996.
[54] Z. Ghahramani and G. E. Hinton, "The EM algorithm for mixtures of factor analyzers," Univ. of Toronto, Toronto, ON, Canada, Tech. Rep. CRG-TR-96-1, May 1996.
[55] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[56] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, and J. Laurila, "Towards rich mobile phone datasets: Lausanne data collection campaign," in *Proc. 7th Int. Conf. Pervasive Services*, 2010.



Xiaowen Dong has been working as a Ph.D. student in the Signal Processing Laboratories (LTS4/LTS2) at Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland since October 2009. His research interests include wavelets, spectral graph theory, and their applications to mobile and online social network analysis.

Before joining EPFL, Mr. Dong received his B.Eng. degree in information engineering from Zhejiang University, Hangzhou, China and his M.Sc. degree in signal processing from Institute for Digital Communications, The University of Edinburgh, Edinburgh, UK.

Pascal Frossard (S'96–M'01–SM'04) received the M.S. and Ph.D. degrees, both in electrical engineering, from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 1997 and 2000, respectively. Between 2001 and 2003, he was a member of the research staff at the IBM T. J. Watson Research Center, Yorktown Heights, NY, where he worked on media coding and streaming technologies. Since 2003, he has been a faculty at EPFL, where he heads the Signal Processing Laboratory (LTS4). His research interests include image representation and coding, visual information analysis, distributed image processing and communications, media streaming systems, and dimensionality reduction problems.

Pierre Vandergheynst received the M.S. degree in physics and the Ph.D. degree in mathematical physics from the Université catholique de Louvain, Louvain-la-Neuve, Belgium, in 1995 and 1998, respectively. From 1998 to 2001, he was a Postdoctoral Researcher with the Signal Processing Laboratory, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. He was Assistant Professor at EPFL (2002–2007), where he is now an Associate Professor.

His research focuses on harmonic analysis, sparse approximations and mathematical data processing in general with applications covering signal, image and high dimensional data processing, sensor networks, computer vision.

He was co-Editor-in-Chief of *Signal Processing* (2002–2006) and Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING (2007–2011), the flagship journal of the signal processing community. He has been on the Technical Committee of various conferences, serves on the steering committee of the SPARS workshop and was co-General Chairman of the EUSIPCO 2008 conference.

Pierre Vandergheynst is the author or co-author of more than 70 journal papers, one monograph and several book chapters. He has received two IEEE best paper awards.

Professor Vandergheynst is a laureate of the Apple 2007 ARTS award and of the 2009–2010 De Boelpaepe prize of the Royal Academy of Sciences of Belgium.

Nikolai Nefedov received M.Sc. degree in radio-physics with Honors (1982) from St. Petersburg State Technical University and Ph.D. degree in communications theory (1989) from St. Petersburg State University of Telecommunications (Russia). During 1982–1985 and 1988–1993 he worked at different research positions in Russian Academy of Science in Institute of Nuclear Physics (St. Petersburg) and Institute of Atmospheric Physics (Moscow). He was with ATMEL Development Center/Finland during 1993–1997, where he was involved in ASIC and DSP design. During 1996–2005 he hold different positions in Helsinki University of Technology, Adjunct Professor since 2004.

He joined Nokia Research Center in 1997 where he worked on wireless communication systems (1997–2005, Finland) and on emerging technologies including complex networks analysis as Principal Scientist (2005–2012, Switzerland). Since 2012 he is Senior Scientist at ISI Laboratory, ETH Zurich.

Dr. Nefedov has been on the Technical Committee of various conferences in field of wireless communications and networks, he is the author or co-author of more than 60 publications and more than 20 patents and patent applications.

His current activities include interdisciplinary research in field of statistical physics, large-scale dynamical systems, machine learning, and distributed algorithms.