Learning the Right Layers: a Data-Driven Layer-Aggregation Strategy for Semi-Supervised Learning on Multilayer Graphs

Sara Venturini [1] **Andrea Cristofari** [2] **Francesco Rinaldi** [1] **Francesco Tudisco** [3]

Abstract

Clustering (or community detection) on multilayer graphs poses several additional complications with respect to standard graphs as different layers may be characterized by different structures and types of information. One of the major challenges is to establish the extent to which each layer contributes to the cluster assignment in order to effectively take advantage of the multilayer structure and improve upon the classification obtained using the individual layers or their union. However, making an informed a-priori assessment about the clustering information content of the layers can be very complicated. In this work, we assume a semi-supervised learning setting, where the class of a small percentage of nodes is initially provided, and we propose a parameter-free Laplacian-regularized model that learns an optimal nonlinear combination of the different layers from the available input labels. The learning algorithm is based on a Frank-Wolfe optimization scheme with inexact gradient, combined with a modified Label Propagation iteration. We provide a detailed convergence analysis of the algorithm and extensive experiments on synthetic and real-world datasets, showing that the proposed method compares favourably with a variety of baselines and outperforms each individual layer when used in isolation.

1. Introduction

Graph-based Semi-Supervised Learning (GSSL) has achieved great success in various real-world applications, where only a relatively small amount of labeled samples are available (Song et al., 2022). Given a graph and a set of initially labeled nodes, the aim of GSSL is to infer the labels of the remaining unlabeled nodes. To this end, exploiting the graph structure is particularly important, especially when the percentage of input labels (size of the training dataset) is small and when no features are available for the nodes. Based on the so-called smoothness assumption, one of the most successful approaches for GSSL relies on a Laplacian regularization formulation, where we aim at minimizing a loss function that simultaneously forces consistency with the initial labels and with the graph structure. The minimizer can be interpreted as a new node embedding, which is then used to classify the unlabeled nodes. After the pioneering work by Zhou et al. (2003); Belkin et al. (2004); Yang et al. (2016), this approach has been widely explored in the machine learning literature (Hein et al., 2013; Welling & Kipf, 2016; Gasteiger et al., 2018; Mercado et al., 2019; Huang et al., 2020; Tudisco et al., 2021; Prokopchik et al., 2022).

While graphs are a popular and successful tool to model data interactions, many empirical systems and real-world datasets are characterized by multiple types of interactions or relationships simultaneously, and are actually better described by multilayer graphs (De Domenico et al., 2013; Gao et al., 2012; Wang et al., 2015). For instance, transportation systems are characterized by different transportation means such as train, bus, etc (De Domenico et al., 2014), scientific data is characterized by co-authorship, co-citation, as well as topic and institution affinities (Higham et al., 2022), people in a social environment interact at different layers such as friendship, acquaintance or business, etc (Dickison et al., 2016). Also, many biological systems are characterized by multiple types of relationships among their constituents (Bentley et al., 2016; Mangioni et al., 2018). Multilayer graphs are a standard representation of such data and directly modeling these multilayer interactions has led to improvements in a number of network science and machine learning problems (Bianconi, 2018; Magnani et al., 2021).

Even though potentially very useful and powerful, multilayer graph models pose an intrinsic fundamental challenge. Multilayer graphs can have a large number of layers describing a variety of different properties, however, it is a-priori

[1]Department of Mathematics "Tullio Levi-Civita", University of Padova, Padova, Italy [2]Department of Civil Engineering and Computer Science Engineering, University of Rome "Tor Vergata", Rome, Italy [3]School of Mathematics, Gran Sasso Science Institute, 67100 L'Aquila, Italy. Correspondence to: Sara Venturini <sara.venturini@math.unipd.it>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

not clear whether all the layers are actually useful to classify the nodes. Layers may carry the same or complementary clustering information, some layers may be more informative than others, and certain layers can potentially be just noise (i.e. they carry no information about the node clusters). Deciding which of these situations better describes a given dataset and identifying which are the most (and the least) informative layers is both very useful and highly challenging. In fact, the construction of the networks in many applications is not straightforward, and making an informed a-priori assessment on the presence of noise, the different types of layer structures, and the clustering information content in general, can be very complicated (Bentley et al., 2016; Choobdar et al., 2019; Peel et al., 2022).

Several GSSL algorithms for multilayer networks have been developed in recent years. The majority of these methods propose to aggregate the information carried by the different layers into a single-layer graph, using different forms of aggregation functions, such as sum, min, max, etc (Tsuda et al., 2005; Argyriou et al., 2005; Zhou & Burges, 2007; Kato et al., 2008; Karasuyama & Mamitsuka, 2013; Nie et al., 2016; Mondragon et al., 2018; Ye & Akoglu, 2018; Mercado et al., 2019; Viswanathan et al., 2019; Bergermann et al., 2021). However, most of the time, the proposed combinations are meant to be effective for a particular setting and thus require a-priori knowledge of the type of clustering information the layers and the whole dataset carry along. Moreover, even though some methods perform well in more than one setting, e.g. (Mercado et al., 2019; Gujral & Papalexakis, 2018), they do not provide information on whether certain layers are more informative than others, whether the information is complementary or not, or whether there is some uninformative (noisy) layer.

In this work, we propose a parameter-free Laplacian-regularized model that learns an optimal combination of the different layers from the available input labels. In our model, the layers are combined via nonlinear generalized mean functions which include as special cases several aggregation functions previously used in the literature. The optimal aggregation parameters are computed via a tailored bilevel inexact-gradient optimization scheme. We provide a detailed convergence analysis of the optimization method, which we also extensively test numerically. Our tests on synthetic and real-world datasets show that the resulting GSSL method for multilayer networks very favorably compares with available alternatives in terms of accuracy performance and that it succeeds in identifying the most relevant and least relevant layers, as well as complementary information across the layers.

2. Related Work on Multilayer GSSL

In this section, we provide a brief review of available semi-supervised learning algorithms for multilayer graphs, mostly focusing on those designed to work on feature-less multilayer networks.

Similar to our model, several approaches are based on learning an optimal set of parameters in the aggregation function of the multilayer graph, leveraging a multilayer version of the Laplacian-regularization formulation of GSSL. Tsuda et al. (2005) propose a method for protein classification using multiple protein networks. The multilayer graph is aggregated via a weighted linear combination, whose weights are learned in a variational min-max fashion aimed at minimizing the worst-case graph consistency function. The resulting method performs well in the presence of noisy or irrelevant layers. Similarly, Argyriou et al. (2005) compute an optimal linear combination of Laplacian kernels, which solves an extended regularization problem on the multilayer graph, enforcing a joint minimization over both the data and the set of graph kernels. Then, Zhou & Burges (2007) show that the resulting convex combination of graph Laplacians generalizes the normalized cut function to multilayer networks. An alternative formulation is proposed in (Nie et al., 2016) where the optimal weights in the weighted linear combination of layers' Laplacian are defined implicitly via a dual Lagrangian formulation. The resulting method is a parameter-free method for optimal layer weights. Finally, Karasuyama & Mamitsuka (2013) suggest an approach to efficiently linearly aggregate multiple graphs under the Laplacian regularization framework, by performing a form of alternate optimization via label propagation combined with sparse integration. Unlike the approach we propose here, all these methods are based on linear aggregation functions (convex combinations) and the optimal weights are model-based, rather than data-driven, aiming at optimizing some form of worst-case setting.

Nonlinear layer aggregation functions such as max, min, and their generalization, provide additional modeling power. Using a Log-Euclidean matrix function formulation of the generalized power mean of graph Laplacians, Mercado et al. (2019) propose a regularizer based on a one-parameter family of matrix means that includes the arithmetic, geometric and harmonic means as particular cases. This approach is revised and improved in (Bergermann et al., 2021), based on diffuse interface methods and fast matrix-vector products. While able to reach competitive performance, this approach requires an extensive exploration of the parameter defining the mean, which can be computationally prohibitive. Entrywise minimum and maximum aggregation functions are used in the multilink model proposed in (Mondragon et al., 2018; Bianconi, 2018; Ghorbanchian et al., 2022).

Deviating from the Laplacian regularization formulation,

Eswaran et al. (2017) propose a method based on fast belief propagation on heterogeneous graphs, with nodes of different types, while Gujral & Papalexakis (2018) design a parameter-free algorithm based on tensor factorization, which aims at finding both overlapping and non-overlapping communities.

While very popular in the single-layer setting, only a few extensions of geometric deep learning and graph neural networks to the multilayer setting are available so far and are mostly designed for the case of multilayer graphs having intra-layers connections, i.e. edges connecting different nodes from one layer to another. Among the available ones, (Ghorbani et al., 2019) is based on an extension of the graph convolutional filter by Welling & Kipf (2016), while (Grassia et al., 2021) proposes a graph neural network whose graph filter is a parametric aggregated Laplacian, parametrized in terms of an MLP.

3. Learning the Most Relevant Layers

Problem Set-up

We consider a multilayer graph, specifically a multiplex (alternatively known as multicolor, or multiview graph), defined as $\boldsymbol{G} = \{G^{(1)}, \dots, G^{(K)}\}$, with K layers, each being a weighted undirected graph $G^{(k)} = (V, E^{(k)}, w_k)$ with $V = \{1, \dots, N\}$, $E^{(k)} \subseteq V \times V$, and $w_k : E_k \to \mathbb{R}_+$. To each layer corresponds a weighted adjacency matrix $A^{(k)}$, whose entries $A^{(k)}_{ij} = w_k(ij) > 0$ represent the strength of the tie between i and j, if $ij \in E$, and $A^{(k)}_{ij} = 0$ if $ij \notin E$.

Using the terminology proposed in (Magnani et al., 2021), we assume $\boldsymbol{G}$ consists of a set $C = \{C_1, \dots, C_m\}$ of communities (or labels) that is total (i.e., every node belongs to at least one $C_j \in C$), node-disjoint (i.e., no node belongs to more than one cluster), and pillar (i.e., each node belongs to the same community across the layers). Further, we assume that for each $C_j \in C$ we are given a set of input known labels $O_j \subseteq V$ which are one-hot encoded into the matrix $Y \in \mathbb{R}^{n \times m}$, with $Y_{ij} = 1$ if $i \in C_j$, and $Y_{ij} = 0$ otherwise.

The goal is to learn the unknown labels. In our setting, we assume no node feature is available. In other words, we focus on the setting in which one has access only to topological information about the graph structure, and has some input knowledge about the community assignment of some nodes. This is a common setting in e.g. network and social science applications (Magnani et al., 2021).

Generalized Mean Adjacency Model

In order to learn a classifier that effectively takes into account the multilayer graph structure, we design a nonlinear aggregation strategy that optimally learns the aggregation parameters and computes a classifier based on a multilayer Laplacian-regularization model. To this end, we first briefly review the standard Laplacian-regularization model for single-layer graphs.

If only one layer is available, i.e. if we are dealing with the standard graph case, a successful approach to impose local and global consistency with the available input labels and with the graph structure is to minimize the following Laplacian-regularized GSSL loss function

$$\varphi(X) := \|X - Y\|_F^2 + \tfrac{\lambda}{2}\mathrm{Tr}(X^\top L X) \qquad (1)$$

over all $X \in \mathbb{R}^{n \times m}$. Here $L = D - A$ is the Laplacian matrix of the single-layer graph at hand, with $D = \mathrm{diag}(A\mathbf{1})$ the diagonal matrix of the (weighted) degrees. Simple linear algebra passages show that the obtained solution $X^* = \mathrm{argmin}\, \varphi(X)$ is entrywise positive, thus the entries X^*_{ij} can be interpreted as a classifier that provides a score quantifying the likelihood that node i belongs to community C_j. Hence, we assign to each node i the label C_{k^*}, with $k^* = \mathrm{argmax}_j\, X^*_{ij}$. Note that, if $y^{(k)}$ is the k-th column of Y, with one-hot information about the input labels in O_k, then one can equivalently write $\varphi(X) = \sum_{k=1}^m \varphi_k(x^{(k)})$, where $x^{(k)}$ are the columns of X and

$$\varphi_k(x) = \sum_{i=1}^N |x_i - y_i^{(k)}|^2 + \frac{\lambda}{2} \sum_{i,j=1}^N A_{ij}\,(x_i - x_j)^2 \ . \qquad (2)$$

Clearly, as the φ_k are independent of each other, in this single-graph setting minimizing φ is equivalent to minimizing each φ_k individually.

When we are given K layers, imposing smoothness with respect to the edge structure is more challenging. As the communities are assumed to be consistent across the layers, a standard approach is to tackle the problem after layer aggregation. If the aggregating function is linear, this boils down to choosing a set of weights $\boldsymbol{\beta}_k > 0$ with $\sum_k \boldsymbol{\beta}_k = 1$ and replace A in (1) or (2) with $A^{lin} = \sum_k \boldsymbol{\beta}_k A^{(k)}$. This approach has been widely explored and is considered for example in (Tsuda et al., 2005; Argyriou et al., 2005; Ye & Akoglu, 2018). As the cost function in (1) is quadratic, this is equivalent to considering the classifier $X^* = \sum_k \boldsymbol{\beta}_k X^*_k$, with X^*_k solution to (1) for $A = A^{(k)}$. Another possibility is to consider "nonlinear aggregations" (Mondragon et al., 2018; Mercado et al., 2019; Bergermann et al., 2021; Ghorbanchian et al., 2022). For example, using the concept of multilinks (Bianconi, 2018), Mondragon et al. (2018) replace the multilayer network with a single-layer graph with adjacency matrix with entries $A^{max}_{ij} = \max_k A^{(k)}_{ij}$ or $A^{min}_{ij} = \min_k A^{(k)}_{ij}$. The maximum-based model corresponds to assuming the edge ij exists in the aggregated graph if at least one edge between i and j is present in one of the layers. Similarly, the minimum-based one corresponds to the case where edges are kept in the aggregated graph if

they are present in all the layers. Clearly, these approaches are particularly effective when all the links in all the layers can be trusted, or when no layer can be trusted individually, respectively. However, in real-world applications, layers may contain complementary community information and some layers may be (partially) "noisy", i.e. they may contain limited or no information about the communities at all (Choobdar et al., 2019; Mercado et al., 2019).

The use of the parameters $\boldsymbol{\beta}_k$ in the linear model allows us to give different weights to the layers, if knowledge about their community information content is available. However, making an informed a-priori assessment on the presence of noise or on the different types of community structure across the layers can be very complicated (Choobdar et al., 2019). In order to overcome this modeling limitation, we design a nonlinear aggregation strategy that includes linear-, maximum-, and minimum-based aggregation strategies as special cases and learns optimal aggregation parameters from the available input labels.

Both the linear combination A^{lin} and the minimum, maximum matrices A^{min}, A^{max} can be seen as particular cases of more general nonlinear aggregations based on the *generalized mean adjacency matrix* $A(\alpha, \boldsymbol{\beta})$, entrywise defined as

$$A(\alpha, \boldsymbol{\beta})_{ij} = \left(\sum_{k=1}^{K} \boldsymbol{\beta}_k \left(A_{ij}^{(k)} \right)^{\alpha} \right)^{1/\alpha}, \tag{3}$$

where $\sum_k \boldsymbol{\beta}_k = 1$, $\boldsymbol{\beta}_k > 0$ as above, and $\alpha \in \mathbb{R}$. In fact, $A^{lin} = A(1, \boldsymbol{\beta})$, and $A^{min} = \lim_{\alpha \to -\infty} A(\alpha, \boldsymbol{\beta})$, $A^{max} = \lim_{\alpha \to +\infty} A(\alpha, \boldsymbol{\beta})$. As illustrated in Table 1, the nonlinearity introduced by the parameter α allows us model a broad class of aggregation functions through (3), including the maximum, the minimum, the harmonic, and the geometric means. Hence, using the generalized mean in (3) enables us to properly tune between all the different aggregation methods in Table 1. This is particularly useful as linear aggregations alone may be inappropriate for a variety of multilayer network topologies. For example, consider the case of a multilayer graphs with $K \gg 1$ layers, such that certain relevant edges appear only in one layer, which itself does not contain edges that are instead present in all the remaining layers. In this case, we need to consider the union (thus the maximum mean) of the edges across the layers, while their linear aggregation would most likely fail to capture the whole edge information. This is somehow shown by the "complementary" setting in our synthetic experiments in §5.1.

Note that, despite a similar terminology, $A(\alpha, \boldsymbol{\beta})$ is an elementwise function and thus is very different from the matrix function generalized mean considered in (Mercado et al., 2019).

Letting $D(\alpha, \boldsymbol{\beta}) = \mathrm{diag}(A(\alpha, \boldsymbol{\beta})\mathbf{1})$ be the degree matrix of the generalized mean adjacency matrix, and $L(\alpha, \boldsymbol{\beta}) = D(\alpha, \boldsymbol{\beta}) - A(\alpha, \boldsymbol{\beta})$ its Laplacian matrix, we extend (1) to the multilayer setting by considering the following

$$\varphi(X, Y; \alpha, \boldsymbol{\beta}, \lambda) = \|X - Y\|_F^2 + \tfrac{\lambda}{2} \mathrm{Tr}(X^\top L(\alpha, \boldsymbol{\beta}) X)$$

and the corresponding class-wise function $\varphi_k(x, y; \alpha, \boldsymbol{\beta}, \lambda)$, obtained by replacing A with $A(\alpha, \boldsymbol{\beta})$ in (2). In order to learn the parameters $\boldsymbol{\theta} := (\alpha, \boldsymbol{\beta}, \lambda)$, we split the available input labels into training and test sets, with corresponding one-hot matrices Y^{tr} and Y^{te}, and consider the bilevel optimization model

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & H(Y^{te}, X_{Y^{tr};\boldsymbol{\theta}}) \\ \text{s.t.} \quad & X_{Y^{tr};\boldsymbol{\theta}} = \mathrm{argmin}_X \, \varphi(X, Y^{tr}; \boldsymbol{\theta}) \\ & \boldsymbol{\theta} = (\alpha, \boldsymbol{\beta}, \lambda), \ \alpha \in \mathbb{R}, \ \boldsymbol{\beta} \geq 0, \ \textstyle\sum_k \boldsymbol{\beta}_k = 1, \ \lambda \in \mathbb{R} \end{aligned} \tag{4}$$

where H is the multiclass cross-entropy loss

$$H(Y, X) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} Y_{ij} \log \left(\frac{X_{ij}}{\sum_{j=1}^{N} X_{ij}} \right).$$

The resulting embedding X^* for the learned parameters is then used to classify the unlabeled nodes in the usual way.

Note that, unlike the single-layer case, using (5) rather than (4) in this setting may yield different results. In particular, if different layers carry information about different communities, using a one-vs-all cross-entropy model may be more effective. Thus, as an alternative to (4), we consider

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & h(y^{te}, x_{y^{tr};\boldsymbol{\theta}}) \\ \text{s.t.} \quad & x_{y^{tr};\boldsymbol{\theta}} = \mathrm{argmin}_x \, \varphi_k(x, y^{tr}; \boldsymbol{\theta}) \\ & \boldsymbol{\theta} = (\alpha, \boldsymbol{\beta}, \lambda), \ \alpha \in \mathbb{R}, \ \boldsymbol{\beta} \geq 0, \ \textstyle\sum_k \boldsymbol{\beta}_k = 1, \ \lambda \in \mathbb{R} \end{aligned} \tag{5}$$

which we solve for each community k, individually, using the binomial cross-entropy loss

$$h(y, x) = -\frac{1}{N} \sum_{i=1}^{N} \left(y_i \log(x_i) + (1 - y_i) \log(1 - x_i) \right).$$

4. Optimization with Inexact Gradient Computations

In order to compute the generalized mean-based classifier, we use a gradient-free optimization algorithm combined with a form of parametric Label Propagation, as detailed in §4.2. In particular, we consider a Frank-Wolfe (or conditional gradient) method (Frank & Wolfe, 1956; Jaggi, 2013; Bomze et al., 2021) with inexact gradient and a tailored line search. The method is described and analyzed below, limiting our attention to (4) for the sake of simplicity. Everything transfers straightforwardly to (5).

Table 1. Entries of the generalized mean adjacency matrix $A(\boldsymbol{\theta})$, for particular choices of the parameters $(\alpha, \boldsymbol{\beta}) \in \mathbb{R}^{K+1}$.

$\alpha \to -\infty$ Minimum (MIN)	$\alpha = -1, \boldsymbol{\beta}_k = 1/K$ Harmonic (HARM)	$\alpha \to 0, \boldsymbol{\beta}_k = 1/K$ Geometric (GEO)	$\alpha = 1, \boldsymbol{\beta}_k = 1/K$ Arithmetic (ARIT)	$\alpha \to +\infty$ Maximum (MAX)
$\min_{k=1,\dots,K} A_{ij}^{(k)}$	$\left(\frac{1}{K}\sum_{k=1}^{K}\frac{1}{A_{ij}^{(k)}}\right)^{-1}$	$\left(\prod_{k=1}^{K} A_{ij}^{(k)}\right)^{1/K}$	$\frac{1}{K}\sum_{k=1}^{K} A_{ij}^{(k)}$	$\max_{k=1,\dots,K} A_{ij}^{(k)}$

Algorithm 1 Frank-Wolfe algorithm with inexact gradient

1: **Given** $\theta_0 \in S$
2: **For** $n = 0, 1, \dots$
3: Compute $\widetilde{\nabla} f(\theta_n)$ as an estimate of $\nabla f(\theta_n)$
4: Compute $\hat{\theta}_n \in \operatorname{argmin}_{\theta \in S} \widetilde{\nabla} f(\theta_n)^\top (\theta - \theta_n)$ and set $d_n = \hat{\theta}_n - \theta_n$
5: Compute a stepsize $\eta_n \in (0, 1]$ by a line search
6: Set $\theta_{n+1} = \theta_n + \eta_n d_n$
7: **End for**

Note that, fixing the parameters $\boldsymbol{\theta} = (\alpha, \boldsymbol{\beta}, \lambda)$, the inner problem $\min_X \varphi(X, Y; \boldsymbol{\theta})$ can be solved explicitly. In fact, a direct computation shows that $\nabla_X \varphi(X, Y; \boldsymbol{\theta}) = 2\{(X - Y) + \lambda L(\alpha, \boldsymbol{\beta})\}X$. Thus,

$$\operatorname{argmin}_X \varphi(X, Y; \boldsymbol{\theta}) = \big(I + \lambda L(\alpha, \boldsymbol{\beta})\big)^{-1} Y\,. \quad (6)$$

Using (6), we can rewrite (4) by replacing the optimality constraint at the inner level with its explicit solution. Moreover, as the generalized mean converges fast to maximum and minimum for $\alpha \to \pm\infty$, we limit α within an interval $\alpha \in [-a, a]$, for a large enough $a > 0$. Similarly, we restrict our study to $\lambda \in [l_0, l_1]$. Altogether, we reformulate (4) as

$$\min_{\boldsymbol{\theta} \in S} \; f(\boldsymbol{\theta})\,, \quad (7)$$

where $f(\boldsymbol{\theta}) := H(Y^{te}, (I + \lambda L(\boldsymbol{\theta}))^{-1} Y^{tr})$ and, for $a > 0$, $S = \{(\alpha, \boldsymbol{\beta}, \lambda) \in \mathbb{R}^{K+2} : \alpha \in [-a, a], \boldsymbol{\beta}_k > 0, \sum_k \boldsymbol{\beta}_k = 1, \lambda \in [l_0, l_1]\}$.

As mentioned above, we use a Frank-Wolfe based method to solve (7). The rationale behind the algorithm is to compute, at every iteration n, a direction d_n minimizing a linear approximation of f around the current point $\boldsymbol{\theta}_n$. Then we obtain the next point $\boldsymbol{\theta}_{n+1}$ by moving along d_n with a stepsize η_n, chosen by a proper line search. Although f is a smooth real-valued function, the computation of its gradient ∇f can be extremely expensive in practice. To overcome this issue, instead of ∇f, in the algorithm we use an estimate $\widetilde{\nabla} f$. The resulting method is presented in Algorithm 1.

Note that the linear problem at line 1 of Algorithm 1 is particularly simple due to the box-plus-simplex form of the constraint set S, and it can be solved separately in the variables α, $\boldsymbol{\beta}$, and λ. In fact, for the variable α, we aim at minimizing a linear function over the box $[-a, a]$, which implies $\hat{\alpha}_n = -a$ if $\widetilde{\nabla}_\alpha f(\boldsymbol{\theta}_n) > 0$, and $\hat{\alpha}_n = a$ otherwise. Similarly, for the variable λ, we set $\hat{\lambda}_n = l_0$ if $\widetilde{\nabla}_\lambda f(\boldsymbol{\theta}_n) > 0$, and $\hat{\lambda}_n = l_1$ otherwise. For the variables $\boldsymbol{\beta} \in \mathbb{R}^K$, we have to minimize a linear function over the unit simplex, which yields $\hat{\boldsymbol{\beta}}_n = e_{\hat{\jmath}}$, where $\hat{\jmath} = \operatorname{argmin}_{j=1,\dots,K} [(\widetilde{\nabla}_{\boldsymbol{\beta}} f(\boldsymbol{\theta}_n)]_j$ and $e_{\hat{\jmath}}$ is the $\hat{\jmath}$-th vector of the canonical basis of $\mathbb{R}^K$.

4.1. Convergence Analysis

To analyze the convergence of Algorithm 1, we first introduce some useful notation. Let $g_n = -\nabla f(\boldsymbol{\theta}_n)^\top d_n$, $\widetilde{g}_n = -\widetilde{\nabla} f(\boldsymbol{\theta}_n)^\top d_n$ and $g_n^{FW} = -\nabla f(\boldsymbol{\theta}_n)^\top d_n^{FW}$, where $d_n^{FW} \in \operatorname{argmin}_{\boldsymbol{\theta} \in S}\{\nabla f(\boldsymbol{\theta}_n)^\top (\boldsymbol{\theta} - \boldsymbol{\theta}_n)\} - \boldsymbol{\theta}_n$ is the direction obtained by the Frank-Wolfe algorithm with exact gradient. Inspired by (Freund & Grigas, 2016), we assume that the estimate $\widetilde{\nabla} f$ satisfies the following condition.

Assumption 4.1. For every n, there exists $\epsilon_n \geq 0$ such that

$$|(\nabla f(\boldsymbol{\theta}_n) - \widetilde{\nabla} f(\boldsymbol{\theta}_n))^\top (\boldsymbol{\theta} - \boldsymbol{\theta}_n)| \leq \epsilon_n \quad \forall\, \boldsymbol{\theta} \in S. \quad (8)$$

Since S is a convex set, a point $\boldsymbol{\theta}^* \in S$ is said to be stationary for (7) when $\nabla f(\boldsymbol{\theta}^*)^\top (\boldsymbol{\theta} - \boldsymbol{\theta}^*) \geq 0$ for all $\boldsymbol{\theta} \in S$. Then, g_n^{FW} is an optimality measure, i.e. $g_n^{FW} = 0$ if and only if $\boldsymbol{\theta}_n \in S$ is a stationary point. Now we show that, when Assumption 4.1 is satisfied with a sufficiently small ϵ_n and the stepsize η_n is generated with a suitable line search, Algorithm 1 obtains a stationary point at a sublinear rate on non-convex objectives with a Lipschitz continuous gradient. The constant in the convergence rate depends on the quality of the gradient estimate (the more precise the estimate, the smaller the constant). The proof can be found in Appendix A.

Theorem 4.2. *Let ∇f be Lipschitz continuous with constant M, and let S be compact with finite diameter Δ. Let $\{\boldsymbol{\theta}_n\}$ be a sequence generated by Algorithm 1, where $\widetilde{\nabla} f$ satisfies Assumption 4.1 with*

$$\epsilon_n \leq \frac{\sigma}{1 + \sigma}\, \widetilde{g}_n, \quad 0 \leq \sigma < \frac{1}{3}, \quad (9)$$

and the step size η_n satisfies

$$\eta_n \geq \bar{\eta}_n = \min\left(1, \frac{\widetilde{g}_n}{M\|d_n\|^2}\right), \quad (10)$$

$$f(\boldsymbol{\theta}_n) - f(\boldsymbol{\theta}_n + \eta_n d_n) \geq \rho \bar{\eta}_n \widetilde{g}_n, \quad (11)$$

with some fixed $\rho > 0$. Then,

$$g_n^* \leq \max\left(\sqrt{\frac{\Delta^2 M(f(\boldsymbol{\theta}_0) - f^*)}{n\rho(1-\sigma)^2}}, \frac{2(f(\boldsymbol{\theta}_0) - f^*)}{n(1-3\sigma)}\right), \tag{12}$$

where $g_n^* = \min_{0\leq i\leq n-1} g_i^{FW}$ and $f^* = \min_{\boldsymbol{\theta}\in S} f(\boldsymbol{\theta})$.

Note that, in our setting, $\Delta \leq 2a + \sqrt{2}$. Condition (9) can be easily satisfied by a proper calculation of the gradient estimate $\widetilde{\nabla} f$ (see §4.2). Conditions (10)–(11) can be satisfied with suitable line searches/stepsize rules (see, e.g., (Bomze et al., 2020; 2021; Rinaldi & Zeffiro, 2022)). In particular, Lemma 4.3 below shows that this is the case for the modified Armijo line search rule which sets

$$\eta_n = \delta^j, \tag{13}$$

where j is the smallest non-negative integer such that

$$f(\boldsymbol{\theta}_n) - f(\boldsymbol{\theta}_n + \eta_n d_n) \geq \gamma\eta_n \widetilde{g}_n, \tag{14}$$

with $\gamma \in (0, 1/2)$ and $\delta \in (0, 1)$ being two fixed parameters. The proof can be found in Appendix B.

Lemma 4.3. *Let Assumption 4.1 hold with*

$$\epsilon_n \leq \frac{\sigma}{1+\sigma}\widetilde{g}_n, \quad 0 \leq \sigma < \frac{1}{2}. \tag{15}$$

At iteration n, if η_n is determined by the Armijo line search described in (13)–(14), then

$$\eta_n \geq \min\{1, 2\delta(1-\gamma-\sigma)\}\bar{\eta}_n, \tag{16}$$

with $\bar{\eta}_n$ being defined as in (10).

Note that in the proof of Lemma 4.3 we prove

$$\eta_n \geq \min\left(1, c\frac{\widetilde{g}_n}{M\|d_n\|^2}\right) \text{ for some } c > 0$$

for the Armijo line search. When $c \geq 1$ then $\bar{\eta}_n$ is of course a lower bound for the step size η_n, and when $c < 1$ we can still recover (10) by considering $\widetilde{M} = M/c$ instead of M as Lipschitz constant. The complexity analysis of the method can be obtained straightforwardly from Theorem 4.2. Details are reported in Appendix C.

4.2. Implementation Details

In Algorithm 1, we approximate the gradient with the finite difference method:

$$\widetilde{\nabla} f(\boldsymbol{\theta}_n) = \sum_{i=1}^{K+2} \frac{f(\boldsymbol{\theta}_n + h_n e_i) - f(\boldsymbol{\theta}_n)}{h_n} e_i, \tag{17}$$

where h_n is a suitably chosen positive parameter. As shown in (Berahas et al., 2022), this approach gives good gradient approximations in practice.

From (Sahu et al., 2019), we have that Equation (8) is satisfied when using the finite difference approach with $\epsilon_n = \frac{M\Delta(2+K)}{2}h_n$. We notice that condition (9) can in turn be satisfied at each iteration k by suitably choosing h_n in the finite difference approximation, e.g., $h_n \leq \xi\tau$, with $\xi = \frac{2\sigma}{(1+\sigma)M\Delta(2+K)}$ and τ stopping condition tolerance.

In our experiments, we start with $h_0 = 10^{-4}$ and set $h_n = \frac{h_{n-1}}{2}$ for $n = 1, 2, \ldots$. Furthermore, we stop the algorithm when $\widetilde{g}_n = -\widetilde{\nabla} f(\boldsymbol{\theta}_n)^\top d_n \leq \tau$, with $\tau = 10^{-4}$.

In order to compute $f(\boldsymbol{\theta})$ for a given set of parameters $\boldsymbol{\theta} = (\alpha, \boldsymbol{\beta}, \lambda)$, we run a form of modified parametric Label Propagation algorithm:

$$X^{(r+1)} = \lambda A(\alpha, \boldsymbol{\beta})(I + \lambda D(\alpha, \boldsymbol{\beta}))^{-1} X^{(r)} + \\ + (I + \lambda D(\alpha, \boldsymbol{\beta}))^{-1} Y\,,$$

which propagates the input labels in Y and converges to the solution of the linear system (6). In fact, as $\sum_j A(\alpha, \boldsymbol{\beta})_{ij} = D(\alpha, \boldsymbol{\beta})_{ii}$ for all i, a direct application of the first Gershgorin circle theorem (Varga, 2010) to the matrix $A(\alpha, \boldsymbol{\beta})(I + \lambda D(\alpha, \boldsymbol{\beta}))^{-1}$ implies that the spectral radius of $A(\alpha, \boldsymbol{\beta})(I + \lambda D(\alpha, \boldsymbol{\beta}))^{-1}$ is smaller than one and thus $X^{(r)} \to \operatorname{argmin}_X \varphi(X, Y; \boldsymbol{\theta})$, as $r \to \infty$.

We apply the multistart version of Frank-Wolfe (Martí et al., 2013), where the algorithm is applied with different initial points, and we choose the best solution according to the value of the optimized function f. In particular, we start from 10 random points $\boldsymbol{\theta}_0$, among which we include the particular choices $\boldsymbol{\theta}_0 = (1, 1/K, \ldots, 1/K, 1)$ and $\boldsymbol{\theta}_0 = (-1, 1/K, \ldots, 1/K, 1)$, which correspond to the arithmetic and the harmonic means. In all the experiments, we restricted the study of the parameters α and λ as follows: $\alpha \in [-20, 20]$ and $\lambda \in [0.1, 10]$. The latter boils down to the standard $[0.1, 0.9]$ search interval for the variable $\lambda/(1+\lambda)$, usually employed in label propagation algorithms. For the methods obtained with a fixed choice of generalized mean aggregation function (see Table 1 for details), we fixed $\lambda = 1$.

5. Numerical Results

We perform tests on different synthetic and real-world multilayer networks. For each data set, we are given an input set of known labels Y, which we split into a train Y^{tr} set with 80% of the available labels, and a test Y^{te} set, formed by the remaining 20%. Since the resulting optimal parameters can change depending on the selection of the training and test sets, we initially randomly split the input labeled nodes into 5 sets of equal size and then cyclically assign one of the sets to Y^{te} and the remaining ones to Y^{tr}. For each of these choices, we run Algorithm 1 with 10 starting points as discussed in §4.2, and we eventually choose the

parameters that yield the lowest value of the loss function $f(\boldsymbol{\theta})$ over the test set among the five runs. Once the optimal weights are computed, we run standard Label Propagation on the resulting aggregated graph and we asses the accuracy performance on the held-out test set, which is comprised of all the initially non-labeled points.

We implement both the multiclass (MULTI) and the binomial (BINOM) versions of our method, corresponding to the bilevel optimization problems in (4) and (5), respectively. Our `Python` implementation is available at `https://github.com/saraventurini/Learning-the-right-layers-on-multilayer-graphs`.

We considered both synthetic and real-world networks, performing extensive experiments to compare the proposed approach, which learns the parameters of the generalized mean from the available input data, against standard Label propagation on each single layer, as well as the methods corresponding to the proposed generalized mean aggregation function for some special choices of the parameters (those illustrated in Table 1), and four multilayer graph semi-supervised learning baselines:

- **SGMI**: Sparse Multiple Graph Integration (Karasuyama & Mamitsuka, 2013), based on label propagation by sparse integration, with parameters $\lambda_1 = 1, \lambda_2 = 10^{-3}$;
- **AGML**: Auto-weighted Multiple Graph Learning (Nie et al., 2016), which is a parameter-free method for optimal graph layer weights;
- **SMACD**: Semi-supervised Multi-Aspect Community Detection (Gujral & Papalexakis, 2018), which is a tensor factorization method for semi-supervised learning;
- **GMM**: Generalized Matrix Means, which is a Laplacian-regularization approach based on the Log-Euclidean matrix function formulation of the power mean Laplacian, with parameter $p = -1$ (Mercado et al., 2019);

Notice that SGMI and GMM need a parameter choice, which we have made following the indications in the corresponding papers, while AGML and SMACD, as well as the proposed methods MULTI and BINOM, are parameter-free. We also tested against the two multilayer graph neural networks discussed in §2, which however performed poorly, probably due to the absence of features in our test settings.

5.1. Synthetic Datasets

We created synthetic datasets with 3 communities of 400 nodes each, and 3 layers. In particular, for each layer, we generated 3 isotropic Gaussian blobs of points $p_i \in \mathbb{R}^5$, with a variable standard deviation. The adjacency matrix of the network is then formed by means of a symmetrized k-NN graph with $k = 5$, weighted with the Euclidean kernel $\exp(-\|p_i - p_j\| + \min_{ij} \|p_i - p_j\|)$. We considered three settings (illustrated in Figure 5):

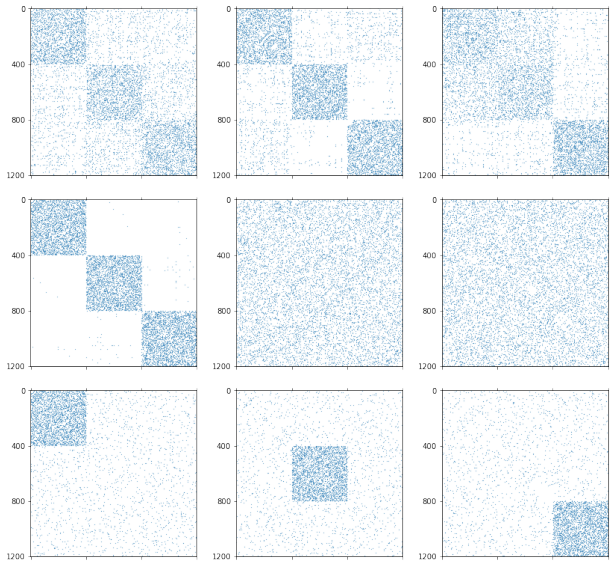

Figure 1. Synthetic datasets settings. (top) informative case, (middle) noisy case, (bottom) complementary case.

- *Informative case*: layers are formed by 3 isotropic Gaussian blobs and all show the same community structure;
- *Noisy case*: one layer is informative and the other two are noise. The noisy layers are generated by a random reshuffling of the informative ones.
- *Complementary case*: each layer carries information concerning only one cluster, while is noisy for the remaining ones. The noise layers are sparser here than in the previous setting (we shuffle k-NN layers with $k = 1$).

The informative isotropic Gaussian blobs have standard deviation $std \in \{5, 6, 7, 8\}$ for the informative case, as this is the easiest setting, while we test for $std \in \{2, 3, 4, 5\}$ in the other two settings, as these are more challenging. The percentage of input labels is 20% of the overall number of nodes in each of the communities.

Table 3 reports the average accuracy and standard deviation score across 5 network samples, as compared to the accuracy of the individual layers (computed ignoring the other layers), reported in the first three columns, and those achieved with fixed a-priori choices of the parameters (as in Table 1). The proposed BINOM and MULTI perform well across all settings, most of the time outperforming each individual layer and the considered baselines. In particular, in all the settings, MIN, GEO and SMACD perform poorly. AGML works well mostly in the informative setting. ARIT, HARM, and MAX show good performances in the informative and complementary cases, but not in the noisy one. SGMI achieves high accuracy only in the noisy case. GMM performs well only in the informative and noisy settings.

While the best performance is sometimes achieved by some particular aggregation function (such as MAX or HARM), all the baselines are setting-specific and have poor perfor-

Table 2. Example of learned parameters by BINOM (B) and MULTI (M) on the synthetic datasets of Table 3.

		INFO					NOISY					COMPL				
	k	β_1	β_2	β_3	α	λ	β_1	β_2	β_3	α	λ	β_1	β_2	β_3	α	λ
B	1	0.33	0.32	0.35	-2.8	1.12	1.0	0.0	0.0	20.0	10.0	0.83	0.12	0.05	10.27	8.62
	2	0.32	0.46	0.22	-1.89	6.36	1.0	0.0	0.0	20.0	10.0	0.13	0.50	0.37	10.35	3.62
	3	0.23	0.36	0.41	-3.54	1.02	1.0	0.0	0.0	20.0	10.0	0.11	0.24	0.65	10.95	4.69
M	-	0.31	0.34	0.35	-3.65	1.74	1.0	0.0	0.0	5.03	0.97	0.30	0.39	0.31	-6.96	0.67

mance in certain settings, e.g. in the presence of noise. When measured across all settings, BINOM and MULTI perform best. This is highlighted by the Average Performance Ratio (APR) score values, reported in the last line of Table 3, which are quantified as follows: denoting the accuracy of algorithm a on dataset d as $\mathcal{A}_{a,d}$, let the performance ratio be $r_{a,d} = \mathcal{A}_{a,d}/\max\{\mathcal{A}_{a,d} \text{ over all } a\}$. The APR of each algorithm is then obtained by averaging $r_{a,d}$ over all the datasets d. For any algorithm, the closer the average performance ratio is to 1, the better the overall performance. We further compare the performances using the average rank (AR), computed by assigning 1 to the best-performing method, 2 to the second best, and so on, for each dataset. We highlight that the APR is a more informative metric as it also takes into account the value of the accuracy of the individual tests, while AR does not.

Moreover, by inspecting the learned weights, the proposed methods allow us to make an assessment of the structure of the multilayer network and the presence of noisy or less informative layers. This is illustrated in Table 2, where we show an example of learned weights resulting from Algorithm 1 Notice that, looking at the $\boldsymbol{\beta}$ parameter: in the informative case, the weight is equally distributed among the layers; in the noisy case, the methods consider just the informative layer disregarding completely the noisy layers; in the complementary case, MULTI distributes the weight equally among the layers, while BINOM gives a higher weight to the community correspondent to each layer.

In Appendix C we report a CPU-time comparison with the methods in Table 3 as the number of nodes increases, which highlights the efficiency of our framework.

5.2. Real World Datasets

We consider nine real-world datasets frequently used to assess performance of multilayer graph clustering (Mercado et al., 2019; Magnani et al., 2021; Venturini et al., 2022):

- *3sources*: news articles covered by news sources BBC, Reuters, and Guardian (169 nodes, 6 communities, 3 layers) (Greene & Cunningham, 2009; Liu et al., 2013);
- *BBC*: BBC news articles (685 nodes, 5 communities, 4 layers) (Greene & Cunningham, 2005);
- *BBCSport*: BBC Sport articles (544 nodes, 5 communities, 2 layers) (Greene & Cunningham, 2009);
- *Wikipedia*: Wikipedia articles (693 nodes, 10 communities, 2 layers) (Rasiwasia et al., 2010);
- *UCI*: hand-written UCI digits dataset with six different sets of features (2000 nodes, 10 communities, 6 layers) (Dua & Efi, 2017; Liu et al., 2013);
- *cora*: citations dataset (2708 nodes, 7 communities, 2 layers) (McCallum et al., 2000);
- *citeseer*: citations dataset (3312 nodes, 6 communities, 2 layers) (Lu & Getoor, 2003);
- *dkpol*: five types of relationships between employees of a university department (490 nodes, 10 communities, 3 layers) (Hanteer et al., 2018);
- *aucs*: three types of online relations between Danish Members of the Parliament on Twitter (61 nodes, 9 communities, 5 layers) (Rossi & Magnani, 2015).

For each dataset, we assume either 1% or 10% of the labels are initially known, for each class. Tables 4 and 5 report the average accuracy ($\pm$ standard deviation) across 3 samples of the known labels. In addition to the original datasets, we show the performance when one additional noisy layer is added. In Appendix D we report additional results for different known label percentages and with the addition of two layers of noise. We do not compare the methods against the AGML baseline as that method is designed for graphs with communities of the same size. The results confirm the same behavior observed in the synthetic case, showing that BINOM and MULTI match or overcome the baselines in most cases, and are the best-performing methods overall, across all settings. We also emphasize that BINOM and MULTI are the only techniques that consistently match or outperform the single layers used in isolation, showing that our approach is able to effectively take advantage of the multilayer structure in all different settings. This is a particularly remarkable and desirable property, which is directly related to a recent data challenge (Choobdar et al., 2019).

In Appendix E we report a table with the different parameters learned by the methods. The numbers are averaged over three random samplings of the initially labeled nodes.

6. Conclusion

We proposed a parameter-free method for semi-supervised classification on multiplex networks that identifies relevant layers by learning a nonlinear aggregation function from the known labels. We formulate the model as a bilevel optimization problem which we solve using an inexact Frank-Wolfe algorithm combined with a parametric Label Propagation scheme. We provide a detailed convergence analysis of the method. Experimental results against single-layer approaches and a variety of baselines, on both synthetic and real-world datasets, demonstrate that the proposed method is able to identify relevant layers and thus obtain consistent and robust performance across different clustering settings, in particular when some layers are mostly just noise.

Table 3. Accuracy (mean ± standard deviation) over five random samples of synthetically generated multilayer graphs, for different levels of std in the isotropic Gaussian blobs forming the clusters.

	std	I	II	III	MIN	GEOM	ARIT	HARM	MAX	BINOM	MULTI	SGMI	AGML	SMACD	GMM
INFO	5	0.83±0.1	0.86±0.05	0.84±0.1	0.33±0.0	0.33±0.0	0.95±0.03	0.96±0.03	0.96±0.03	0.97±0.02	0.93±0.04	0.87±0.04	0.93±0.05	0.48±0.18	0.93±0.04
	6	0.77±0.11	0.8±0.05	0.79±0.11	0.33±0.0	0.33±0.0	0.9±0.05	0.92±0.04	0.92±0.05	0.91±0.06	0.87±0.07	0.8±0.06	0.86±0.09	0.47±0.16	0.89±0.05
	7	0.71±0.1	0.74±0.05	0.74±0.11	0.33±0.0	0.33±0.0	0.86±0.07	0.88±0.06	0.87±0.07	0.87±0.06	0.83±0.08	0.74±0.07	0.8±0.1	0.46±0.14	0.85±0.07
	8	0.66±0.1	0.69±0.04	0.7±0.11	0.33±0.0	0.33±0.0	0.81±0.08	0.83±0.08	0.82±0.09	0.8±0.09	0.78±0.08	0.69±0.07	0.74±0.1	0.44±0.14	0.8±0.08
NOISY	2	0.99±0.02	0.35±0.01	0.36±0.01	0.33±0.0	0.33±0.0	0.67±0.02	0.68±0.03	0.68±0.03	0.99±0.01	0.99±0.02	0.98±0.03	0.63±0.03	0.35±0.03	0.86±0.04
	3	0.95±0.05	0.35±0.0	0.36±0.01	0.33±0.0	0.33±0.0	0.66±0.04	0.66±0.04	0.66±0.04	0.96±0.04	0.95±0.05	0.94±0.07	0.6±0.06	0.36±0.04	0.82±0.07
	4	0.89±0.08	0.36±0.0	0.35±0.01	0.33±0.0	0.33±0.0	0.63±0.05	0.63±0.05	0.63±0.05	0.9±0.08	0.89±0.08	0.88±0.1	0.57±0.07	0.36±0.03	0.77±0.1
	5	0.83±0.1	0.35±0.01	0.36±0.01	0.33±0.0	0.33±0.0	0.58±0.06	0.59±0.06	0.58±0.06	0.85±0.09	0.83±0.1	0.8±0.12	0.53±0.07	0.35±0.04	0.7±0.11
COMPL	2	0.41±0.0	0.47±0.0	0.48±0.01	0.33±0.0	0.33±0.0	0.89±0.02	0.94±0.02	0.92±0.02	0.86±0.01	0.89±0.02	0.43±0.04	0.3±0.01	0.52±0.05	0.69±0.03
	3	0.41±0.01	0.47±0.01	0.47±0.01	0.33±0.0	0.33±0.0	0.86±0.04	0.9±0.05	0.89±0.05	0.85±0.02	0.87±0.04	0.44±0.04	0.3±0.02	0.44±0.08	0.67±0.04
	4	0.4±0.01	0.46±0.01	0.47±0.01	0.33±0.0	0.33±0.0	0.82±0.06	0.86±0.06	0.85±0.06	0.82±0.01	0.83±0.06	0.43±0.04	0.18±0.16	0.71±0.17	0.65±0.05
	5	0.4±0.01	0.46±0.02	0.47±0.01	0.33±0.0	0.33±0.0	0.79±0.07	0.82±0.08	0.81±0.07	0.81±0.07	0.8±0.07	0.43±0.04	0.18±0.17	0.46±0.13	0.63±0.06
APR		-	-	-	0.37	0.37	0.87	0.90	0.89	**0.98**	0.97	0.78	0.61	0.50	0.86
AR		-	-	-	10.0	10.0	5.0	2.9	3.5	**2.5**	3.8	6.1	8.6	8.7	5.1

Table 4. Accuracy (mean ± std) over three random samples of the 1% of input labels, on real-world datasets (+ one layer of noise).

	I	II	III	IV	V	VI		MIN	GEOM	ARIT	HARM	MAX	BINOM	MULTI	SGMI	SMACD	GMM
3sources	0.72	0.69	0.77	-	-	-		0.66±0.01	0.66±0.01	0.79±0.04	0.71±0.04	0.76±0.04	0.79±0.04	0.79±0.04	0.65±0.04	0.58±0.06	0.69±0.06
							(+1 noise)	0.35±0.01	0.35±0.01	0.73±0.06	0.55±0.05	0.69±0.07	0.73±0.06	0.73±0.06	0.52±0.22	0.65±0.08	0.62±0.02
BBC	0.77	0.76	0.72	0.75	-	-		0.35±0.01	0.35±0.01	0.83±0.01	0.81±0.01	0.82±0.01	0.83±0.01	0.83±0.01	0.64±0.03	0.59±0.07	0.68±0.03
							(+1 noise)	0.33±0.0	0.33±0.0	0.78±0.01	0.74±0.01	0.76±0.01	0.78±0.01	0.83±0.01	0.23±0.01	0.62±0.04	0.6±0.02
BBCSport	0.87	0.76	-	-	-	-		0.63±0.07	0.63±0.07	0.87±0.03	0.87±0.03	0.87±0.03	0.87±0.03	0.87±0.03	0.73±0.09	0.79±0.09	0.7±0.06
							(+1 noise)	0.36±0.0	0.36±0.0	0.73±0.03	0.71±0.02	0.73±0.03	0.73±0.03	0.73±0.03	0.24±0.02	0.79±0.04	0.54±0.04
Wikipedia	0.17	0.61	-	-	-	-		0.17±0.01	0.17±0.01	0.42±0.03	0.42±0.04	0.42±0.03	0.42±0.03	0.42±0.03	0.58±0.03	0.31±0.05	0.43±0.05
							(+1 noise)	0.15±0.0	0.15±0.0	0.34±0.01	0.34±0.01	0.34±0.01	0.34±0.01	0.33±0.02	0.15±0.0	0.26±0.07	0.33±0.03
UCI	0.79	0.72	0.9	0.43	0.91	0.69		0.1±0.0	0.1±0.0	0.85±0.01	0.79±0.02	0.82±0.02	0.85±0.01	0.81±0.02	0.56±0.14	0.35±0.11	0.58±0.02
							(+1 noise)	0.1±0.0	0.1±0.0	0.82±0.01	0.75±0.02	0.78±0.02	0.5±0.02	0.82±0.01	0.13±0.0	0.26±0.06	0.54±0.0
cora	0.58	0.49	-	-	-	-		0.3±0.01	0.3±0.01	0.56±0.04	0.56±0.04	0.56±0.04	0.57±0.03	0.56±0.03	0.55±0.05	0.37±0.07	0.45±0.03
							(+1 noise)	0.3±0.0	0.3±0.0	0.43±0.02	0.42±0.02	0.43±0.02	0.49±0.02	0.53±0.09	0.58±0.02	0.36±0.07	0.4±0.02
citeseer	0.41	0.5	-	-	-	-		0.23±0.01	0.23±0.01	0.53±0.04	0.54±0.03	0.54±0.04	0.55±0.03	0.54±0.03	0.4±0.01	0.3±0.08	0.38±0.03
							(+1 noise)	0.21±0.0	0.21±0.0	0.36±0.02	0.36±0.02	0.36±0.02	0.39±0.06	0.49±0.03	0.39±0.01	0.37±0.06	0.32±0.01
dkpol	0.29	0.16	0.67	-	-	-		0.15±0.0	0.15±0.0	0.67±0.03	0.66±0.03	0.67±0.03	0.67±0.03	0.67±0.03	0.19±0.04	0.24±0.06	0.57±0.02
							(+1 noise)	0.14±0.0	0.14±0.0	0.57±0.04	0.55±0.03	0.56±0.03	0.65±0.04	0.65±0.04	0.17±0.01	0.34±0.15	0.31±0.02
aucs	0.34	0.36	0.61	0.8	0.72	-		0.3±0.01	0.3±0.01	0.85±0.04	0.8±0.06	0.85±0.05	0.84±0.04	0.85±0.03	0.75±0.07	0.5±0.16	0.81±0.05
							(+1 noise)	0.27±0.0	0.27±0.0	0.81±0.04	0.37±0.03	0.65±0.03	0.81±0.04	0.81±0.04	0.76±0.09	0.42±0.03	0.77±0.09
APR								0.41	0.41	0.94	0.87	0.91	0.94	**0.97**	0.66	0.65	0.77
AR								9.1	9.1	3.0	5.1	3.9	2.8	**2.5**	6.5	6.8	6.1

Table 5. Accuracy (mean ± std) over three random samples of the 10% of input labels, on real-world datasets (+ one layer of noise).

	I	II	III	IV	V	VI		MIN	GEOM	ARIT	HARM	MAX	BINOM	MULTI	SGMI	SMACD	GMM
3sources	0.76	0.72	0.75	-	-	-		0.71±0.07	0.71±0.07	0.79±0.07	0.69±0.05	0.75±0.05	0.75±0.07	0.74±0.05	0.75±0.04	0.62±0.04	0.8±0.05
							(+1 noise)	0.34±0.01	0.34±0.01	0.74±0.1	0.54±0.06	0.66±0.09	0.76±0.11	0.73±0.06	0.58±0.27	0.6±0.08	0.76±0.04
BBC	0.83	0.83	0.79	0.83	-	-		0.38±0.01	0.38±0.01	0.91±0.01	0.89±0.01	0.9±0.01	0.88±0.01	0.86±0.04	0.76±0.02	0.69±0.03	0.87±0.01
							(+1 noise)	0.33±0.0	0.33±0.0	0.89±0.01	0.84±0.01	0.87±0.02	0.87±0.02	0.87±0.01	0.76±0.03	0.66±0.02	0.84±0.01
BBCSport	0.90	0.88	-	-	-	-		0.79±0.0	0.79±0.0	0.92±0.01	0.92±0.0	0.92±0.0	0.92±0.02	0.88±0.01	0.84±0.03	0.73±0.1	0.88±0.02
							(+1 noise)	0.36±0.0	0.36±0.0	0.86±0.02	0.82±0.01	0.85±0.02	0.91±0.01	0.87±0.0	0.83±0.04	0.6±0.07	0.77±0.02
Wikipedia	0.18	0.65	-	-	-	-		0.19±0.01	0.19±0.01	0.51±0.03	0.51±0.03	0.51±0.03	0.62±0.04	0.64±0.02	0.61±0.03	0.24±0.04	0.57±0.03
							(+1 noise)	0.15±0.0	0.15±0.0	0.42±0.01	0.42±0.01	0.42±0.01	0.57±0.06	0.62±0.03	0.59±0.02	0.25±0.04	0.47±0.02
UCI	0.92	0.81	0.96	0.57	0.97	0.82		0.11±0.0	0.11±0.0	0.95±0.01	0.88±0.01	0.92±0.01	0.97±0.01	0.96±0.01	0.94±0.01	0.33±0.05	0.93±0.01
							(+1 noise)	0.1±0.0	0.1±0.0	0.96±0.0	0.88±0.01	0.92±0.0	0.97±0.0	0.96±0.01	0.94±0.01	0.3±0.1	0.93±0.01
cora	0.73	0.62	-	-	-	-		0.34±0.01	0.34±0.01	0.69±0.01	0.69±0.01	0.69±0.01	0.74±0.03	0.76±0.01	0.72±0.01	0.34±0.08	0.69±0.01
							(+1 noise)	0.3±0.0	0.3±0.0	0.57±0.01	0.56±0.01	0.57±0.01	0.63±0.01	0.76±0.01	0.72±0.02	0.28±0.01	0.58±0.01
citeseer	0.51	0.62	-	-	-	-		0.29±0.01	0.29±0.01	0.65±0.01	0.65±0.01	0.65±0.01	0.66±0.01	0.65±0.01	0.51±0.01	0.36±0.1	0.58±0.01
							(+1 noise)	0.21±0.0	0.21±0.0	0.53±0.02	0.52±0.02	0.53±0.02	0.59±0.03	0.63±0.0	0.52±0.02	0.32±0.03	0.49±0.01
dkpol	0.32	0.16	0.69	-	-	-		0.16±0.0	0.16±0.0	0.73±0.06	0.67±0.09	0.69±0.07	0.62±0.06	0.76±0.01	0.31±0.03	0.26±0.09	0.63±0.04
							(+1 noise)	0.14±0.0	0.14±0.0	0.62±0.02	0.59±0.02	0.6±0.02	0.54±0.08	0.72±0.08	0.31±0.05	0.24±0.01	0.34±0.01
aucs	0.34	0.36	0.61	0.8	0.72	-		0.3±0.01	0.3±0.01	0.85±0.04	0.8±0.06	0.85±0.05	0.85±0.04	0.85±0.02	0.75±0.07	0.58±0.03	0.81±0.05
							(+1 noise)	0.27±0.0	0.27±0.0	0.81±0.04	0.37±0.03	0.65±0.03	0.81±0.04	0.81±0.04	0.76±0.09	0.56±0.07	0.77±0.09
APR								0.38	0.38	0.93	0.85	0.89	0.95	**0.98**	0.85	0.55	0.88
AR								9.2	9.2	3.1	5.7	4.4	2.6	**2.4**	5.1	8.1	5.0

References

Argyriou, A., Herbster, M., and Pontil, M. Combining graph laplacians for semi–supervised learning. *Advances in Neural Information Processing Systems*, 18, 2005.

Belkin, M., Matveeva, I., and Niyogi, P. Regularization and semi-supervised learning on large graphs. In *International Conference on Computational Learning Theory*, pp. 624–638. Springer, 2004.

Bentley, B., Branicky, R., Barnes, C. L., Chew, Y. L., Yemini, E., Bullmore, E. T., Vértes, P. E., and Schafer, W. R. The multilayer connectome of Caenorhabditis elegans. *PLoS Computational Biology*, 12(12):e1005283, 2016.

Berahas, A. S., Cao, L., Choromanski, K., and Scheinberg, K. A theoretical and empirical comparison of gradient approximations in derivative-free optimization. *Foundations of Computational Mathematics*, 22(2):507–560, 2022.

Bergermann, K., Stoll, M., and Volkmer, T. Semi-supervised learning for aggregated multilayer graphs using diffuse interface methods and fast matrix-vector products. *SIAM Journal on Mathematics of Data Science*, 3(2):758–785, 2021.

Bianconi, G. *Multilayer networks: structure and function*. Oxford University Press, 2018.

Bomze, I. M., Rinaldi, F., and Zeffiro, D. Active set complexity of the away-step Frank–Wolfe algorithm. *SIAM Journal on Optimization*, 30(3):2470–2500, 2020.

Bomze, I. M., Rinaldi, F., and Zeffiro, D. Frank–Wolfe and friends: a journey into projection-free first-order optimization methods. *4OR*, 19(3):313–345, 2021.

Choobdar, S., Ahsen, M. E., Crawford, J., Tomasoni, M., Fang, T., Lamparter, D., Lin, J., Hescott, B., Hu, X., Mercer, J., et al. Assessment of network module identification across complex diseases. *Nature Methods*, 16(9): 843–852, 2019.

De Domenico, M., Solé-Ribalta, A., Cozzo, E., Kivelä, M., Moreno, Y., Porter, M. A., Gómez, S., and Arenas, A. Mathematical formulation of multilayer networks. *Physical Review X*, 3(4):041022, 2013.

De Domenico, M., Solé-Ribalta, A., Gómez, S., and Arenas, A. Navigability of interconnected networks under random failures. *Proceedings of the National Academy of Sciences*, 111(23):8351–8356, 2014.

Dickison, M. E., Magnani, M., and Rossi, L. *Multilayer social networks*. Cambridge University Press, 2016.

Dua, D. and Efi, K. UCI machine learning repository multi-objective particle swarm optimization: Theory, 2017.

Eswaran, D., Günnemann, S., Faloutsos, C., Makhija, D., and Kumar, M. Zoobp: Belief propagation for heterogeneous networks. *Proceedings of the VLDB Endowment*, 10(5):625–636, 2017.

Frank, M. and Wolfe, P. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2): 95–110, 1956.

Freund, R. M. and Grigas, P. New analysis and results for the Frank–Wolfe method. *Mathematical Programming*, 155(1):199–230, 2016.

Gao, J., Buldyrev, S. V., Stanley, H. E., and Havlin, S. Networks formed from interdependent networks. *Nature Physics*, 8(1):40–48, 2012.

Gasteiger, J., Bojchevski, A., and Günnemann, S. Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations*, 2018.

Ghorbanchian, R., Latora, V., and Bianconi, G. Hyperdiffusion on multiplex networks. *Journal of Physics: Complexity*, 3(3):035009, 2022.

Ghorbani, M., Baghshah, M. S., and Rabiee, H. R. Mgcn: semi-supervised classification in multi-layer graphs with graph convolutional networks. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 208–211, 2019.

Grassia, M., De Domenico, M., and Mangioni, G. mgnn: Generalizing the graph neural networks to the multilayer case. *arXiv preprint arXiv:2109.10119*, 2021.

Greene, D. and Cunningham, P. Producing accurate interpretable clusters from high-dimensional data. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 486–494. Springer, 2005.

Greene, D. and Cunningham, P. A matrix factorization approach for integrating multiple data views. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 423–438. Springer, 2009.

Gujral, E. and Papalexakis, E. E. Smacd: Semi-supervised multi-aspect community detection. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pp. 702–710. SIAM, 2018.

Hanteer, O., Rossi, L., D'Aurelio, D. V., and Magnani, M. From interaction to participation: The role of the imagined audience in social media community detection and

an application to political communication on twitter. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 531–534. IEEE, 2018.

Hein, M., Setzer, S., Jost, L., and Rangapuram, S. S. The total variation on hypergraphs: learning on hypergraphs revisited. *Advances in Neural Information Processing Systems*, 26, 2013.

Higham, K., Contisciani, M., and De Bacco, C. Multilayer patent citation networks: A comprehensive analytical framework for studying explicit technological relationships. *Technological Forecasting and Social Change*, 179: 121628, 2022.

Huang, Q., He, H., Singh, A., Lim, S.-N., and Benson, A. Combining label propagation and simple models outperforms graph neural networks. In *International Conference on Learning Representations*, 2020.

Jaggi, M. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*, pp. 427–435. PMLR, 2013.

Karasuyama, M. and Mamitsuka, H. Multiple graph label propagation by sparse integration. *IEEE Transactions on Neural Networks and Learning Systems*, 24(12):1999–2012, 2013.

Kato, T., Kashima, H., and Sugiyama, M. Robust label propagation on multiple networks. *IEEE Transactions on Neural Networks*, 20(1):35–44, 2008.

Liu, J., Wang, C., Gao, J., and Han, J. Multi-view clustering via joint nonnegative matrix factorization. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 252–260. SIAM, 2013.

Lu, Q. and Getoor, L. Link-based classification using labeled and unlabeled data. In *ICML 2003 Workshop on The Continuum_from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003.

Magnani, M., Hanteer, O., Interdonato, R., Rossi, L., and Tagarelli, A. Community detection in multiplex networks. *ACM Computing Surveys (CSUR)*, 54(3):1–35, 2021.

Mangioni, G., Jurman, G., and De Domenico, M. Multilayer flows in molecular networks identify biological modules in the human proteome. *IEEE Transactions on Network Science and Engineering*, 7(1):411–420, 2018.

Martí, R., Resende, M. G., and Ribeiro, C. C. Multi-start methods for combinatorial optimization. *European Journal of Operational Research*, 226(1):1–8, 2013.

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

Mercado, P., Tudisco, F., and Hein, M. Generalized matrix means for semi-supervised learning with multilayer graphs. *Advances in Neural Information Processing Systems*, 32, 2019.

Mondragon, R. J., Iacovacci, J., and Bianconi, G. Multilink communities of multiplex networks. *PloS one*, 13(3): e0193821, 2018.

Nesterov, Y. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

Nie, F., Li, J., Li, X., et al. Parameter-free auto-weighted multiple graph learning: a framework for multiview clustering and semi-supervised classification. In *IJCAI*, pp. 1881–1887, 2016.

Peel, L., Peixoto, T. P., and De Domenico, M. Statistical inference links data and theory in network science. *Nature Communications*, 13(1):6794, 2022.

Prokopchik, K., Benson, A. R., and Tudisco, F. Nonlinear feature diffusion on hypergraphs. In *International Conference on Machine Learning*, pp. 17945–17958. PMLR, 2022.

Rasiwasia, N., Costa Pereira, J., Coviello, E., Doyle, G., Lanckriet, G. R., Levy, R., and Vasconcelos, N. A new approach to cross-modal multimedia retrieval. In *Proceedings of the 18th ACM International Conference on Multimedia*, pp. 251–260, 2010.

Rinaldi, F. and Zeffiro, D. Avoiding bad steps in Frank-Wolfe variants. *Computational Optimization and Applications*, pp. 1–40, 2022.

Rossi, L. and Magnani, M. Towards effective visual analytics on multiplex and multilayer networks. *Chaos, Solitons & Fractals*, 72:68–76, 2015.

Sahu, A. K., Zaheer, M., and Kar, S. Towards gradient free and projection free stochastic optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3468–3477. PMLR, 2019.

Song, Z., Yang, X., Xu, Z., and King, I. Graph-based semi-supervised learning: A comprehensive review. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Tsuda, K., Shin, H., and Schölkopf, B. Fast protein classification with multiple networks. *Bioinformatics*, 21 (suppl_2):ii59–ii65, 2005.

Tudisco, F., Benson, A. R., and Prokopchik, K. Nonlinear higher-order label spreading. In *Proceedings of the Web Conference 2021*, pp. 2402–2413, 2021.

Varga, R. S. *Geršgorin and his circles*, volume 36. Springer Science & Business Media, 2010.

Venturini, S., Cristofari, A., Rinaldi, F., and Tudisco, F. A variance-aware multiobjective Louvain-like method for community detection in multiplex networks. *Journal of Complex Networks*, 10(6):cnac048, 2022.

Viswanathan, K., Sachdeva, S., Tomkins, A., and Ravi, S. Improved semi-supervised learning with multiple graphs. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3032–3041. PMLR, 2019.

Wang, Z., Wang, L., Szolnoki, A., and Perc, M. Evolutionary games on multilayer networks: a colloquium. *The European physical Journal B*, 88(5):1–15, 2015.

Welling, M. and Kipf, T. N. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*, 2016.

Yang, Z., Cohen, W., and Salakhudinov, R. Revisiting semi-supervised learning with graph embeddings. In *International Conference on Machine Learning*, pp. 40–48. PMLR, 2016.

Ye, J. and Akoglu, L. Robust semi-supervised learning on multiple networks with noise. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 196–208. Springer, 2018.

Zhou, D. and Burges, C. J. Spectral clustering and transductive learning with multiple views. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 1159–1166, 2007.

Zhou, D., Bousquet, O., Lal, T., Weston, J., and Schölkopf, B. Learning with local and global consistency. *Advances in Neural Information Processing Systems*, 16, 2003.

A. Proof of Theorem 4.2

The following chain of inequalities holds:

$$-\nabla f(\boldsymbol{\theta}_n)^\top d_n^{FW} \geq -\nabla f(\boldsymbol{\theta}_n)^\top d_n \geq -\widetilde{\nabla} f(\boldsymbol{\theta}_n)^\top d_n - \epsilon_n \geq -\widetilde{\nabla} f(\boldsymbol{\theta}_n)^\top d_n^{FW} - \epsilon_n \geq -\nabla f(\boldsymbol{\theta}_n)^\top d_n^{FW} - 2\epsilon_n, \quad (18)$$

where we used (8) in the second and the last inequality, while the first and the third inequality follow from the definition of d_n^{FW} and d_n. In particular, using the definitions of $\widetilde{g}^n$, g_n and g_n^{FW}, from (18) we can write

$$g_n^{FW} \geq \widetilde{g}_n - \epsilon_n, \quad (19)$$

$$\widetilde{g}_n \geq g_n^{FW} - \epsilon_n, \quad (20)$$

$$g_n \geq \widetilde{g}_n - \epsilon_n \quad (21)$$

Using (9) and (19), we also have

$$\epsilon_n \leq \sigma(\widetilde{g}_n - \epsilon_n) \leq \sigma g_n^{FW}, \quad (22)$$

Now, let us distinguish two cases.

- If $\bar{\eta}_n < 1$, from (10) it follows that $\dfrac{\widetilde{g}_n}{M\|d_n\|^2} < 1$. Using (11) we can write

$$f(\boldsymbol{\theta}_n) - f(\boldsymbol{\theta}_n + \eta_n d_n) \geq \rho\bar{\eta}_n\widetilde{g}_n = \frac{\rho}{M\|d_n\|^2}\widetilde{g}_n^2 \geq \frac{\rho\widetilde{g}_n^2}{\Delta^2 M},$$

  where the last inequality follows from $\|d_n\| \leq \Delta$. Observe that, from (20) and (22), we have $\widetilde{g}_n \geq (1-\sigma)g_n^{FW}$. Therefore,

$$f(\boldsymbol{\theta}_n) - f(\boldsymbol{\theta}_{n+1}) \geq \frac{\rho(1-\sigma)^2}{\Delta^2 M}(g_n^{FW})^2. \quad (23)$$

- If $\bar{\eta}_n = 1$, from (10) it follows that $\dfrac{\widetilde{g}_n}{M\|d_n\|^2} \geq 1$ and, since $\eta_n \leq 1$ from the instructions of the algorithm, then $\eta_n = 1$. By the standard descent lemma we can write

$$f(\boldsymbol{\theta}_{n+1}) = f(\boldsymbol{\theta}_n + d_n) \leq f(\boldsymbol{\theta}_n) - g_n + \frac{M}{2}\|d_n\|^2 \leq f(\boldsymbol{\theta}_n) - (\widetilde{g}_n - \epsilon_n) + \frac{M}{2}\|d_n\|^2,$$

  where we used (21) in the last inequality. Since we are analyzing the case where $\widetilde{g}_n \geq \|d_n\|^2 M$, we obtain

$$f(\boldsymbol{\theta}_n) - f(\boldsymbol{\theta}_{n+1}) \geq \frac{\widetilde{g}_n}{2} - \epsilon_n.$$

  Using (20) and (22), we also have

$$\frac{\widetilde{g}_n}{2} - \epsilon_n \geq \frac{g_n^{FW}}{2} - \frac{3}{2}\epsilon_n \geq \frac{g_n^{FW}}{2} - \frac{3}{2}\sigma g_n^{FW} = \frac{1-3\sigma}{2}g_n^{FW}.$$

  Therefore,

$$f(\boldsymbol{\theta}_n) - f(\boldsymbol{\theta}_{n+1}) \geq \frac{1-3\sigma}{2}g_n^{FW}. \quad (24)$$

Now, based on the two cases analyzed above, we partition the iterations $\{0, 1, \ldots, T-1\}$ into two subsets N_1 and N_2 defined as follows:

$$N_1 = \{n < T \colon \bar{\eta}_n < 1\}, \quad N_2 = \{n < T \colon \bar{\eta}_n = 1\}.$$

Using (23) and (24), we can write:

$$
\begin{aligned}
f(\boldsymbol{\theta}_0) - f^* &\geq \sum_{n=0}^{T-1} (f(\boldsymbol{\theta}_n) - f(\boldsymbol{\theta}_{n+1})) \\
&= \sum_{N_1} (f(\boldsymbol{\theta}_n) - f(\boldsymbol{\theta}_{n+1})) + \sum_{N_2} (f(\boldsymbol{\theta}_n) - f(\boldsymbol{\theta}_{n+1})) \\
&\geq \sum_{N_1} \frac{\rho(1-\sigma)^2}{\Delta^2 M} (g_n^{FW})^2 + \sum_{N_2} \frac{1-3\sigma}{2} g_n^{FW} \\
&\geq |N_1| \min_{n \in N_1} \frac{\rho(1-\sigma)^2}{\Delta^2 M} (g_n^{FW})^2 + |N_2| \min_{n \in N_2} \frac{1-3\sigma}{2} g_n^{FW} \\
&\geq (|N_1| + |N_2|) \min \left(\frac{\rho(1-\sigma)^2}{\Delta^2 M} (g_T^*)^2, \frac{1-3\sigma}{2} g_T^* \right) \\
&= T \min \left(\frac{\rho(1-\sigma)^2}{\Delta^2 M} (g_T^*)^2, \frac{1-3\sigma}{2} g_T^* \right),
\end{aligned}
$$

where the last inequality follows from the definition of g_T^*. Hence,

$$
\begin{aligned}
\frac{\rho(1-\sigma)^2}{\Delta^2 M} (g_T^*)^2 \leq \frac{1-3\sigma}{2} g_T^* \;\Rightarrow\; g_T^* \leq \sqrt{\frac{\Delta^2 M (f(\boldsymbol{\theta}_0) - f^*)}{T\rho(1-\sigma)^2}}, \\
\frac{\rho(1-\sigma)^2}{\Delta^2 M} (g_T^*)^2 > \frac{1-3\sigma}{2} g_T^* \;\Rightarrow\; g_T^* \leq \frac{2(f(\boldsymbol{\theta}_0) - f^*)}{T(1-3\sigma)},
\end{aligned}
$$

leading to the desired result.

B. Proof of Lemma 4.3

Reasoning as in the proof of Theorem 4.2, we have that (21) holds. By the standard descent lemma, we have

$$
f(\boldsymbol{\theta}_n) - f(\boldsymbol{\theta}_n + \eta d_n) \geq \eta g_n - \eta^2 \frac{M\|d_n\|^2}{2} \geq \eta(\widetilde{g}_n - \epsilon_n) - \eta^2 \frac{M\|d_n\|^2}{2}, \quad \forall \eta \in \mathbb{R}, \tag{25}
$$

where the last inequality follows from (21). Then,

$$
f(\boldsymbol{\theta}_n) - f(\boldsymbol{\theta}_n + \eta d_n) \geq \gamma \eta \widetilde{g}_n \quad \forall \eta \in \left[0, 2\frac{(1-\gamma)\widetilde{g}_n - \epsilon_n}{M\|d_n\|^2} \right].
$$

Since η_n is computed by (13)–(14), we can write

$$
\begin{aligned}
\eta_n &\geq \min \left(1, 2\delta \frac{(1-\gamma)\widetilde{g}_n - \epsilon_n}{M\|d_n\|^2} \right) \\
&\geq \min \left(1, 2\delta \frac{(1-\gamma-\sigma)\widetilde{g}_n}{M\|d_n\|^2} \right) \\
&\geq \min(1, 2\delta(1-\gamma-\sigma))\bar{\eta}_n,
\end{aligned} \tag{26}
$$

where the second inequality follows from (15).

C. Complexity Analysis

We detail below the computational cost of the method proposed in Section 3 and provide a table with a time-execution comparison with respect to the competing methods from Table 3 and Tables 4-5.

In Theorem 4.2 we have shown a sublinear convergence rate of the duality gap g_n^{FW}, that is, $g_n^* \leq \max(c_1 n^{-\frac{1}{2}}, c_2 n^{-1})$ with appropriate constants c_1 and c_2. Then, complexity results can be straightforwardly obtained by standard arguments of information-based complexity theory (Nesterov, 2003). In particular, in our case we have a worst-case complexity of $\mathcal{O}(\epsilon^{-2})$ for the number of iterations to drive g_n^* below ϵ. Additionally, we can easily give an upper bound on the number of arithmetic operations carried out at every iteration of the FW method: each iteration requires $\mathcal{O}(K)$ function evaluations to estimate the gradient and $\mathcal{O}(K)$ operations to solve the linear subproblem (the line search has a cost $\mathcal{O}(1)$ assuming the knowledge of the Lipschitz constant M), where K is the number of layers. Moreover, each function evaluation requires the solution of problem (4) (or problem (5)), that is, r iterations of Label Propagation algorithm yielding a cost of $\mathcal{O}(Nr)$ if the graph is sparse, where N is the number of nodes. Summing up, each iteration of the FW method a cost of $\mathcal{O}(KNr)$ and then we need $\mathcal{O}(\epsilon^{-2}KNr)$ arithmetic operations to drive g_n^* below ϵ.
In Table 6, we report the average time-execution comparison over three runs (in seconds) on synthetic datasets generated as in Subsection 5.1 (3 communities equal size and 3 layers) as the number of nodes increases.

Table 6. Average time-execution comparison over three runs (in seconds) on synthetic datasets with 3 communities of equal size and 3 layers, as the number of nodes N increases.

N	1	2	3	MIN	GEOM	ARIT	HARM	MAX	BINOM	MULTI	SGMI	AGML	SMACD	GMM
1200	0.007	0.007	0.007	0.011	0.012	0.009	0.005	0.005	6.1	15.017	0.155	0.711	26.397	1.027
2400	0.011	0.01	0.01	0.017	0.018	0.014	0.005	0.005	6.493	17.614	0.72	5.434	58.344	9.136
3600	0.014	0.013	0.012	0.021	0.022	0.017	0.005	0.006	6.914	17.143	1.47	13.608	97.081	27.543
4800	0.016	0.014	0.014	0.024	0.026	0.022	0.006	0.007	9.675	17.973	3.132	31.458	127.851	73.991
6000	0.018	0.016	0.016	0.028	0.032	0.028	0.008	0.008	9.074	28.186	6.809	60.876	132.44	173.659
7200	0.02	0.018	0.017	0.033	0.037	0.033	0.009	0.009	9.98	26.953	10.618	123.825	188.622	299.11
8400	0.024	0.021	0.02	0.04	0.045	0.04	0.01	0.01	11.984	29.765	10.586	132.112	157.038	488.309
9600	0.025	0.023	0.019	0.038	0.041	0.037	0.009	0.009	12.959	32.789	21.794	294.769	285.303	789.337
10800	0.034	0.027	0.023	0.044	0.173	0.045	0.011	0.011	32.9	98.419	47.944	561.55	348.047	1040.754
12000	0.026	0.023	0.022	0.047	0.051	0.047	0.011	0.011	30.391	96.565	44.969	546.674	393.805	1626.726

D. Additional Results

In Tables 7-10, we report tests performed with different number of initial known labels per community (1%, 5%, 10%, 15%), considering the informative case and the adding of one or two layers of noise. The results are aligned with those presented in Tables 4 and 5, with the proposed BINOM and MULTI approaches being overall the best performing.

Table 7. Accuracy (mean ± std) over three random samples of the 1% of input labels, on real-world datasets (+ one and two layers of noise).

	I	II	III	IV	V	VI		MIN	GEOM	ARIT	HARM	MAX	BINOM	MULTI	SGMI	SMACD	GMM
3sources	0.72	0.69	0.77	-	-	-		0.66±0.01	0.66±0.01	0.79±0.04	0.71±0.04	0.76±0.04	0.79±0.04	0.79±0.04	0.65±0.04	0.58±0.06	0.69±0.06
							(+1 noise)	0.35±0.01	0.35±0.01	0.73±0.06	0.55±0.05	0.69±0.07	0.73±0.06	0.73±0.06	0.52±0.22	0.65±0.08	0.62±0.02
							(+2 noise)	0.34±0.0	0.34±0.0	0.7±0.04	0.48±0.02	0.63±0.07	0.7±0.04	0.7±0.04	0.52±0.22	0.55±0.07	0.53±0.03
BBC	0.77	0.76	0.72	0.75	-	-		0.35±0.01	0.35±0.01	0.83±0.01	0.81±0.01	0.82±0.01	0.83±0.01	0.83±0.01	0.64±0.03	0.59±0.07	0.68±0.03
							(+1 noise)	0.33±0.0	0.33±0.0	0.78±0.01	0.74±0.01	0.76±0.01	0.78±0.01	0.83±0.01	0.23±0.01	0.62±0.04	0.6±0.02
							(+2 noise)	0.33±0.0	0.33±0.0	0.73±0.01	0.68±0.02	0.71±0.01	0.73±0.01	0.73±0.01	0.23±0.01	0.56±0.04	0.55±0.02
BBCSport	0.87	0.76	-	-	-	-		0.63±0.07	0.63±0.07	0.87±0.03	0.87±0.03	0.87±0.03	0.87±0.03	0.87±0.03	0.73±0.09	0.79±0.09	0.7±0.06
							(+1 noise)	0.36±0.0	0.36±0.0	0.73±0.03	0.71±0.02	0.73±0.03	0.73±0.03	0.73±0.03	0.24±0.02	0.79±0.04	0.54±0.04
							(+2 noise)	0.36±0.0	0.36±0.0	0.66±0.03	0.62±0.03	0.65±0.03	0.66±0.03	0.77±0.07	0.24±0.02	0.73±0.16	0.5±0.04
Wikipedia	0.17	0.61	-	-	-	-		0.17±0.01	0.17±0.01	0.42±0.03	0.42±0.04	0.42±0.03	0.42±0.03	0.42±0.03	0.58±0.03	0.31±0.05	0.43±0.05
							(+1 noise)	0.15±0.0	0.15±0.0	0.34±0.01	0.34±0.01	0.34±0.01	0.34±0.01	0.33±0.02	0.15±0.0	0.26±0.07	0.33±0.03
							(+2 noise)	0.15±0.0	0.15±0.0	0.32±0.01	0.31±0.02	0.31±0.02	0.32±0.01	0.32±0.01	0.15±0.0	0.23±0.08	0.28±0.03
UCI	0.79	0.72	0.9	0.43	0.91	0.69		0.1±0.0	0.1±0.0	0.85±0.01	0.79±0.02	0.82±0.02	0.85±0.01	0.81±0.02	0.56±0.14	0.35±0.11	0.58±0.02
							(+1 noise)	0.1±0.0	0.1±0.0	0.82±0.01	0.75±0.02	0.78±0.02	0.5±0.02	0.82±0.01	0.13±0.0	0.26±0.06	0.54±0.0
							(+2 noise)	0.1±0.0	0.1±0.0	0.79±0.01	0.71±0.02	0.75±0.02	0.89±0.03	0.89±0.03	0.13±0.0	0.23±0.06	0.51±0.01
cora	0.58	0.49	-	-	-	-		0.3±0.01	0.3±0.01	0.56±0.04	0.56±0.04	0.56±0.04	0.57±0.03	0.56±0.03	0.55±0.05	0.37±0.07	0.45±0.03
							(+1 noise)	0.3±0.0	0.3±0.0	0.43±0.02	0.42±0.02	0.43±0.02	0.49±0.02	0.53±0.09	0.58±0.02	0.36±0.07	0.4±0.02
							(+2 noise)	0.3±0.0	0.3±0.0	0.35±0.02	0.34±0.02	0.34±0.02	0.44±0.04	0.53±0.09	0.58±0.02	0.36±0.05	0.36±0.01
citeseer	0.41	0.5	-	-	-	-		0.23±0.01	0.23±0.01	0.53±0.04	0.54±0.03	0.54±0.04	0.55±0.03	0.54±0.03	0.4±0.01	0.3±0.08	0.38±0.03
							(+1 noise)	0.21±0.0	0.21±0.0	0.36±0.02	0.36±0.02	0.36±0.02	0.39±0.06	0.49±0.03	0.39±0.01	0.37±0.06	0.32±0.01
							(+2 noise)	0.21±0.0	0.21±0.0	0.3±0.02	0.3±0.02	0.3±0.02	0.32±0.04	0.5±0.02	0.39±0.01	0.25±0.05	0.29±0.0
dkpol	0.29	0.16	0.67	-	-	-		0.15±0.0	0.15±0.0	0.67±0.03	0.66±0.03	0.67±0.03	0.67±0.03	0.67±0.03	0.19±0.04	0.24±0.06	0.57±0.02
							(+1 noise)	0.14±0.0	0.14±0.0	0.57±0.04	0.55±0.03	0.56±0.03	0.65±0.04	0.65±0.04	0.17±0.01	0.34±0.15	0.31±0.02
							(+2 noise)	0.14±0.0	0.14±0.0	0.51±0.03	0.49±0.03	0.5±0.03	0.33±0.02	0.33±0.02	0.17±0.01	0.22±0.03	0.23±0.04
aucs	0.34	0.36	0.61	0.8	0.72	-		0.3±0.01	0.3±0.01	0.85±0.04	0.8±0.06	0.85±0.05	0.84±0.04	0.85±0.03	0.75±0.07	0.5±0.16	0.81±0.05
							(+1 noise)	0.27±0.0	0.27±0.0	0.81±0.04	0.37±0.03	0.65±0.03	0.81±0.04	0.81±0.04	0.76±0.09	0.42±0.03	0.77±0.09
							(+2 noise)	0.27±0.0	0.27±0.0	0.77±0.05	0.33±0.02	0.52±0.04	0.73±0.1	0.73±0.1	0.76±0.09	0.52±0.08	0.75±0.09
APR								0.4	0.4	0.92	0.83	0.88	0.92	**0.96**	0.62	0.64	0.74
AR								9.1	9.1	2.9	5.4	4.1	2.8	**2.4**	6.5	6.6	6.0

Table 8. Accuracy (mean ± std) over three random samples of the 5% of input labels, on real-world datasets (+ one and two layers of noise).

	I	II	III	IV	V	VI		MIN	GEOM	ARIT	HARM	MAX	BINOM	MULTI	SGMI	SMACD	GMM
3sources	0.73	0.72	0.77	-	-	-		0.68±0.02	0.68±0.02	0.8±0.06	0.72±0.03	0.78±0.04	0.76±0.07	0.72±0.07	0.72±0.03	0.58±0.05	0.77±0.05
							(+1 noise)	0.35±0.01	0.35±0.01	0.74±0.06	0.55±0.04	0.68±0.06	0.69±0.09	0.61±0.08	0.56±0.25	0.64±0.1	0.71±0.05
							(+2 noise)	0.34±0.0	0.34±0.0	0.72±0.04	0.47±0.02	0.62±0.05	0.79±0.05	0.72±0.07	0.55±0.27	0.7±0.1	0.65±0.05
BBC	0.81	0.80	0.77	0.81	-	-		0.37±0.02	0.37±0.02	0.88±0.02	0.85±0.01	0.87±0.02	0.84±0.02	0.83±0.03	0.73±0.01	0.59±0.07	0.81±0.03
							(+1 noise)	0.33±0.0	0.33±0.0	0.85±0.02	0.78±0.02	0.82±0.02	0.82±0.01	0.8±0.03	0.57±0.28	0.58±0.06	0.74±0.02
							(+2 noise)	0.33±0.0	0.33±0.0	0.81±0.02	0.71±0.02	0.76±0.02	0.78±0.08	0.79±0.01	0.42±0.26	0.64±0.04	0.7±0.02
BBCSport	0.89	0.86	-	-	-	-		0.78±0.01	0.78±0.01	0.91±0.01	0.9±0.01	0.91±0.01	0.91±0.02	0.88±0.01	0.81±0.04	0.79±0.09	0.84±0.02
							(+1 noise)	0.36±0.0	0.36±0.0	0.82±0.03	0.79±0.03	0.8±0.02	0.87±0.01	0.87±0.01	0.41±0.28	0.75±0.11	0.68±0.04
							(+2 noise)	0.35±0.0	0.35±0.0	0.73±0.04	0.67±0.03	0.7±0.03	0.85±0.03	0.87±0.01	0.41±0.28	0.72±0.08	0.62±0.03
Wikipedia	0.17	0.60	-	-	-	-		0.18±0.01	0.18±0.01	0.44±0.03	0.44±0.04	0.44±0.04	0.57±0.02	0.57±0.08	0.59±0.02	0.31±0.06	0.48±0.04
							(+1 noise)	0.15±0.0	0.15±0.0	0.34±0.01	0.34±0.01	0.34±0.01	0.47±0.02	0.5±0.1	0.16±0.01	0.28±0.05	0.37±0.01
							(+2 noise)	0.15±0.0	0.15±0.0	0.3±0.01	0.29±0.01	0.3±0.01	0.42±0.03	0.57±0.02	0.16±0.01	0.23±0.0	0.32±0.02
UCI	0.9	0.8	0.95	0.51	0.96	0.8		0.1±0.0	0.1±0.0	0.92±0.01	0.86±0.01	0.89±0.01	0.95±0.03	0.94±0.01	0.87±0.07	0.29±0.09	0.85±0.01
							(+1 noise)	0.1±0.0	0.1±0.0	0.93±0.01	0.85±0.02	0.9±0.01	0.96±0.01	0.91±0.04	0.91±0.0	0.29±0.09	0.82±0.01
							(+2 noise)	0.1±0.0	0.1±0.0	0.93±0.01	0.82±0.02	0.89±0.01	0.96±0.01	0.94±0.01	0.91±0.0	0.26±0.06	0.81±0.02
cora	0.69	0.59	-	-	-	-		0.32±0.01	0.32±0.01	0.66±0.01	0.66±0.01	0.66±0.01	0.7±0.03	0.7±0.05	0.67±0.02	0.37±0.06	0.61±0.01
							(+1 noise)	0.3±0.0	0.3±0.0	0.53±0.01	0.52±0.01	0.53±0.01	0.59±0.02	0.64±0.02	0.67±0.02	0.31±0.04	0.5±0.01
							(+2 noise)	0.3±0.0	0.3±0.0	0.44±0.01	0.43±0.01	0.44±0.01	0.64±0.02	0.64±0.05	0.67±0.02	0.35±0.06	0.45±0.0
citeseer	0.46	0.59	-	-	-	-		0.27±0.0	0.27±0.0	0.62±0.01	0.62±0.01	0.62±0.01	0.63±0.01	0.63±0.01	0.45±0.02	0.26±0.05	0.51±0.01
							(+1 noise)	0.21±0.0	0.21±0.0	0.49±0.01	0.47±0.01	0.48±0.01	0.5±0.02	0.6±0.02	0.46±0.02	0.26±0.05	0.41±0.01
							(+2 noise)	0.21±0.0	0.21±0.0	0.41±0.01	0.39±0.01	0.4±0.02	0.48±0.01	0.6±0.02	0.46±0.02	0.24±0.05	0.38±0.01
dkpol	0.3	0.16	0.69	-	-	-		0.15±0.0	0.15±0.0	0.7±0.04	0.69±0.03	0.7±0.03	0.65±0.05	0.59±0.09	0.27±0.03	0.24±0.08	0.59±0.04
							(+1 noise)	0.14±0.0	0.14±0.0	0.58±0.04	0.57±0.04	0.58±0.03	0.63±0.08	0.6±0.02	0.3±0.01	0.24±0.03	0.34±0.03
							(+2 noise)	0.14±0.0	0.14±0.0	0.53±0.03	0.51±0.03	0.52±0.03	0.51±0.1	0.57±0.03	0.3±0.01	0.19±0.04	0.27±0.03
aucs	0.34	0.36	0.61	0.8	0.72	-		0.3±0.01	0.3±0.01	0.85±0.04	0.8±0.06	0.85±0.05	0.86±0.05	0.86±0.05	0.75±0.07	0.5±0.16	0.81±0.05
							(+1 noise)	0.27±0.0	0.27±0.0	0.81±0.04	0.37±0.03	0.65±0.03	0.81±0.04	0.81±0.04	0.76±0.09	0.48±0.08	0.77±0.09
							(+2 noise)	0.27±0.0	0.27±0.0	0.77±0.05	0.33±0.02	0.52±0.04	0.77±0.05	0.77±0.05	0.76±0.09	0.48±0.16	0.75±0.09
APR								0.37	0.37	0.9	0.8	0.86	0.95	**0.96**	0.74	0.56	0.8
AR								9.3	9.3	3.0	5.4	4.2	**2.3**	2.8	5.6	7.6	5.6

Table 9. Accuracy (mean ± std) over three random samples of the 10% of input labels, on real-world datasets (+ one and two layers of noise).

	I	II	III	IV	V	VI		MIN	GEOM	ARIT	HARM	MAX	BINOM	MULTI	SGMI	SMACD	GMM
								0.71±0.07	0.71±0.07	0.79±0.07	0.69±0.05	0.75±0.05	0.75±0.07	0.74±0.05	0.75±0.04	0.62±0.04	0.8±0.05
3sources	0.76	0.72	0.75	-	-	-	(+1 noise)	0.34±0.01	0.34±0.01	0.74±0.1	0.54±0.06	0.66±0.09	0.76±0.11	0.73±0.06	0.58±0.27	0.6±0.08	0.76±0.04
							(+2 noise)	0.33±0.0	0.33±0.0	0.7±0.08	0.46±0.06	0.6±0.08	0.72±0.15	0.75±0.07	0.57±0.28	0.61±0.21	0.71±0.05
								0.38±0.01	0.38±0.01	0.91±0.01	0.89±0.01	0.9±0.01	0.88±0.01	0.86±0.04	0.76±0.02	0.69±0.03	0.87±0.01
BBC	0.83	0.83	0.79	0.83	-	-	(+1 noise)	0.33±0.0	0.33±0.0	0.89±0.01	0.84±0.01	0.87±0.02	0.87±0.02	0.87±0.01	0.76±0.03	0.66±0.02	0.84±0.01
							(+2 noise)	0.33±0.0	0.33±0.0	0.88±0.01	0.79±0.01	0.84±0.02	0.87±0.01	0.83±0.03	0.76±0.03	0.65±0.07	0.81±0.01
								0.79±0.0	0.79±0.0	0.92±0.01	0.92±0.0	0.92±0.0	0.92±0.02	0.88±0.01	0.84±0.03	0.73±0.1	0.88±0.02
BBCSport	0.90	0.88	-	-	-	-	(+1 noise)	0.36±0.0	0.36±0.0	0.86±0.02	0.82±0.01	0.85±0.02	0.91±0.01	0.87±0.0	0.83±0.04	0.6±0.07	0.77±0.02
							(+2 noise)	0.35±0.0	0.35±0.0	0.8±0.02	0.74±0.02	0.77±0.02	0.9±0.01	0.87±0.01	0.64±0.31	0.78±0.03	0.73±0.03
								0.19±0.01	0.19±0.01	0.51±0.03	0.51±0.03	0.51±0.03	0.62±0.04	0.64±0.02	0.61±0.03	0.24±0.04	0.57±0.03
Wikipedia	0.18	0.65	-	-	-	-	(+1 noise)	0.15±0.0	0.15±0.0	0.42±0.01	0.42±0.01	0.42±0.01	0.57±0.06	0.62±0.03	0.59±0.02	0.25±0.04	0.47±0.02
							(+2 noise)	0.15±0.0	0.15±0.0	0.39±0.01	0.38±0.01	0.38±0.01	0.56±0.06	0.59±0.03	0.59±0.02	0.23±0.08	0.43±0.02
								0.11±0.0	0.11±0.0	0.95±0.01	0.88±0.01	0.92±0.01	0.97±0.01	0.96±0.01	0.94±0.01	0.33±0.05	0.93±0.01
UCI	0.92	0.81	0.96	0.57	0.97	0.82	(+1 noise)	0.1±0.0	0.1±0.0	0.96±0.0	0.88±0.01	0.92±0.0	0.97±0.0	0.96±0.01	0.94±0.01	0.3±0.1	0.93±0.01
							(+2 noise)	0.1±0.0	0.1±0.0	0.96±0.0	0.88±0.01	0.92±0.0	0.97±0.0	0.96±0.01	0.94±0.01	0.33±0.05	0.93±0.01
								0.34±0.01	0.34±0.01	0.69±0.01	0.69±0.01	0.69±0.01	0.74±0.03	0.76±0.01	0.72±0.01	0.34±0.08	0.69±0.01
cora	0.73	0.62	-	-	-	-	(+1 noise)	0.3±0.0	0.3±0.0	0.57±0.01	0.56±0.01	0.57±0.01	0.63±0.01	0.76±0.01	0.72±0.02	0.28±0.01	0.58±0.01
							(+2 noise)	0.3±0.0	0.3±0.0	0.49±0.01	0.47±0.01	0.48±0.01	0.64±0.04	0.74±0.03	0.72±0.02	0.37±0.02	0.55±0.0
								0.29±0.01	0.29±0.01	0.65±0.01	0.65±0.01	0.65±0.01	0.66±0.01	0.65±0.01	0.51±0.01	0.36±0.1	0.58±0.01
citeseer	0.51	0.62	-	-	-	-	(+1 noise)	0.21±0.0	0.21±0.0	0.53±0.02	0.52±0.02	0.53±0.02	0.59±0.03	0.63±0.0	0.52±0.02	0.32±0.03	0.49±0.01
							(+2 noise)	0.21±0.0	0.21±0.0	0.47±0.02	0.44±0.01	0.46±0.01	0.61±0.03	0.63±0.0	0.52±0.02	0.26±0.05	0.45±0.01
								0.16±0.0	0.16±0.0	0.73±0.06	0.67±0.09	0.69±0.07	0.62±0.06	0.76±0.01	0.31±0.03	0.26±0.09	0.63±0.04
dkpol	0.32	0.16	0.69	-	-	-	(+1 noise)	0.14±0.0	0.14±0.0	0.62±0.02	0.59±0.02	0.6±0.02	0.54±0.08	0.72±0.08	0.31±0.05	0.24±0.01	0.34±0.01
							(+2 noise)	0.14±0.0	0.14±0.0	0.58±0.02	0.54±0.03	0.55±0.02	0.45±0.11	0.69±0.07	0.31±0.05	0.2±0.04	0.27±0.02
								0.3±0.01	0.3±0.01	0.85±0.04	0.8±0.06	0.85±0.05	0.85±0.04	0.85±0.02	0.75±0.07	0.58±0.03	0.81±0.05
aucs	0.34	0.36	0.61	0.8	0.72	-	(+1 noise)	0.27±0.0	0.27±0.0	0.81±0.04	0.37±0.03	0.65±0.03	0.81±0.04	0.81±0.04	0.76±0.09	0.56±0.07	0.77±0.09
							(+2 noise)	0.27±0.0	0.27±0.0	0.77±0.05	0.33±0.02	0.52±0.04	0.77±0.05	0.77±0.05	0.76±0.09	0.6±0.09	0.76±0.07
APR								0.36	0.36	0.9	0.8	0.86	0.94	**0.98**	0.84	0.56	0.85
AR								9.3	9.3	3.2	6.1	4.7	2.5	**2.4**	4.8	7.8	5.0

Table 10. Accuracy (mean ± std) over three random samples of the 15% of input labels, on real-world datasets (+ one and two layers of noise).

	I	II	III	IV	V	VI		MIN	GEOM	ARIT	HARM	MAX	BINOM	MULTI	SGMI	SMACD	GMM
								0.74±0.06	0.74±0.06	0.83±0.05	0.75±0.04	0.8±0.03	0.82±0.03	0.76±0.04	0.74±0.04	0.66±0.09	0.8±0.03
3sources	0.77	0.75	0.78	-	-	-	(+1 noise)	0.34±0.01	0.34±0.01	0.79±0.05	0.57±0.04	0.71±0.03	0.79±0.06	0.72±0.07	0.75±0.05	0.61±0.09	0.77±0.02
							(+2 noise)	0.34±0.0	0.34±0.0	0.76±0.09	0.47±0.04	0.63±0.04	0.75±0.08	0.72±0.02	0.75±0.05	0.63±0.15	0.76±0.07
								0.39±0.02	0.39±0.02	0.91±0.01	0.89±0.01	0.9±0.01	0.89±0.01	0.85±0.03	0.77±0.02	0.64±0.12	0.89±0.01
BBC	0.85	0.84	0.8	0.84	-	-	(+1 noise)	0.33±0.0	0.33±0.0	0.9±0.01	0.86±0.0	0.88±0.01	0.89±0.01	0.86±0.03	0.79±0.01	0.62±0.05	0.87±0.01
							(+2 noise)	0.33±0.0	0.33±0.0	0.88±0.01	0.83±0.01	0.86±0.0	0.88±0.01	0.85±0.03	0.79±0.01	0.64±0.01	0.84±0.01
								0.81±0.01	0.81±0.01	0.94±0.01	0.93±0.0	0.93±0.0	0.94±0.01	0.91±0.02	0.84±0.02	0.77±0.09	0.91±0.02
BBCSport	0.91	0.9	-	-	-	-	(+1 noise)	0.36±0.0	0.36±0.0	0.9±0.01	0.87±0.01	0.89±0.0	0.91±0.02	0.87±0.02	0.84±0.03	0.85±0.02	0.86±0.02
							(+2 noise)	0.35±0.0	0.35±0.0	0.87±0.01	0.81±0.01	0.84±0.01	0.94±0.01	0.88±0.01	0.84±0.03	0.85±0.07	0.8±0.03
								0.21±0.01	0.21±0.01	0.56±0.02	0.56±0.02	0.56±0.02	0.64±0.01	0.64±0.02	0.62±0.02	0.31±0.05	0.62±0.01
Wikipedia	0.17	0.65	-	-	-	-	(+1 noise)	0.15±0.0	0.15±0.0	0.51±0.01	0.51±0.01	0.52±0.01	0.61±0.02	0.63±0.03	0.61±0.01	0.32±0.06	0.55±0.01
							(+2 noise)	0.15±0.0	0.15±0.0	0.48±0.01	0.46±0.03	0.47±0.02	0.6±0.02	0.66±0.01	0.61±0.01	0.23±0.02	0.51±0.01
								0.11±0.0	0.11±0.0	0.96±0.01	0.89±0.01	0.93±0.0	0.97±0.0	0.97±0.01	0.95±0.01	0.35±0.11	0.96±0.01
UCI	0.92	0.82	0.96	0.59	0.97	0.83	(+1 noise)	0.1±0.0	0.1±0.0	0.97±0.0	0.9±0.0	0.94±0.0	0.97±0.0	0.97±0.01	0.95±0.01	0.26±0.06	0.96±0.01
							(+2 noise)	0.1±0.0	0.1±0.0	0.97±0.0	0.89±0.01	0.94±0.0	0.96±0.01	0.97±0.01	0.95±0.01	0.29±0.01	0.96±0.01
								0.35±0.01	0.35±0.01	0.71±0.0	0.7±0.0	0.71±0.0	0.77±0.04	0.77±0.02	0.75±0.01	0.41±0.03	0.73±0.01
cora	0.75	0.64	-	-	-	-	(+1 noise)	0.3±0.0	0.3±0.0	0.61±0.0	0.59±0.0	0.6±0.0	0.65±0.02	0.66±0.07	0.75±0.01	0.35±0.07	0.65±0.0
							(+2 noise)	0.3±0.0	0.3±0.0	0.54±0.01	0.52±0.01	0.53±0.0	0.7±0.03	0.72±0.01	0.75±0.01	0.36±0.05	0.62±0.01
								0.31±0.01	0.31±0.01	0.67±0.0	0.67±0.01	0.67±0.01	0.68±0.01	0.68±0.01	0.55±0.02	0.39±0.08	0.61±0.01
citeseer	0.56	0.65	-	-	-	-	(+1 noise)	0.21±0.0	0.21±0.0	0.57±0.0	0.56±0.0	0.57±0.01	0.62±0.03	0.66±0.03	0.55±0.03	0.35±0.07	0.54±0.01
							(+2 noise)	0.21±0.0	0.21±0.0	0.51±0.01	0.48±0.01	0.5±0.01	0.66±0.04	0.64±0.01	0.55±0.03	0.36±0.13	0.52±0.01
								0.16±0.01	0.16±0.01	0.75±0.05	0.68±0.06	0.7±0.05	0.67±0.05	0.8±0.04	0.34±0.04	0.25±0.06	0.69±0.05
dkpol	0.35	0.16	0.69	-	-	-	(+1 noise)	0.14±0.0	0.14±0.0	0.65±0.04	0.61±0.02	0.63±0.03	0.5±0.06	0.71±0.09	0.36±0.05	0.26±0.05	0.39±0.04
							(+2 noise)	0.14±0.0	0.14±0.0	0.6±0.02	0.56±0.01	0.57±0.02	0.52±0.07	0.73±0.07	0.36±0.05	0.19±0.04	0.32±0.05
								0.3±0.01	0.3±0.01	0.85±0.04	0.8±0.06	0.85±0.05	0.84±0.04	0.85±0.04	0.75±0.07	0.55±0.04	0.81±0.05
aucs	0.34	0.36	0.61	0.8	0.72	-	(+1 noise)	0.28±0.01	0.28±0.01	0.82±0.05	0.44±0.02	0.73±0.04	0.82±0.05	0.82±0.05	0.76±0.09	0.48±0.06	0.78±0.05
							(+2 noise)	0.27±0.0	0.27±0.0	0.78±0.04	0.32±0.05	0.57±0.03	0.78±0.04	0.78±0.04	0.76±0.09	0.54±0.05	0.78±0.07
APR								0.36	0.36	0.93	0.83	0.89	0.95	**0.97**	0.87	0.57	0.89
AR								9.4	9.4	3.1	6.1	4.7	**2.6**	2.7	4.9	7.8	4.3

E. Learned Parameters

In Table 11, we report the different parameters learned by the methods on the real datasets in Subsection 5.2. The numbers are averaged over three random samplings of the initially labeled nodes. We emphasize that:

- SGMI always assigns all the weight to one single layer, which may change when the initial labels change;
- SMACD computes a coupled matrix-tensor nonnegative factorization. The parameters shown here are the norm of the rows of the coupling kernel in the computed factorization (averaged over the 3 runs);
- GMM has only one parameter, and it is always fixed a-priori to $p = -1$;
- BINOM learns different parameters for different classes (which are denoted as B_1 B_2 B_3 and so forth).

Table 11. Different parameters learned by the methods on real datasets. The numbers are averaged over three random samplings of the initially labeled nodes.

Dataset		B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	MULTI	SGMI	SMACD
3sources	$\boldsymbol{\beta}_1$	0.39	0.59	0.33	0.6	0.59	0.33					0.6	0.6	0.32
	$\boldsymbol{\beta}_2$	0.21	0.19	0.33	0.2	0.22	0.33					0.2	0.4	0.33
	$\boldsymbol{\beta}_3$	0.39	0.22	0.33	0.19	0.19	0.33					0.2	0	0.35
	α	0.14	0.14	1	8.1	0.17	1					-17.94		
	λ	4.82	4.76	1	0.89	4.81	1					0.1		
BBC	$\boldsymbol{\beta}_1$	0.23	0.17	0.26	0.24	0.27						0.16	0.2	0.25
	$\boldsymbol{\beta}_2$	0.21	0.17	0.18	0.26	0.23						0.54	0.4	0.25
	$\boldsymbol{\beta}_3$	0.25	0.12	0.19	0.15	0.15						0.15	0.2	0.24
	$\boldsymbol{\beta}_4$	0.3	0.54	0.37	0.35	0.35						0.15	0.2	0.26
	α	2.87	-3.69	0.42	-7.88	0.39						0.13		
	λ	1.66	3.58	1.77	4.33	1.75						0.68		
BBCSport	$\boldsymbol{\beta}_1$	0.57	0.46	0.6	0.45	0.42						0.44	0.8	0.48
	$\boldsymbol{\beta}_2$	0.43	0.54	0.4	0.55	0.58						0.56	0.2	0.52
	α	4.21	0.27	-4	0.31	0.16						0.08		
	λ	3.75	2.59	10	2.58	2.43						2.22		
Wikipedia	$\boldsymbol{\beta}_1$	0.32	0.34	0.27	0.35	0.32	0.21	0.27	0.42	0.27	0.07	0.2	0	0.17
	$\boldsymbol{\beta}_2$	0.68	0.66	0.73	0.65	0.68	0.79	0.73	0.58	0.73	0.93	0.8	1	0.83
	α	6.62	0.18	-3.78	0.46	8.8	-7.85	-3.83	2.48	4.16	8.15	0.13		
	λ	8.38	4.71	5.3	4.78	7.23	6.48	5.27	5.41	5.9	8.88	6.45		
UCI	$\boldsymbol{\beta}_1$	0.02	0.09	0.06	0.06	0.14	0.09	0.21	0.09	0.09	0	0.16	0	0.17
	$\boldsymbol{\beta}_2$	0.43	0.1	0.27	0.07	0.11	0.27	0	0.08	0.51	0	0.16	0	0.17
	$\boldsymbol{\beta}_3$	0	0.26	0.29	0.23	0.12	0.31	0.4	0.08	0.07	0.4	0.17	0.2	0.18
	$\boldsymbol{\beta}_4$	0.23	0.09	0.05	0.06	0.12	0.09	0	0.09	0.09	0	0.16	0	0.13
	$\boldsymbol{\beta}_5$	0.09	0.37	0.29	0.52	0.39	0.14	0.4	0.58	0.15	0.6	0.2	0.8	0.18
	$\boldsymbol{\beta}_6$	0.23	0.09	0.05	0.06	0.11	0.09	0	0.08	0.09	0	0.16	0	0.17
	α	10.05	2.28	4.21	-9.55	6.36	0.26	-3.51	2.31	2.35	-12	0.22		
	λ	8.04	4.46	6.52	4.34	2.79	4.77	8.19	4.59	4.47	10	1.03		
cora	$\boldsymbol{\beta}_1$	0.73	0.93	0.75	0.73	0.83	0.73	0.74				0.71	1	0.07
	$\boldsymbol{\beta}_2$	0.27	0.07	0.25	0.27	0.17	0.27	0.26				0.29	0	0.93
	α	1.05	-7.73	1.95	0.71	-3.6	0.66	1.45				0.28		
	λ	2.31	5.67	2.69	2.32	3.88	2.18	2.58				0.91		
citeseer	$\boldsymbol{\beta}_1$	0.29	0.5	0.44	0.46	0.54	0.43					0.67	1	0.31
	$\boldsymbol{\beta}_2$	0.71	0.5	0.56	0.54	0.46	0.57					0.33	0	0.69
	α	8.03	1.87	3.11	2.17	3.27	3.24					0.49		
	λ	4.92	2.04	2.69	2.13	2.5	2.66					1.09		
dkpol	$\boldsymbol{\beta}_1$	0.2	0.15	0.26	0.03	0.24	0.19	0.24	0.19	0.22	0.2	0.36	1	0.03
	$\boldsymbol{\beta}_2$	0.24	0.2	0.23	0.14	0.35	0.23	0.35	0.19	0.24	0.26	0.32	0	0.01
	$\boldsymbol{\beta}_3$	0.57	0.66	0.51	0.82	0.42	0.58	0.42	0.62	0.54	0.54	0.32	0	0.96
	α	6.35	8.82	5.29	12.55	0.78	6.77	0.62	9.29	-2.21	5.63	0.26		
	λ	4.3	5.45	4.65	7.77	2.24	4.39	2.31	5.67	2.22	3.94	0.97		
aucs	$\boldsymbol{\beta}_1$	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17		0.21	0	0.03
	$\boldsymbol{\beta}_2$	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19		0.15	0	0.18
	$\boldsymbol{\beta}_3$	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13		0.1	0	0.16
	$\boldsymbol{\beta}_4$	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28		0.42	0.8	0.31
	$\boldsymbol{\beta}_5$	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23		0.12	0.2	0.31
	α	1.38	1.38	1.38	1.38	1.38	1.38	1.38	1.38	1.38		1.05		
	λ	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7		2.56		