Computational Statistics and Data Analysis 139 (2019) 99–116

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Fusing data depth with complex networks: Community detection with prior information

Yahui Tian, Yulia R. Gel [*,1]

Department of Mathematical Sciences, University of Texas in Dallas, USA

ARTICLE INFO

Article history:
Received 27 December 2017
Received in revised form 17 September 2018
Accepted 11 January 2019
Available online 16 May 2019

Keywords:
Community detection
Sparse complex networks
Data depth
Outliers

ABSTRACT

A new nonparametric supervised algorithm is proposed for detecting multiple communities in complex networks using the Depth vs. Depth (DD(G)) classifier. The key idea behind the new clustering method is the notion of robust and data-driven data depth methodology that still remains new and unexplored in network sciences. The developed new DD(G)-method is inherently geometric and allows to simultaneously account for network communities and outliers. Although the data-based classifier operates within a supervised learning framework, the related nonparametric notion of depth in networks can be used in a more general context, including (semi) supervised and unsupervised learning. Utility of the new approach is illustrated by using the benchmark political blogs data, "dark" terrorist networks, and analysis of bill cosponsorship in the Italian Parliament.

1. Introduction

Many real-world networks show the presence of communities, i.e., the phenomena where certain network features tend to cluster into local compact groups. Identification of network communities has a broad range of applications, from customer segmentation to discovering micro ecosystems in food webs to detecting gang formation and terrorist activity in crime networks. The problem of identifying communities has been extensively studied in the literature and nowadays still remains one of the most active research areas in network analysis (for overview of algorithms see, e.g., Fortunato, 2010; Goldenberg et al., 2010; Scott and Carrington, 2011; Plantié and Crampes, 2013; Li and Zhang, 2013; Harenberg et al., 2014; Wilson et al., 2014; Li et al., 2015; Estrada and Knight, 2015, and references therein).

Although a notion of *network community* is not defined uniquely and varies among domains of network applications (Leskovec et al., 2010; Estrada and Knight, 2015), conventionally a community is thought of as a cohesive set of vertices that are more strongly or better connected with each other than with external vertices. That is, community membership can be characterized in terms of how "close", or "central" a given vertex is to other members with respect to a particular attribute. Moreover, there exists no "gold-standard" rule that allows to reliably examine results of community detection, and in practice detected communities are often validated using non-comprehensive anecdotal procedures (Yang and Leskovec, 2012; Newman and Clauset, 2016). The problem is further exacerbated by presence of (usually multiple) outliers. While outliers are long known to have a considerable impact on community detection algorithms (Aggarwal, 2013), until very recently the two naturally tightly-knit problems of network clustering and outlier detection have been

* Correspondence to: 800 West Campbell Road, Richardson, TX 75080, USA.
E-mail address: ygl@utdallas.edu (Y.R. Gel).

[1] Supported in part by the National Science Foundation grants IIS 1633331 and DMS 1736368.

https://doi.org/10.1016/j.csda.2019.01.007
0167-9473/

mainly studied as independent problems (Perozzi et al., 2014; Cai and Li, 2015). This is partially due to the fact that the underlying probabilistic geometry of the network data is usually not taken into account explicitly in the community detection process. At the same time, many popular methods for community detection, for example, various versions of spectral clustering based on the K-means algorithm and model-based clustering, are sensitive to outliers, especially since a single graph can contain multiple different groups of outliers or anomalies (Gao et al., 2010).

To address the above challenges, we propose to introduce a concept of *data depth* into the network community detection that allows to integrate and systematize ideas on centrality, community, and outliers; and to the best of our knowledge, the notion of data depth is yet a new and unexplored tool for analysis of communities in complex networks. Data depth is a nonparametric and inherently geometric method that was introduced initially in a setting of multivariate data analysis. The idea is to assign a numeric value to each data point in respect to its centrality within a given data cloud. Depth contours based on such natural center-outward orderings of all data points can be then used to visualize and simultaneously evaluate clusters as well as outlyingness and anomaly structures, and thus to more efficiently recover latent mechanisms behind data structures. Data depth is a quickly developing field that gains increasing momentum in view of wide applicability of depth concepts to classification, data visualization, high dimensional and functional data analysis (for overview see, e.g., Zuo and Serfling, 2000; Cuevas et al., 2007; López-Pintado and Romo, 2009; Hyndman and Shang, 2010; Nieto-Reyes and Battey, 2016, and references therein). Given the proven power of depth function methodology with multivariate and functional data, it is highly appealing to extend these ideas to the complex setting of network data. Indeed, one such contribution is provided by Fraiman et al. (2015), but it concerns a random sample of graphs following a probability model on the space of all graphs of a given size, and more recently Tian and Gel (2017) show utility of depth-based (dis)similarity measure based on L_1-depth, for unsupervised network clustering. In the present treatment as well as in many applied studies, however, we deal with a data set consisting of a *single network* with a small set of vertices with known membership, and the goals of detecting the network community structure and identifying its center, outliers, and other structural features.

Our key idea is to adapt the so-called data depth vs. depth classifier, or DD(G)-classifier, as a primary supervised community recovery method within a spectral clustering framework. The DD(G)-classifier is a completely nonparametric and data-driven procedure (Li et al., 2012; Cuesta-Albertos et al., 2017) which first displays the depth values of sample objects with respect to the underlying clusters and then automatically determines the best separating curves to classify objects by their depth representation. In addition, for a case of two clusters, the DD(G)-classifier allows a comprehensive visualization of data structures as a two-dimensional scatterplot (Li et al., 2012). Recently, the depth vs. depth nonparametric classification has been systematically enhanced to a general multi-class case in functional data analysis (Mosler and Mozharovskyi, 2015). The proposed DD(G)-classifier for network community detection also relies on estimation of the optimal separating curve and, hence, operates within a supervised learning framework. While clearly knowledge of some prior network information might be considered as a restriction, such setting has practical appeal, especially in social networks. However, supervised network clustering yet appears to remain an under-explored area. Indeed, most of the current methods for network community recovery fall either into an unsupervised framework (arguably, the dominant fraction of (at least) statistical studies on complex networks) or into a semi-supervised framework (i.e., the research direction mainly explored in computer science community and, to a much less extent, in statistics). In many applications, however, particularly, on analysis of various *dark* (i.e., criminal, terrorist and illicit) networks, there often exists some prior knowledge, and the primary interest is to cluster the remaining network data, given this prior set of labeled training data (Scott and Carrington, 2011; Everton, 2012; Campbell et al., 2013). Furthermore, the idea of the DD(G)-classifier can be intrinsically integrated as a part of semi-supervised network clustering, enhancing, for instance, choice of similarity measures and reducing the impact of anomalies. Finally, the notion of data depth which is a new concept in network studies, can be used in a much more general context of analysis, description and visualization of complex network structures.

The main contributions of our study are summarized as follows:

- We bring the concept of *data depth* into the complex network analysis, with a particular focus on a case when only a *single realization* of a network is available. Although data depths are widely used in multivariate and functional data analysis, the depth concept yet remains an unexplored tool in network sciences.
- We develop a novel nonparametric and data-driven procedure for community detection in complex networks, based on the inherently geometric tools of *data depth* and *depth–depth (DD(G)) classifier*. To the best of our knowledge, this is the first work that employs the data depth for analysis of communities in complex networks in a supervised setting.
- We investigate the impact of regularization in conjunction with a data depth framework. Echoing the recent results on regularized spectral clustering of Chaudhuri et al. (2012), Joseph and Yu (2016), we find that the best performance of the new DD(G)-classifier in stochastically block models (SBM) is also achieved under regularization.
- We validate the effectiveness of the new data depth classification procedure on a wide range of synthetic networks with varying degree of sparseness, number of communities and outliers. We illustrate utility of the new community detection method for tracking terrorist groups, organization of political blogs, and the bill cosponsorship in the Italian Parliament.

The paper is organized as follows. In Section 2, we introduce basic notations and review a concept of data depth. Section 3 proposes the new nonparametric community detection algorithm based on DD(G)-classifiers and provides an insight on its theoretical properties of the new method. Simulations studies are presented in Section 4. Section 5 illustrates application of the new DD(G)-classifier to detecting communities in terrorist networks, political blogs and bill cosponsorship in the Italian Parliament. The paper is concluded by discussion in Section 6.

2. Background on graphs and data depth

Graph preliminaries. Consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a vertex set $\mathcal{V}$ and edge set $\mathcal{E}$. Let n be cardinality of $\mathcal{V}$, that is, order of $\mathcal{G}$. We assume if edge $e_{uv} \in \mathcal{E}$, then $u \neq v$.

Let A be the corresponding $n \times n$ symmetric adjacency matrix, i.e., $A_{ij} = 1$ if there is an edge between vertices i and j, otherwise $A_{ij} = 0$. Let Q be the diagonal matrix of degrees, i.e., $Q_{ii} = \sum_{j=1}^{n} A_{ij}$. A graph Laplacian is defined as

$$L = Q^{-1/2} A Q^{-1/2}. \tag{1}$$

As network density decreases, number of low-degree vertices increases, which in turn results in zero eigenvalues of L and elevates variability of the Laplacian estimators, including but not limited to adversely impacting a community extraction task (see Rohe et al., 2011; Chaudhuri et al., 2012; Amini et al., 2013; Joseph and Yu, 2016; Le and Vershynin, 2015, and references therein). Hence, following Amini et al. (2013) and Joseph and Yu (2016), we also consider a *regularized Laplacian* in the following form:

$$L_\tau = Q_\tau^{-1/2} A_\tau Q_\tau^{-1/2}, \tag{2}$$

where τ is a regularizer; $A_\tau = A + \tau J, J = \frac{1}{n}\mathbf{1}$, $\mathbf{1}$ is a $n \times n$ matrix with all elements 1, and $Q_{ii,\tau} = \sum_{j=1}^{n} A_{ij} + \tau$. Joseph and Yu (2016) suggest to select an optimal regularizer by minimizing the Davis–Kahan bound, i.e. the bound on the distance between the sample and population Laplacians. The method of Joseph and Yu (2016) is called *DKest*, and throughout our study, whenever regularization is considered, we adopt DKest to select optimal τ. (We use the Quasi-Newton method to tackle the minimization in DKest.)

One of the most popular methods for network community detection is based on spectral clustering (SC), with an idea of embedding a graph $\mathcal{G}$ into a collection of multivariate sample points. Given a number of communities K, we identify orthogonal eigenvectors $\mathbf{x}_{\cdot j},\ j = 1, \ldots, K$ of the Laplacian L (or its regularized counterpart L_τ) corresponding to the largest K eigenvalues, and construct an $n \times K$-matrix $X = [\mathbf{x}_{\cdot 1}, \ldots, \mathbf{x}_{\cdot K}]$. Each row of X, $\mathbf{x}_i \equiv \mathbf{x}_{i\cdot}$, provides a representation in $\mathbb{R}^K$ of a vertex in $\mathcal{V}$, and thus we obtain n sample points in $\mathbb{R}^K$. We can now cluster this multivariate dataset into K communities using any appropriate classifier — that is, with a conventional method of K-means or we can follow an *alternative route* and augment spectral clustering with a *data depth* methodology on $\mathbb{R}^K$.

Why Data Depth? In the last two decades, a concept of data depth is shown to be an attractive nonparametric tool to analyze, classify and visualize multivariate data without making prior assumptions about underlying probability distributions. A new impetus has been recently given to a data depth methodology due to its multifaceted utility in high dimensional and functional data analysis. However, a concept of data depth remains yet unexplored in analysis of complex networks.

Given a notion of data depth, we can measure the "depth" (or "outlyingness") of a given object or a set of objects with respect to an observed data cloud. A higher value of a data depth implies a deeper location, or higher centrality in the data cloud. By plotting such a natural center-outward ordering of depth values that serves as a topological map of the data, the presence of clusters, outliers and anomalies can be evaluated simultaneously in a quick and visual manner.

Definition. A data depth is a function that measures how closely an observed point $x \in \mathbb{R}^d$, $d \geq 2$, is located to the "center" of a finite set $\mathcal{X} \in \mathbb{R}^d$, or relative to F, a probability distribution in $\mathbb{R}^d$. As outlined by Zuo and Serfling (2000), a data depth satisfies the following desirable properties: affine invariant; upper semi-continuous in x; quasiconcave in x (i.e., having convex upper level sets); vanishing as $\|x\| \to \infty$.

Since different notions of depth functions vary in their computational complexity, robustness and sensitivity to certain underlying distributions (Liu et al., 1999; Zuo and Serfling, 2000; Hubert et al., 2008), we consider the following four non-negative and bounded depth functions as our main tools:

- **Tukey (TD) depth**

  $$HS_F(x) = inf_H\{P_F(H)\},$$

  where H is a closed half-space in $\mathbb{R}^d$ and $x \in H$, measures the *tailedness* of the point with respect to the deepest point of the distribution F. The sample version $HS_{F_m}(x)$ is obtained by replacing F in $HS_F(x)$ by the empirical distribution F_m. For moderate d the Tukey depth can be estimated exactly using computationally efficient algorithms of Dyckerhoff and Mozharovskyi (2016a). For higher dimensions the classical Tukey depth can be approximated by the Random Tukey depth (RTD) (Cuesta-Albertos and Nieto-Reyes, 2008).

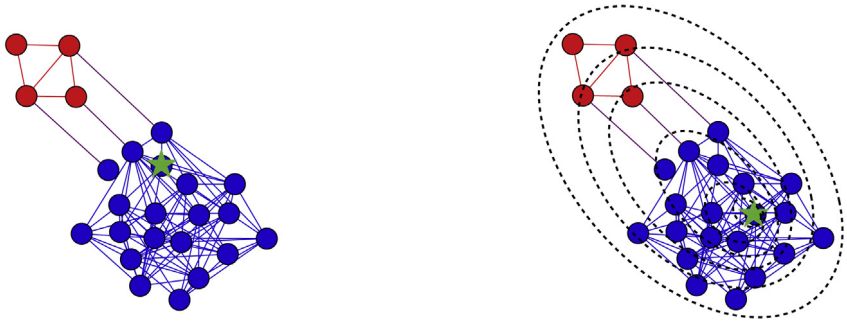

Fig. 1. Toy example: clustering with the presence of outliers (in red). Colors represent ground truth labels, green stars represent community centers yielded by K-means (left panel) and depth-based clustering (right panel). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- **Mahalanobis (MhD) depth**

$$MhD_F = [1 + (x - \mu_F)' \Sigma_F^{-1} (x - \mu_F)]^{-1},$$

where μ_F and Σ_F are the mean vector and covariance matrix of F, respectively. The sample version $MhD_{F_m}(x)$ is obtained by replacing μ_F and Σ_F with their sample estimates. The Mahalanobis depth measures the *outlyingness* of the point with respect to the deepest point of the distribution, and allows to easily handle the elliptical family of distributions, including a Gaussian case. However, the Mahalanobis depth is less robust and fails to distinguish any two distributions with the same first two moments.

- **Random Projection (RPD) depth**

$$RPD_F(x) = \max_{u \in \mathbb{U}} |u'x - Med(F_u)|/MAD(F_u),$$

where F_u is the distribution of $u'x$, Med(F_u) is the median of F_u, $MAD(F)_u$ is the median absolute deviation of F_u. Finally, $\mathbb{U} = \{u_1, u_2, \ldots, u_k\}$, where u_i is a randomly selected projection from d-dimensional hypersphere. The sample version $RPD_{F_m}(x)$ is obtained by replacing the median and MAD with their sample estimates. Random projection depth is the stochastic approximation of projection depth which also measures the *outlyingness* of the point with respect to the deepest point of the distribution. RPD is robust against possible extreme observations.

Remark. In addition, we evaluated simplicial, spatial and zonoid data depths. However, we find that their finite sample clustering performance is less competitive, and thus we omit these depths from further analysis.

Besides their role for descriptive analysis, depth-based approaches are also being applied for *classification* and *clustering* problems, as well as for visualization of the detected clusters, in high-dimensional and functional data studies, often in conjunction with microarray gene expression and other biomedical applications (see Jörnsten, 2004; Li et al., 2012; Mosler and Mozharovskyi, 2015; Cuesta-Albertos et al., 2017, and references therein). Choice of the most feasible depth function remains an open problem and largely depends on the underlying probabilistic geometry of the data. Since there exists no universally optimal depth, selection of depth function is typically dictated by particular desired properties that are required in a given study, such as robustness, behavior of the depth outside of the convex support, computational speed etc. Overall, depth-based approaches provide two main advantages: they are more robust against outliers and are intrinsically geometric.

Fig. 1 illustrates this idea. Clustering results of K-means (see the left panel) is affected by a small group of outliers, i.e., the estimated cloud center (denoted by a green star) is forced to pull toward outliers because of the conventional measure of distance. In contrast, depth-based approaches to clustering and classification are based on a distance between a point and the remaining data cloud (i.e., not just between a pair of points), thus reducing the impact of outliers and delivering more robust results. The right panel of Fig. 1 depicts a contour plot based on depth values, and the vertex with the highest depth value is assigned to be the center (denoted by a green star). Visually, the center found by data depth is closer to the true center than the one found by K-means (left panel).

What additional insight does the depth-based approach to classification allow us to gain? In short, the core idea behind the data depth is to better capture a *probabilistic geometry* of the underlying data, that is, the position of each observation in respect to the whole data cloud of interest and to quantify a probability that a certain point belongs to the community, given a geometric structure of this community. Then, how to extend the data depth idea to networks?

3. A depth vs. depth (DD(G)) classifier for network community detection

In this section, we introduce a new supervised algorithm for network community detection using the data depth concept. In particular, we adapt the data depth vs. data depth, DD(G), classifier approach of Li et al. (2012) and Cuesta-Albertos et al. (2017) to a case multi-class extraction in complex graph-structured data. Moreover, for a case of two

communities, the DD(G) classifier reduces to a data depth vs. data depth (DD) plot, which allows to visualize the evaluated clusters along with various types of anomalies and outliers (Li et al., 2012). The new depth-based method for network community extraction inherently accounts for the underlying probabilistic geometry of the spectrally embedded graph-structured data.

In particular, consider an undirected graph $\mathcal{G}$ with the Laplacian L or its regularized counterpart L_τ, and let K be a known number of communities. Then, spectral embedding of $\mathcal{G}$ yields an $n \times K$-matrix $X = [\mathbf{x}_{\cdot 1}, \ldots, \mathbf{x}_{\cdot K}]$ such that each row of X, $\mathbf{x}_i, i = 1, \ldots, n$ corresponds to a vertex in $\mathcal{V}$. Let $\mathcal{X}$, $\mathcal{X} \in \mathbb{R}^K$, be a sample space formed by $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Let $F_k, k = 1, \ldots, K$ be the underlying distributions of $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{n_k}}$, where vertices $v_{i_1}, \ldots, v_{i_{n_k}}$ belong to the kth community of cardinality n_k, and let f_k, $k = 1, \ldots, K$ be the respective density functions.

Let D be a selected data depth function and $D_{F_i}(\mathbf{x})$, $\mathbf{x} \in \mathcal{X}$, be a data depth value of $\mathbf{x}$ in respect to an ith cluster, $i = 1, \ldots, K$. Now we consider a map $\mathbf{x} \to \mathbf{d} = (D_{F_1}(\mathbf{x}), \ldots, D_{F_K}(\mathbf{x}))'$ and assign a vertex v to the ith community if its corresponding representation $\mathbf{x}$ in $\mathbb{R}^K$ satisfies

$$D_{F_i}(\mathbf{x}) > r(D_{F_j}(\mathbf{x})),\ j = 1, \ldots, K,\ i \neq j, \tag{3}$$

where r is a separating function for the DD(G)-classifier. In the case of two clusters, Li et al. (2012) consider a separating function from the polynomial family. In a general multi-class case of $K \geq 2$, Cuesta-Albertos et al. (2017) propose to employ methods such as Generalized Additive Models (GAM), logistic regression, classification trees, k-nearest neighbor (kNN) etc. Clearly, choice of the optimal separating function r_0 requires existence of a (small) training set with known membership labels, and hence, the suggested classification approach belongs to a class of supervised methods. The proposed DD(G) method for community detection is outlined in Algorithm 1.

Algorithm 1: Spectral Clustering-DD(G)-Classifier Algorithm

Input : network $\mathcal{G}$; order of $\mathcal{G}$, n; number of communities K; training data set S_{Tr}, n_{Tr} is cardinality of S_{Tr}; sample version of a depth function D; family of separating functions Γ, binary choice of regularization (TRUE or FALSE).

Output: separating function r; a partition of G into K communities.

if *no regularization* **then**
  matrix $W = L$ in (1)
else
  find a regularization parameter τ with the *DKest* procedure (Joseph and Yu, 2016);
  $W = L_\tau$ in (2);
end
construct X by combining the leading K eigenvectors of W;
define $\mathbf{x}_i \equiv \mathbf{x}_{i\cdot}$ as the i-th row of X;
for $i = 1, \ldots, n_{Tr}$ **do** `// In a training dataset`
  construct a map $\mathbf{x}_i \to \mathbf{d} = (D_1(\mathbf{x}_i), \ldots, D_K(\mathbf{x}_i))'$;
end
find optimal separating curve $r \in \Gamma$ to minimize a misclassification rate;
for $j = 1, \ldots, n - n_{Tr}$ **do** `// In the remaining dataset`
  assign $\mathbf{x}_j$ to a cluster with the highest posterior probability given $\mathbf{x}_j$
end

Theoretical properties of the DD-classifiers for multivariate data have been studied by Li et al. (2012), who show that the DD-classifiers converge to the optimal separating function point-wisely. Moreover, in a case of two clusters, Li et al. (2012) show that the DD-classifier can achieve the Bayes error for a family of elliptical distributions, and showed empirically that the DD-classifier can achieve the Bayes error for more general families.

Below we state an analogous proposition for the DD(G)-classifier in conjunction with a *stochastic block model* (SBM) (Holland et al., 1983), that is a conventional model framework for analysis of community detection algorithms. The key feature of SBM is that the probability of an edge between vertices i and j is determined solely by their group memberships. That is, let Θ be a $n \times K$-membership matrix, where $\Theta_{i,k} = I_{\{k=g_i\}}$, $i = 1, \ldots, n$, $k = 1, \ldots, K$, and g_i is the block membership for vertex i. Let $B = \{B_{lm}\}_{l,m=1}^K$ be a symmetric $K \times K$-block probability matrix, where $B_{lm} \in [0, 1]$. Then, the probability of an edge between vertices i and j is $P_{ij} = B_{g_i g_j}$ for $i, j = 1, \ldots, n$, and a $n \times n$-matrix P is the population counterpart of an adjacency matrix A. Community memberships are fixed beforehand and do not overlap.

In the SBM setting, the population Laplacian $\mathcal{L}$ is then defined as

$$\mathcal{L} = \mathcal{Q}^{-1/2} P \mathcal{Q}^{-1/2}, \tag{4}$$

where $\mathcal{Q}$ is the diagonal matrix of expected degrees. Similarly to (2), regularization of $\mathcal{Q}$ and P, i.e., $\mathcal{Q}_\tau = \mathcal{Q} + \tau I$ and $P_\tau = P + \tau J$, leads to a regularized counterpart of $\mathcal{L}_\tau = \mathcal{Q}_\tau^{-1/2} P \mathcal{Q}_\tau^{-1/2}$. (Here I denotes an identity matrix.)

Proposition 1 (*Bayes Optimality*). *Let A be an adjacency matrix from a stochastic block model (Θ, B) and $K = 2$. If the following conditions are satisfied: (A1) a set of separating functions Γ satisfies the continuity assumption (see Li et al., 2012), (A2) the*

DD(G)-classifier is based on simplicial, half-space, Mahalanobis or projection depths; (A3) as the number of nodes n goes to ∞, $\sqrt{n}(\hat{\boldsymbol{\xi}}_k - \boldsymbol{\xi}_k)$ *converges to* $\mathcal{N}(0, \Sigma_{\boldsymbol{\xi}_k})$, $\Sigma_{\boldsymbol{\xi}_k} > 0$, *where* $\boldsymbol{\xi}_k$ *and* $\hat{\boldsymbol{\xi}}_k$, $k = 1, 2$ *are orthogonal eigenvectors of* $\mathcal{L}$ *and L (or their respective regularized versions) associated with the two leading eigenvalues. Then,*

$$\hat{\triangle}_n(\hat{r}_n) \xrightarrow{a.s.} \triangle(r_B) \text{ and } E(\hat{\triangle}_n(\hat{r}_n)) \to \triangle(r_B), \quad \text{as } \min_{1\le k\le K} n_k \to \infty, \tag{5}$$

where $\hat{\triangle}_n(\hat{r}_n)$ *is the empirical error from Algorithm 1,* $E(\hat{\triangle}_n(\hat{r}_n))$ *is the expected empirical error,* $\triangle(r_B)$ *is the Bayes error, and* n_1 *and* n_2 *are cardinality of the first and second communities, respectively.*

Proof. Let $X \in \mathbb{R}^{n\times 2}$ be the matrix containing the orthogonal vectors $\hat{\boldsymbol{\xi}}_1$ and $\hat{\boldsymbol{\xi}}_2$ as columns, i.e. $X = [\hat{\boldsymbol{\xi}}_1, \hat{\boldsymbol{\xi}}_2]$. For $i = 1, \ldots, n$, let $z_i \in \mathbb{R}^2$ be the vector corresponding to the ith row of X. By construction each vertex i in $\mathcal{G}$ is now represented by z_i. Let $Z_1 = (z_i, 1 \le i \le n : \Theta_{i,1} = 1)$ and $Z_2 = (z_i, 1 \le i \le n : \Theta_{i,2} = 1)$, that is, Z_1 and Z_2 are induced by vertices in $\mathcal{G}$ from the 1st and 2nd communities, respectively. Hence, assumption (A3) and Proposition 1 and Theorem 5 of Li et al. (2012) imply that (5) holds asymptotically, that is, as $\min(n_1, n_2) \to \infty$

$$\hat{\triangle}_n(\hat{r}_n(Z_1, Z_2)) \xrightarrow{a.s.} \triangle(r_B) \text{ and } E(\hat{\triangle}_n(\hat{r}_n(Z_1, Z_2))) \to \triangle(r_B).$$

Remark. Note that according to Li et al. (2012), multivariate normality is not a necessary but sufficient condition, to achieve the Bayes error.

Remark. The key challenge in Proposition 1, however, is the assumption A3 or more generally properties of the distribution of eigenvectors of L. There yet exists no formal result on asymptotic distribution of eigenvectors of L, even under the simplified conditions of the Erdös–Rényi model, and derivation of such a result remains an open fundamental problem. Assumption A3 seems to be closely linked to analogous results on asymptotic distribution of eigenvectors of covariance and, more generally, symmetric random matrices (for recent results see Bai and Silverstein, 2010; Ledoit and Péché, 2011; Fan and Wang, 2015, and references therein). Recent advances on analysis of eigen-structure of random graphs are mainly based either in a Random Dot Product Graph (RDPG) parametrization (Athreya et al., 2016; Tang and Priebe, 2016) or stochastic Kronecker graph parametrization (Radcliffe and Young, 2014). In addition, under the SBM setting, Kadavankandy et al. (2015) discuss asymptotic distribution of eigenvectors of A corresponding to its continuous spectra. Finally, Avrachenkov et al. (2015) study the limiting empirical distribution of the Laplacian eigenvalues of SBM, and these results pave some way for analysis of the asymptotic properties of eigenvectors of L.

Remark. Interestingly, while there exists no theoretical result on distribution of eigenvectors of L, it appears that the Assumption of normality is still plausible. In particular, consider a joint frequency plot of the two leading eigenvectors of L under SBM with two clusters (see Fig. 2), which suggests some approximately bell-shaped distribution. In turn, Fig. 3 shows χ^2-Quantile–Quantile (QQ) plots of the two leading eigenvectors of Laplacian, with and without regularization, left and right panels, respectively. Remarkably, as the left panel of Fig. 3 indicates, eigenvectors of regularized L_τ are substantially closer to a bivariate normal distribution (i.e., the depicted squared Mahalanobis distances more tightly cluster along the solid Quantile–Quantile (QQ) line) than their unregularized counterparts (see the right panel of Fig. 3). Although Fig. 3 shows empirical distribution of the Laplacian eigenvectors for a single realization of SBM, we find similar patterns for other realizations. These empirical findings not only support the claim of Joseph and Yu (2016) on utility of regularization in terms of improving clustering performance but also imply a potential alternative method for selecting an optimal regularizer τ or dimension d for a low-dimensional Euclidean representation of RDPG, that is, by transformations to multivariate normality (Ruppert, 2014).

4. Simulation study

In this section, we evaluate a finite sample performance of the new DD(G) nonparametric classifier for detecting network communities using the synthetic data. To simulate the network data, we consider a stochastic block model (SBM) which is a canonical model for community detection (Abbe, 2016), as well as its recent extension, namely, generalized stochastic block model (GSBM) that allows for outliers and weak communities (Cai and Li, 2015). We evaluate various scenarios of SBM, that is, varying number of communities and network sparseness.

For the new DD(G) community detection method, we consider three data depths: Random Tukey (RTD), Mahalanobis (MhD) and Random Projection (RPD). Number of projections for RTD and RPD are 10 and 50, respectively; The considered classifiers for the DD(G) method are: Logistic regression (GLM), Generalized Additive Models (GAM), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), k-Nearest Neighbor (KNN), and Nonparametric Kernel (NP). As a competing community detection algorithm, we consider a supervised method based on edgewise weighted support vector machine (edgewise-SVM) that has been recently proposed by van Laarhoven and Marchiori (2013). The edgewise-SVM is simple and computationally efficient procedure that assigns an edge to be "within a community" or "between communities" based on its local features. A score (feature) is then assigned to each edge, with a lower number indicating higher probability of being a "between communities" edge. The algorithm removes edges with lowest score iteratively,

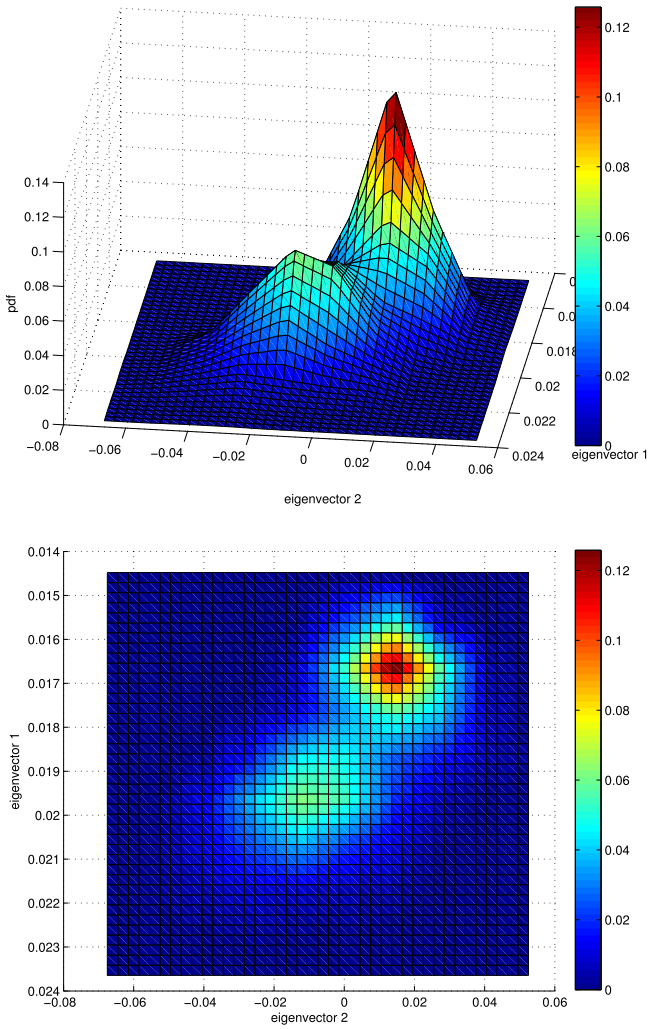

Fig. 2. Frequency plot (left panel) and contour plot (right panel) of the two leading eigenvectors of the graph Laplacian, under a single realization of a Stochastic Block Model (6).

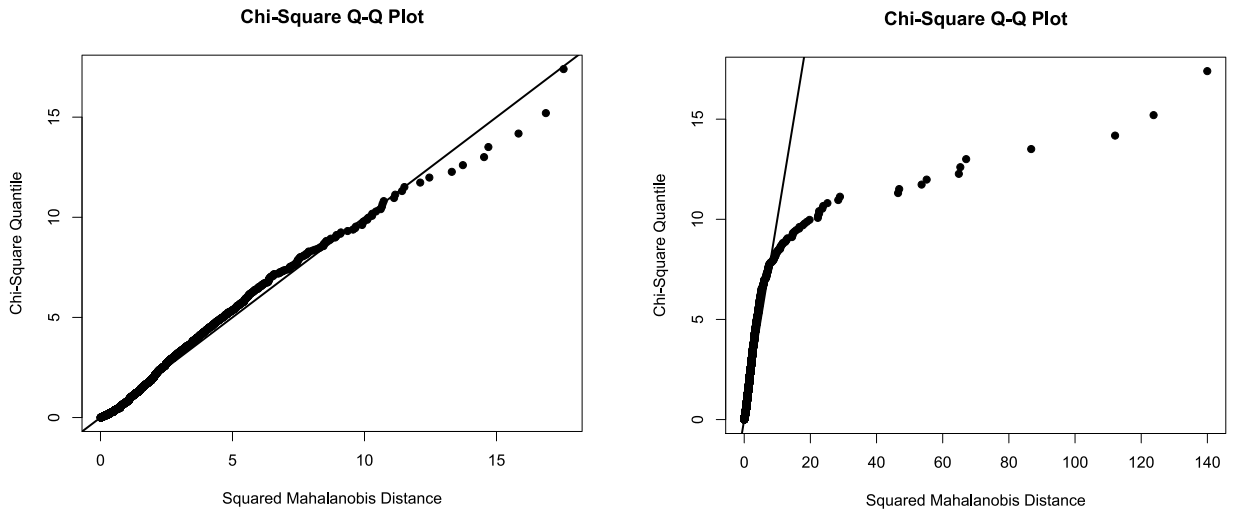

Fig. 3. The χ^2-Quantile–Quantile (QQ) plots of the two leading eigenvectors of the graph Laplacian, under a single realization of a Stochastic Block Model (6). Left and right panels correspond to regularized and unregularized Laplacians, i.e. L_τ and L, respectively.

which leads to a reduced subgraph containing only "within a community" edges, that is, clusters of vertices. In addition, we consider the conventional K-means method which is although an unsupervised classifier, is the most widely used algorithm for network community detection in a spectral setting, which is also our primary framework.

Remark. Note that our simulation studies primarily aim to highlight utility of the new supervised DD(G) algorithm in respect to varying number of communities, network sparsity and structure as well as choice of data depth and separating function rather than to investigate its absolute performance. This is partially due to the lack of comparable competing methods and model frameworks in a supervised setting. Indeed, note that both the edgewise-SVM and K-means are somewhat disadvantaged in the sense that the edgewise-SVM assumes no knowledge on the number of clusters K, while K-means operates in an unsupervised setting. However, to our knowledge, there exists no other, besides SBM, benchmark network model for synthetic data with a *known* number of communities. At the same time, a (relatively small) fraction of network classifiers that operate in a supervised or semi-supervised settings, are mainly evaluated using real-world case studies rather than with synthetic data (see Subbian et al. (2014) and reference therein), thus raising an issue of "ground truth" definition and existence (Yang and Leskovec, 2012; Newman and Clauset, 2016). In turn, most currently available network clustering algorithms with a known number of communities K belong to a family of unsupervised classifiers (Wilson et al., 2014).

Performance Measures We evaluate goodness of clustering using standard performance metrics such as normalized mutual information (NMI) and misclassification rate (Manning et al., 2008). Given the two sets of clusters with a total of n vertices: $\mathbb{R} = \{r_1, \ldots, r_K\}$ and $\mathbb{C} = \{c_1, \ldots, c_J\}$, the NMI is given by

$$NMI(\mathbb{R}, \mathbb{C}) = \frac{I(\mathbb{R}; \mathbb{C})}{[H(\mathbb{R}) + H(\mathbb{C})]/2}.$$

Here I is mutual information

$$\begin{aligned} I(\mathbb{R}; \mathbb{C}) &= \sum_k \sum_j P(r_k \bigcap c_j) \log \frac{P(r_k \bigcap c_j)}{P(r_k)P(c_j)} \\ &= \sum_k \sum_j \frac{|r_k \bigcap c_j|}{n} \log \frac{n|r_k \bigcap c_j|}{|r_k||c_j|}, \end{aligned}$$

where $P(r_k)$, $P(c_j)$, and $P(r_k \bigcap c_j)$ are the probabilities of a vertex being in cluster r_k, c_j and in the intersection of r_k and c_j, respectively; and H is entropy defined by

$$H(\mathbb{R}) = -\sum_k P(r_k) \log P(r_k) = -\sum \frac{|r_k|}{n} \log \frac{|r_k|}{n}.$$

NMI takes values between 0 and 1, and we prefer a clustering partition with a higher NMI.

We define misclassification rate as the total percentage of mislabeled vertices, that is

$$\gamma = \frac{1}{n} \sum_{k=1}^{K} S_i,$$

where S_k is the number of misclassified vertices in the kth community, i.e. $S_k = \sum_{i \in C_k} I(\hat{g}_i = k | g_i \neq k)$ and $C_k = \{1 \leq i \leq n : g_i = k\}$ is a set of labels in a kth community of cardinality n_k.

Remark. An alternative performance measure of community recovery is modularity. Modularity is based on a comparison of a fraction of edges within the estimated communities with a random distribution of edges without any community structure. For practical purposes, modularity is typically more preferable since this measure does not require the knowledge of true membership. However, modularity suffers a resolution limit and is unable to capture dense structure from small communities (Fortunato and Barthelemy, 2007; Kumpula et al., 2007). In turn, NMI and misclassification rates assess accuracy of community recovery from a different perspective, that is, the focus is on a difference between estimated and true memberships. Since our primary focus is on supervised learning (i.e., true membership is given) and the considered data contain a number of small communities (see, for example, Fig. 8 Bill Co-sponsorship in the Italian Parliament), NMI and misclassification rate are chosen as the primary evaluation metrics of accuracy.

Computational time is assessed using statistical software R on an OS X 64 bit laptop with 1.4 GHz Intel Core i5 processor and 4 GB 1600 MHz DDR3 memory. We use R package **fda.usc** (Bande et al., 2016). Code for the DD(G) community detection method is available from https://github.com/DDNetworkCommunityDetection.

Table 1
Average NMI and standard deviation in () under SBM (6). Number of Monte Carlo simulations is 100.

τ		0			DKest optimal		
Depth		RTD	MhD	RPD	RTD	MhD	RPD
$DD(G)$-Classifier	GLM	0.5037 (0.0035)	0.6299 (0.0007)	0.6086 (0.0012)	0.5975 (0.0006)	0.7396 (0.0004)	0.6113 (0.0039)
	GAM	0.5139 (0.0039)	0.6423 (0.0006)	0.6357 (0.0009)	0.6193 (0.0006)	0.7402 (0.0003)	0.6970 (0.0047)
	LDA	0.5058 (0.0036)	0.6332 (0.0007)	0.5232 (0.0012)	0.5995 (0.0006)	0.7316 (0.0004)	0.5948 (0.0078)
	QDA	0.5003 (0.0042)	0.5888 (0.0011)	0.5831 (0.0018)	0.5992 (0.0008)	0.7034 (0.0008)	0.6968 (0.0043)
	KNN	0.5047 (0.0040)	0.6128 (0.0007)	0.5962 (0.0016)	0.6011 (0.0006)	0.7349 (0.0006)	0.6901 (0.0007)
	NP	0.5186 (0.0038)	0.6415 (0.0006)	0.6234 (0.0014)	0.6236 (0.0007)	0.7389 (0.0004)	0.7356 (0.0005)
Edgewise-svm				0.0072 (0.0052)			
K-means			0.0571 (0.0863)			0.40832 (0.0238)	

4.1. Network clustering with two groups

We start from a stochastic block model (SBM) with two communities that is currently the most frequent benchmark case in network community detection. In particular, we consider an SBM with a block probability matrix

$$B = \begin{bmatrix} 0.01 & 0.0025 \\ 0.0025 & 0.003 \end{bmatrix}, \tag{6}$$

which is also the case studied by Joseph and Yu (2016). We assume that the connections within kth community follow an independent Bernoulli distribution with probability B_{kk}, $k = 1, 2$. Following the discussion by Joseph and Yu (2016), we consider both conventional and regularized versions of spectral clustering. The optimal regularizer is obtained using a data-driven technique DKest based on the estimated Davis–Kahan bounds (Joseph and Yu, 2016). We generate a network of order 3000 according to (6), randomly split the data into a training set of 600 vertices (i.e., 300 vertices for each community, or 20% of the network) and the testing set of the remaining 2400 vertices. We use the training set to learn the optimal separating curve for the DD(G) classifier. All evaluation metrics for all considered clustering methods are calculated based on the testing set.

We start from evaluating the proposed DD(G) community classifier in terms of NMI. As Table 1 indicates, the new DD(G)-classifier consistently outperforms the edgewise-svm method, with the highest NMI of 0.7400 delivered by the Mahalanobis-based DD(G)-classifier with GAM vs. 0.0072 yielded by the edgewise-svm. The DD(G) method also delivers almost twice higher NMI than the regularized K-means. Most interestingly, we find two remarkable phenomena. First, while regularization generally improves performance of the DD(G)-classifier, its impact on DD(G) is noticeably less profound than on the K-means. Intuitively, this finding can be explained by intrinsic robustness of data depth functions. Second, the Mahalanobis-based DD(G)-classifier outperforms all other depth-based classifiers, which suggests that despite yet nonexistence of a formal theoretical result, we indeed might hypothesize that eigenvectors of the Laplacian follow a normal distribution. Table 1 reports standard deviation of NMI, implying a relatively similar performance of all DD(G) classifiers and the edgewise-svm method.

We now turn to misclassification rates (see Table 2). (We exclude the edgewise-svm method from this study since edgewise-svm is not provided with a number of communities K.) Similarly to the case of NMI, the DD(G)-classifier delivers consistently more accurate results, with more than three times lower misclassification rate for an optimally regularized case (i.e., 0.04 vs. 0.17 for the Mahalanobis-based DD(G)-classifier and K-means, respectively). Furthermore, while all considered community detection methods benefit from regularization, its impact on the DD(G)-classifier is overall noticeably weaker. The best performance is again delivered by the Mahalanobis-based DD(G)-classifier with GAM.

Computational complexity of the DD-classifier is primarily dictated by the choice of a data depth function. For example, a random Tukey depth is more computational efficient than a Mahalanobis depth in high dimensional settings, since a Mahalanobis depth involves computations of the inverse of covariance matrix while a random Tukey depth requires only a finite number of randomly selected projections (Cuesta-Albertos and Nieto-Reyes, 2008). Furthermore, variations in complexity of a data depth also come from different approximation algorithms (Dyckerhoff and Mozharovskyi, 2016b). For instance, complexity of computing a Tukey depth is $O(n \log n)$ using the algorithm of Rousseeuw and Ruts (1996), and no worse than $O(nd)$ using the method of Paindaveine and Šiman (2012), where d is the dimension of vectors. For example, for a case of SBM with 2 communities and 10,000 nodes, the estimated computational time for the DD-classifier is 0.28 s, 0.29 s, and 0.27 s for RTD, MhD and RPD, respectively, based on GLM. (The results are similar for RTD, MhD and RPD under LDA and QDA.) In turn, the computational time for edgewise-svm and K-means is 0.26 s and 0.19 s, respectively.

Table 2
Misclassification rates and standard deviation in () under SBM (6). Number of Monte Carlo simulations is 100.

τ		0			DKest optimal		
Depth		RTD	MhD	RPD	RTD	MhD	RPD
DD(*G*)-Classifier	GLM	0.1100 (<0.0001)	0.0718 (<0.0001)	0.0783 (<0.0001)	0.0804 (<0.0001)	0.0442 (<0.0001)	0.0773 (<0.0001)
	GAM	0.1070 (<0.0001)	0.0685 (<0.0001)	0.0703 (<0.0001)	0.0745 (<0.0001)	0.0442 (<0.0001)	0.0553 (<0.0001)
	LDA	0.1097 (<0.0001)	0.0715 (<0.0001)	0.1046 (<0.0001)	0.0800 (<0.0001)	0.0464 (<0.0001)	0.0855 (<0.0001)
	QDA	0.1115 (<0.0001)	0.0842 (<0.0001)	0.0866 (<0.0001)	0.0800 (<0.0001)	0.0528 (<0.0001)	0.0550 (<0.0001)
	KNN	0.1088 (<0.0001)	0.0731 (<0.0001)	0.1057 (<0.0001)	0.0800 (<0.0001)	0.0502 (<0.0001)	0.0754 (<0.0001)
	NP	0.1055 (<0.0001)	0.0689 (<0.0001)	0.0742 (<0.0001)	0.0734 (<0.0001)	0.0444 (<0.0001)	0.0453 (<0.0001)
K-means			0.4323 (0.0758)			0.16343 (0.0124)	

4.2. Network clustering with three groups

We now consider an SBM with three communities. That is, we set a 3×3-block probability matrix of the form

$$B = \rho_n \begin{bmatrix} \beta\omega_1 & 1 & \\ 1 & \beta\omega_2 & 1 \\ 1 & 1 & \beta\omega_3 \end{bmatrix}, \tag{7}$$

where a $1 \times K$-vector $\omega = (\omega_1, \omega_2, \omega_3)$ represents inside weights, or relative degrees of vertices within communities, and in general communities may have different inside weights ω_i, $i = 1, \ldots, K$. The parameter β is an out-in ratio that denotes the ratio of the probability of connection between vertices from different communities to the probability of connection between vertices in the same community. The scalar parameter ρ_n is a network density and is selected in such a way that the expected degree of the network is equal to a pre-specified λ. Smaller values of expected degree λ and smaller values of ρ_n imply sparser networks.

One of the primary goals of this section is to study the effect of network sparsity on the new DD(G)-classifier. Hence, we consider networks of order n of 1200 under fixed parameters ω and *oir* and varying parameter λ:

- **Model 1:** $n = 1200$, $K = 3$, $\omega = (1, 5, 5)$, $\beta = 0.7$, $\lambda = 30$;
- **Model 2:** $n = 1200$, $K = 3$, $\omega = (1, 5, 5)$, $\beta = 0.7$, $\lambda = 20$;
- **Model 3:** $n = 1200$, $K = 3$, $\omega = (1, 5, 5)$, $\beta = 0.7$, $\lambda = 10$.

We randomly select 20% of our data as training set, i.e. 80 vertices for each community and the remaining vertices as testing set. Note that among the three models, Model 3 leads to the sparsest networks, and Model 1 describes the densest networks.

Fig. 4 shows the misclassification rates (top panel) and NMI (bottom panel) for DD(G)-classifier, *K*-means and edgewise-svm under Models 1, 2 and 3. For the sake of saving space, we only show results under regularization and include only the best candidate among all data depth and separating curve versions of the DD(G) classifier. For instance, the best performance among all DD(G) classifiers under $\lambda = 20$ (Model 2) is delivered by the Mahalanobis-based DD(G)-classifier with a KNN separating curve. Fig. 4 indicates that the new DD(G) method is applicable even to highly sparse networks (such as Model 3), performance of the DD(G) method improves with increasing density, and the best results are consistently delivered by the Mahalanobis-based DD(G)-classifier with KNN or NP separators. These findings echo the assumption on multivariate normality of the Laplacian leading eigenvectors.

Fig. 5 illustrates performance of the DD(G) method in respect to varying size of training set. We only include the DD(G) classifier with the Random Tukey depth since other depth functions exhibit similar trends. As Fig. 5 shows, misclassification rates almost exponentially decrease with an increase of training set. Remarkably, even for the training set of only 10%, misclassification rate for the DD(G) method is still 100 or more times lower than for misclassification rate for the *K*-means in both the regularized and unregularized cases (i.e., 0.001 vs. 0.160 delivered by the DD(G) classifier with the Mahalanobis depth and KNN and *K*-means, respectively, under regularization; and 0.002 vs. 0.20 delivered by the DD(G) classifier with the Mahalanobis depth and LDA and *K*-means, respectively, without regularization).

Now we investigate the performance of DD(G)-classifier as a function of sample size, number of communities and percentage of training set, under SBMs (6) and (7) (see Table 3). First, let us start from assessing the impact of sample size and percentage of training set. We find that with an increase of training set from 5% to 10%, the improvement in misclassification error is from 5 to 15 times, and the accuracy gain due to the increase of a training set is higher for larger samples (5.11 vs. 15 for the Cases 1 and 2 with 2 communities, respectively, and 6.65 vs. 12 for the Cases 3 and 4

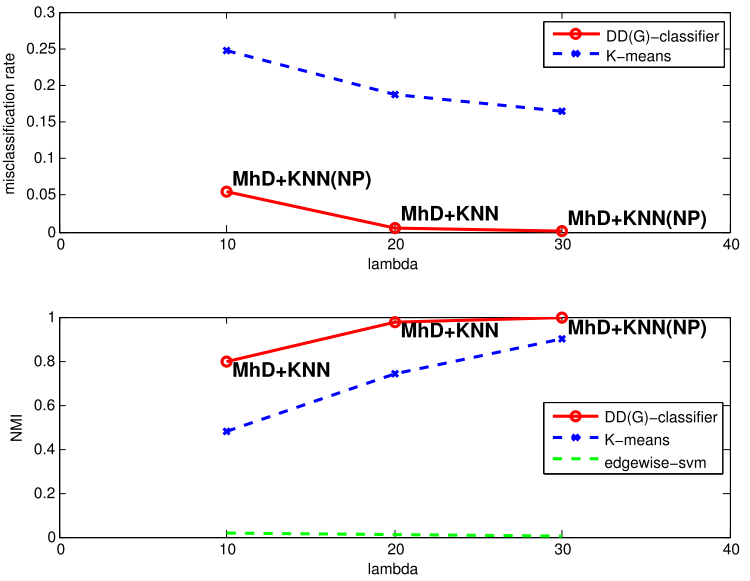

Fig. 4. Performance of DD(G)-classifier, K-means and edgewise-svm as a function of network sparsity. Smaller value of λ implies sparse networks.

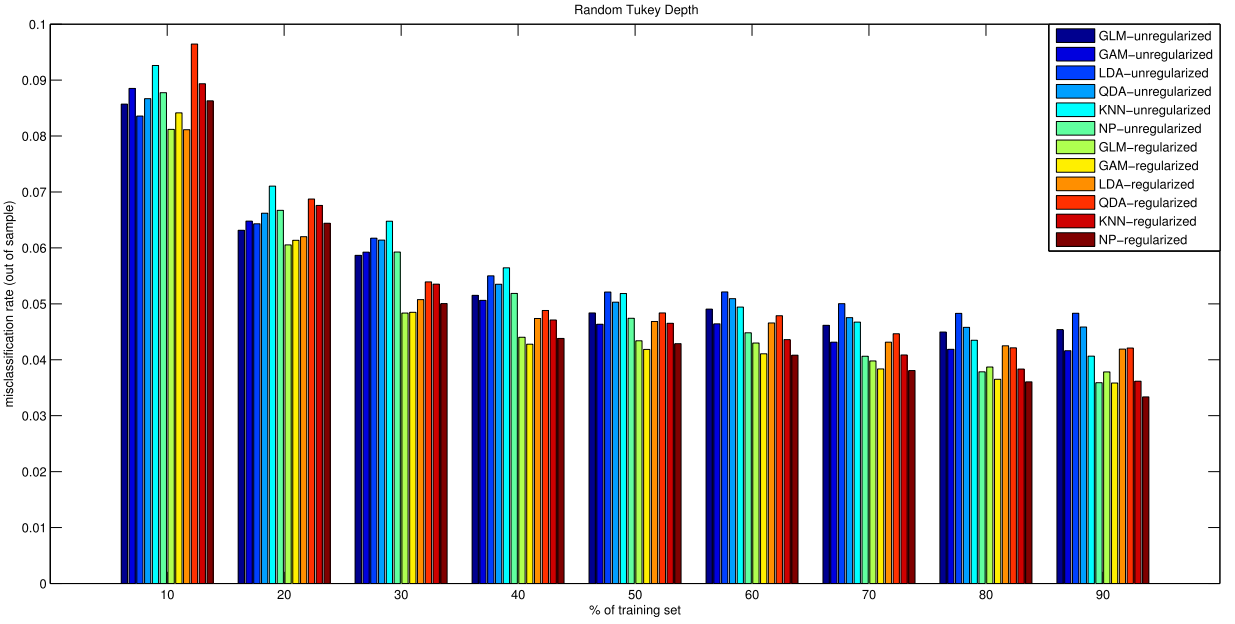

Fig. 5. Performance of DD(G)-classifier as a function of percentage of training set for Random Tukey Depth.

with 3 communities, respectively). The improvement due to an increase of training set from 10% to 20% is slightly less drastic (i.e., from 2.5 to 11 times), and again the gain in accuracy due to a higher proportion of training data is more profound for larger data set. Second, for a fixed proportion of training set, the increase in sample sizes from 100 to 200 vertices per community leads to an accuracy gain from 6 times (i.e. Cases 1 and 2 for 5% of training data) to 77 times (i.e., Cases 1 and 2 for 20% of training data). The trajectory of gain due to the available sample size for 3 communities (i.e., Cases 3 and 4) is similar but tends to be less steep. Third, as can be expected, networks with a higher number of communities consistently exhibit higher misclassification rates. However, the difference due to a number of communities for any given proportion of training data or sample size (i.e., Cases 1 and 3, and Cases 2 and 4) is substantially less than the differences for a network with a fixed number of communities but varying sample size or percentage of a training set. That is, misclassification rates for networks with 2 communities are only from 1.25 to 4.11 times lower than the respective rates for networks with 3 communities. The highest difference of 4.11 times is observed between the Cases 1 and 3 for

Table 3
Performance of DD(G)-classifier as a function of sample size, number of communities and percentage of training set for Random Tukey Depth. Case 1 is 2 communities, with 100 vertices in each community and SBM (6); Case 2 is 2 communities, with 200 vertices in each community and SBM (6); Case 3 is 3 communities, with 100 vertices in each community and SBM (7) with $\lambda = 20$; Case 4 is 3 communities, with 200 vertices in each community and SBM (7) with $\lambda = 20$. Number of Monte Carlo simulations is 100.

	Misclassification error			NMI		
	Percentage of training set					
	5%	10%	20%	5%	10%	20%
Case 1	0.0981	0.0192	0.0077	0.6865	0.9181	0.9588
Case 2	0.0165	0.0011	0.0001	0.9326	0.9916	0.9989
Case 3	0.4030	0.0606	0.0096	0.4171	0.8524	0.9590
Case 4	0.0444	0.0037	0.0004	0.8969	0.9852	0.9978

100 vertices per each community and 5% training data, but overall the ratio of misclassification rates for networks with 2 and 3 communities tends to be around to 3. Similar dynamics is observed in for NMI, although the differences in NMI rates are substantially more minor than the respective differences for misclassification rates. Hence, we can conclude that the most dominant factors in the DD(G)-classifier performance are sample size and proportion of training set, and to a substantially less extent, a number of communities.

In terms of the computational complexity, the CPU running time for the RTD, MhD and RPD-based DD(G) classifiers is 0.98 s, 0.87 s and 0.89 s, respectively. In contrast, finding an optimal regularization using the DKest requires 180 s (with additional 0.003 s for the K-means algorithm itself). The CPU running time for edgewise-svm is 0.16 s.

Overall, we can conclude that the new DD(G) method has a high potential for supervised cluster discovery for sparse and multi-community networks, even under a relatively small size of training set. We find that the DD(G) method is generally more robust to outlying vertices with low degrees, and the Mahalanobis-based DD(G) with KNN separating curve is uniformly the preferred choice across all considered network scenarios.

4.3. Network clustering with outliers

Now we evaluate performance of the DD(G) method in respect to a network with outliers. In particular, we consider a so-called *Generalized Stochastic Block Model (GSBM)* that has been recently proposed by Cai and Li (2015). GSBM is based on incorporating small and weak communities (outliers) into a conventional SBM structure. That is, we consider an undirected and loopless graph $\mathcal{G}$ with $n = n_1 + n_2$ vertices, where n_1 is the number of "inliers" which follow the standard SBM framework and n_2 is the number of "outliers" which connect with other vertices in random. Each inlier vertex is assigned to one of the two communities, while all outliers are placed into the third community.

We consider the same GSBM as the one studied by Cai and Li (2015), that is, we add 30 outliers (i.e., one small and weak community) into a standard 2-block SBM (6). We set a probability of an edge between outliers to be of 0.01. Connection between inliers and outliers is given by an arbitrary (0, 1)-matrix Z, $Z \in \mathbb{R}^{n1 \times n2}$, such that $\mathbb{E}Z = \boldsymbol{\beta}\mathbf{1}^T = [\boldsymbol{\beta}, \ldots, \boldsymbol{\beta}]$ and the component of $\boldsymbol{\beta}$ is 3000 i.i.d. copies of U^2, where U is a uniform random variable on [0, 0.0025]. We randomly choose 600 vertices from inliers and 6 vertices from outliers as a training set and then assess the misclassification rates for different DD(G)-classifiers.

Following Cai and Li (2015), we define a misclassification rate based only on inliers in the dominant 1st and 2nd communities, i.e.

$$\gamma = \frac{1}{n}\sum_{k=1}^{2} S_k,$$

where $|S_k|$ is a number of misclassified vertices in the kth community and $k = 1, 2$. Similarly, NMI is defined from calculation only on inliers and a number of clusters K is set to 3 for both K-means and DD(G) algorithms.

Table 4 and Table 5 summarize the delivered clustering performance in respect to misclassification rates and NMI, respectively. While as expected, the DD(G) method outperforms both the K-means and edgewise-svm, remarkably we find that the impact of regularization on the DD(G) method varies for different depth functions. For example, upon regularization the misclassification rate of the DD(G)-method with projection depth (RPD) and GAM decreases changes from 21.42% to 4.57%. However, performance of the Mahalanobis-based DD(G) method with GAM deteriorates upon regularization, while the Mahalanobis-based DD(G) method with KNN and NP does not exhibit any change. We also observe a similar phenomenon in terms of NMI. Hence, given that the Mahalanobis-based DD(G) method consistently shows the best performance across all other DD(G) versions and taking into account high computational costs of optimal regularizer selection, we recommend to omit regularization in the presence of weak communities.

5. Applications to real data

We now illustrate the new DD(G)-classifier algorithm for community detection in political blogs, terrorism organizations, and bill cosponsorship in the Italian Parliament.

Table 4
Misclassification rates and standard deviation (in ()) under GSBM. Number of Monte Carlo simulations is 100.

τ		0			DKest optimal		
Depth		RTD	MhD	RPD	RTD	MhD	RPD
	GLM	0.0906 (<0.0001)	0.0626 (<0.0001)	0.1114 (<0.0001)	0.0778 (<0.0001)	0.0484 (<0.0001)	0.0718 (0.0002)
	GAM	0.0774 (0.0001)	0.0427 (<0.0001)	0.2142 (<0.0001)	0.0752 (0.0001)	0.0601 (0.0034)	0.0457 (0.0427)
DD(*G*)-Classifier	LDA	0.0947 (0.0001)	0.0826 (<0.0001)	0.1054 (0.0001)	0.0803 (<0.0001)	0.0546 (<0.0001)	0.1193 (0.0004)
	QDA	0.1020 (0.0010)	0.1083 (0.0009)	0.2249 (0.0007)	0.0940 (0.0001)	0.1424 (0.0073)	0.1045 (0.0170)
	KNN	0.0808 (0.0001)	0.0459 (<0.0001)	0.1088 (<0.0001)	0.0789 (0.0001)	0.0448 (<0.0001)	0.0481 (0.0001)
	NP	0.0781 (<0.0001)	0.0463 (<0.0001)	0.0942 (<0.0001)	0.0759 (<0.0001)	0.0437 (<0.0001)	0.0470 (0.0003)
K-means			0.2819 (0.1254)			0.2248 (0.0886)	

Table 5
NMI and standard deviation (value in ()) under GSBM. Number of Monte Carlo simulations is 100.

τ		0			DKest optimal		
Depth		RTD	MhD	RPD	RTD	MhD	RPD
	GLM	0.5616 (0.0003)	0.6692 (0.0004)	0.4940 (0.0002)	0.6019 (0.0004)	0.7187 (0.0005)	0.6310 (0.0024)
	GAM	0.6055 (0.0011)	0.7442 (0.0001)	0.3550 (0.0001)	0.6111 (0.0007)	0.7212 (0.0006)	0.7308 (0.0926)
DD(*G*)-Classifier	LDA	0.5581 (0.0005)	0.6201 (0.0004)	0.5213 (0.0004)	0.5933 (0.0005)	0.7002 (0.0005)	0.5341 (0.0034)
	QDA	0.4823 (0.0020)	0.5601 (0.0037)	0.3135 (0.0028)	0.5464 (0.0007)	0.4966 (0.0195)	0.5546 (0.0237)
	KNN	0.5932 (0.0010)	0.7300 (0.0002)	0.5049 (0.0003)	0.6026 (0.0007)	0.7353 (0.0006)	0.7202 (0.0007)
	NP	0.6032 (0.0012)	0.7288 (0.0002)	0.5508 (0.0002)	0.6120 (0.0005)	0.7399 (0.0006)	0.7254 (0.0033)
Edgewise-svm				0.0074 (0.0049)			
K-means			0.2690 (0.0847)			0.3623 (0.0206)	

5.1. Political blogs

We start from the benchmark data on interactions between liberal and conservative blogs over a period prior to the 2004 U.S. Presidential Election (Adamic and Glance, 2005). The dataset consists of 1490 vertices with an average degree 22.436, and undirected edges represent hyperlinks between blog sites. Fig. 6 depicts the political blogs network, where red vertices represent conservative blogs and blue vertices correspond to liberal blogs. In contrast to Joseph and Yu (2016), we keep all the isolated vertices (vertices with no connection to others). Note that such isolated vertices may be viewed as potential "outliers" or "anomalies".

We randomly choose a training set of 600 vertices (300 vertices for liberal blogs and conservative blogs, respectively). Using the DKest procedure, we select an optimal regularizer τ of 1.95. Regularization tends to improve clustering performance of the DD(G)-classifier for all considered depth functions. For instance, compared to the unregularized DD-plot based on the Mahalanobis depth (right panel of Fig. 7), points in the corresponding regularized DD-plot (left panel of Fig. 7) stay noticeably closer to horizontal and vertical axes, which makes clustering process more effective. Furthermore, misclassification rate for the DD(G)-classifier with the Mahalanobis depth with regularization is four times lower than its unregularized counterpart, that is, 0.10 vs. 0.44, respectively.

We now consider sensitivity of the DD(G)-classifier in respect to a choice of regularization. For instance, if τ is set up as a network order (i.e., $\tau = 1490$), then misclassification rate of the regularized DD(G) increases up to 3% in respect to optimal regularization; in contrast, misclassification rate of the K-means with non-optimal regularization of $\tau = 1490$ increases up to 15%. Similar conclusions are obtained if we remove isolated vertices.

Hence, not only as expected in view of its supervised settings, clustering performance of the DD(G) classifier is higher, but the DD(G) classifier is more resistant to misspecification of an optimal regularizer.

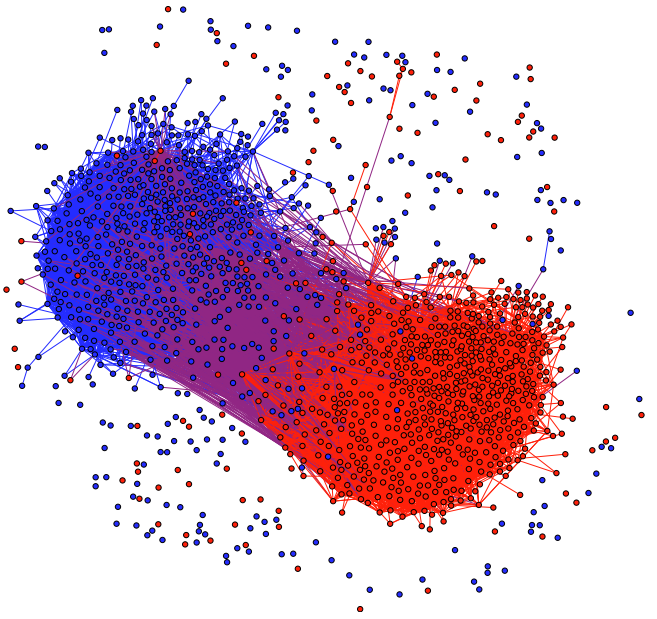

Fig. 6. Political Blogs prior to the 2004 U.S. Presidential Election. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

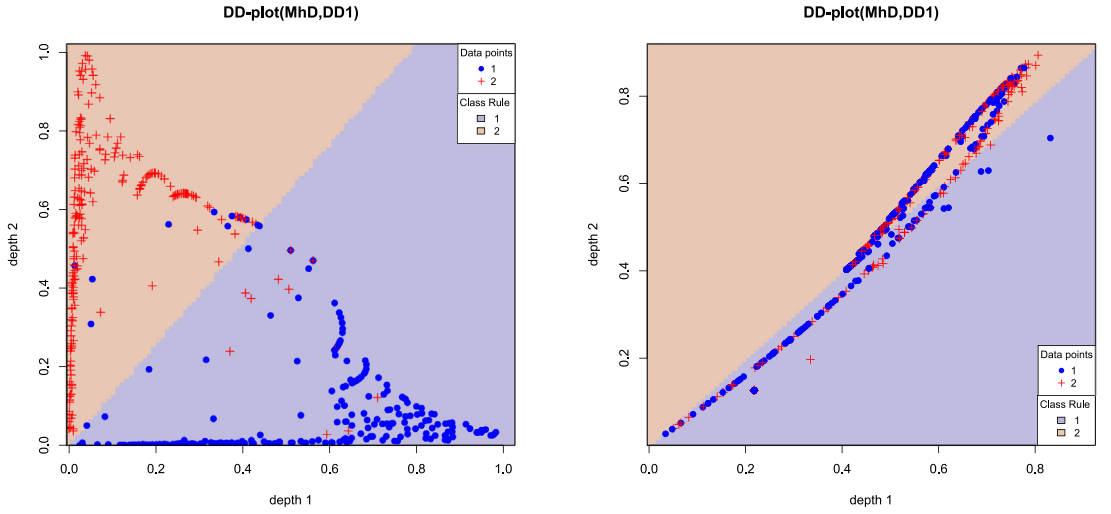

Fig. 7. DD-plot on political blog data based on a Mahalanobis depth. Here '+' and solid 'o' represent blogs from conservative and liberal communities, respectively. Left and right panels correspond to regularized and unregularized Laplacians, i.e. L_τ and L, respectively.

5.2. Discovering communities in terrorist organizations

On the morning of September, 11th, 2001, the Islamic terrorist group al-Qaeda hijacked four passenger airliners and attacked the World Trade Center and the Pentagon, resulting in the loss of approximately 3000 lives. More recently, on the evening of November, 13th, 2015, the Islamic State of Iraq and Syria (ISIS) carried out coordinated terrorist attacks in Paris and its northern suburb, Saint-Denis, which was then followed by a bombing in Brussels. These terrorist attacks presumably belong to the similar dark terrorist network. In order to protect the world from terrorism, it is necessary to strengthen counterterrorism by studying not only successful and failed terrorist attacks (Everton, 2012; Charles and Maras, 2015), but also the dominant ideology and philosophical foundation behind terrorist actions. Moreover, terrorist groups which share the same philosophical foundation tend to affiliate or collaborate with each other, thus naturally, forming a terrorism network. Such networks are unified by a common ideology and/or goal, and allow the participating terrorist groups to share resources and information.

Our dataset includes information of 122 terrorist and extremist organizations and their interactions within the United States between 1970 and 2011. The data are collected by the National Consortium for the Study of Terrorism and Responses to Terrorism (START) (START, 2016). To create the network, we view each terrorist organization as a vertex, and an edge is added if according to START, terrorist organizations either collaborate or are affiliated with each other. Furthermore, membership of each organization is assigned to one of four communities, according to their dominant ideology: Extreme Left Wing, Extreme Right Wing (including all racist ideologies), Ethno-nationalist/Separatist/Religious, and Single Issue groups that primarily rely on various narrowly-defined causes. Hence, the dataset captures a network structure of terrorist groups based on a wide ideological spectrum. The obtained terrorist network indicates a few large subnetworks of ideologically closer groups, and a number of smaller ones comprised of more "fringe" groups. The biggest group in the dataset is formed by various far-right wing organizations, associated with the Ku Klux Klan (KKK). Note that while in 1882 the U.S. Supreme Court declared the Ku Klux Act unconstitutional, the KKK is not officially designated as a terrorist organization in the United States. Nevertheless, in this study we follow classification adopted by START. Prior to September, 11th, 2001, the deadliest terrorist attack on the U.S. soil was the Oklahoma City bombing in 1995, which was organized by two former KKK leaders. Most of these groups were cooperating and were especially active in the earlier decades of the considered time period. Some more modern extreme right groups form smaller networks, likely active in periods different from the bigger network. A number of extreme left groups were cooperating with various "liberation armies", unified by the goal of "liberation" of certain territories, or installing leftist governments. The other smaller networks show linkages between religious groups — Jewish-focused groups, Pakistan–Afghanistan organizations, etc.

We randomly choose a training set of 48 vertices, and select regularization parameter τ of 2.2 according to DKest procedure (Joseph and Yu, 2016). The resulting misclassification rates delivered by the K-means and the DD(G) classifier with the Mahalanobis depth and GAM are 0.38 and 0.25, respectively. The resulting NMI for the K-means, edgewise-svm, and DD(G) classifiers are 0.05, 0.20, and 0.30, respectively. Hence, the new DD(G) method provides the most accurate classification of the dominant ideology and philosophical foundation behind terrorist groups. The DD(G) approach might be then viewed as a preferred classification method for network community extraction in supervised settings, including but not limited to illicit and terrorist graph-structured data, when some a-priori information is available.

5.3. Bill cosponsorship in the Italian parliament

The Italian Parliament is a particularly appropriate venue to test clustering algorithms. Italian politics are notoriously contentious, with frequent government changes and complex coalitions forming each of the recent governments. In the last 70 years, Italy has had 63 governments, which is an unprecedented rate. The latest, 63rd government attempted Constitutional changes affecting the electoral law, but these measures were rejected by the voters, which led to the resignation of Prime Minister Matteo Renzi (Povoledo, 2016). The Italian Parliament is bicameral, with 945 members in total. The electoral law encourages the parties to run in coalitions and to pass an elaborate system of election thresholds with regional idiosyncrasies (Povoledo, 2015). The two chambers of Parliament are governed by separate election laws, with different election thresholds, and in the case of the Chamber of Deputies, a two-round system. The result of these peculiarities is the already mentioned frequent changes of government. Additionally, the system leads to a very high number of parties, around 10 major ones (at least 4% of the vote) and dozens of minor parties with lower support (Ieraci, 2008).

Consequently, the Italian Parliament is an example of a complex political structure that is not easy to analyze without a set of objective statistical tools. The underlying coalitions of parties are not always clear and are not strictly defined. These features make the institution particularly appropriate for cluster analysis. Some of the behaviors that can be analyzed include voting, rhetoric, and bill cosponsorship. Bill cosponsorship is especially interesting because sponsorship can cross party lines and identify networks of behavior.

In the Italian Parliament, a bill can be proposed by a single deputy, or cosponsored by a group of deputies. Bill cosponsorship can be represented by an undirected and loopless network, with vertices representing parliamentarians, and an edge between parliamentarians indicating at least one legislation cosponsored by them (Signorelli and Wit, 2016). Each parliamentarian is a member of one group, typically a political party or a coalition of parties. As a result, the classification/clustering of the network partitions the deputies into political groups.

We consider the bill cosponsorship network for the Italian parliament (2008–2013) collected by Briatte (2016). The dataset consists of 339 vertices and 10151 unweighted edges. Moreover, membership of each parliamentarian is assigned to one of seven political groups: PD (Partito Democratico), FI-PDL (Forza Italia Il Popolo della Libertá), MPA (Movimento per le Autonomie), LN (Lega Nord), IND (mixed or minor group), IDV (Italia dei Valori), FLI-TP (Futuro e Libertá per l'Italia, Terzo Polo). Fig. 8 depicts the cosponsorship network, colors denote parliament groups.

We randomly choose a training set of 135 vertices, the resulting misclassification rate delivered by the K-means and the DD(G) classifier with random projection depth and NP are 0.15 and 0.08, respectively. The resulting NMI for the K-means, edgewise and DD(G) classifier are 0.70, 0.33 and 0.77, respectively. The DD(G) approach provides the most accurate classification of parliament groups. Since this network is relatively dense, regularization does not help improve the classification and is omitted.

The results depict a dense network, which denotes cooperation between the members of parliament. Most of the interactions are within party, but there are numerous connections across parties. Legislative institutions require a

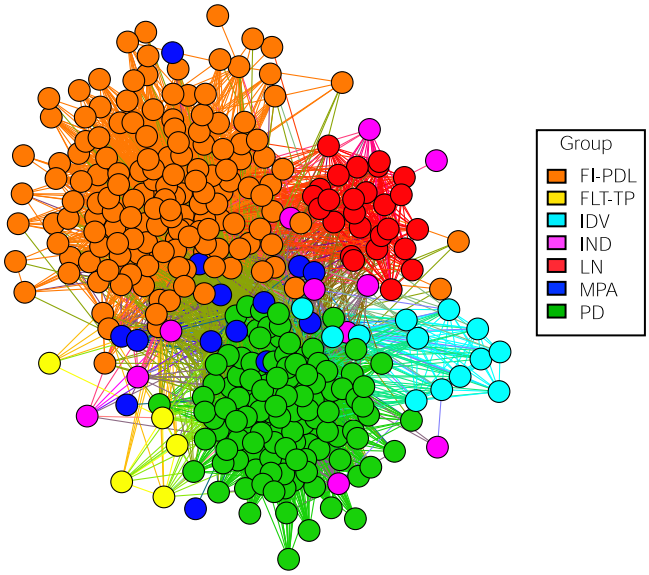

Fig. 8. Bill Cosponsorship in the Italian Parliament.

minimum level of cooperation to provide their functions so these complex networks are expected to occur even in a highly divided parliament. The classifier identifies cosponsorship of members of smaller groups with those in larger groups, which is also expected — the smaller groups do not have enough power and need to work with others to remain relevant. Our classifier performs well and identifies political networks that can be further analyzed by researchers in the political and social sciences.

Furthermore, the depth-based approach allows us to shed light on underlying probabilistic geometry of the data and its role behind hidden mechanisms of network community formation and structure. Let us illustrate this idea in application to two Italian politicians: Francesco Rutelli and Claudio Gustavino. Francesco Rutelli is one of the key Italian politicians who is labeled in the current dataset to belong to MPA (Movimento per le Autonomie). Rutelli's random projection depth value in respect to the MPA membership is only 0.10, and his depth value in respect to FLI-TP (Futuro e Libertá per l'Italia, Terzo Polo) is 0.08. That is, Ruttelli appears to be on a borderline. A deeper analysis of his career reveals that while Rutelli is indeed an active politician, his focus is largely on the European Union (e.g., he is currently President of European Democratic Party), and moreover, he has been a member of 7(!) different parties in the last 30–40 years. In 2009, Francesco Rutelli founded Alleanza per l'Italia (API), or Alliance for Italy. The movement was successful in various regional elections, and later API merged with Futuro e Libertá per l'Italia (FLI) to form Nuovo Polo per l'Italia (NPI), sometimes referred to as Terzo Polo (Third Pole). NPI was a centrist coalition of parties, including Movimento per le Autonomie (MPA) with Raffaele Lombardo as the leader. Both MPA and API are center-right parties with a focus on Christian democratic values. Hence, while being a prominent political leader, Ruttelli does not appear to be rooted in one party but rather to follow some (arguably centrist) trend and switches his political affiliations. The depth-based classifier (DD(G)) that is proposed in this paper captures this dynamics of Rutelli's career and, while Rutelli is (correctly) classified by DD(G) to belong to MPA, his probability to be in this party is only around 10%. In turn, closeness of Rutelli's random projection depth in respect to FLI reflects his leadership of forming the coalition of MPA and FLI. Now, let us consider Claudio Gustavino. Gustavino is a medical doctor who has joined the political arena relatively recently. Gustavino's random projection depth value in respect to FLI is 0.50, and the DD(G) classifier indicates that the probability that he belongs to FLI is high (i.e., 90%). That is, Gustavino appears to be deeply rooted within FLI. Indeed, Claudio Gustavino's affiliation with FLI turns out to be long-standing and his political views matched closely the views of his party. His ideology was stable during his time in office and he did not switch parties. He is strongly classified as a member of FLI, which makes sense given his ideological proximity to the core values of FLI and the stability of his political views during the period he was a member of the party.

These findings suggest that information on data depth and probabilistic geometry of a network can be used not only for classification but also as an additional input to link imputation and link prediction, for example, forming a new coalition or team, which constitutes another interesting future research direction.

6. Conclusions

The current project is primarily motivated by the two overarching questions. First, can we simultaneously combine network community recovery and outlier detection? Second, can we develop a community detection algorithm that is simultaneously more robust to network sparsity and is computationally efficient? Data depth methodology offers an

important arsenal of tools to address these challenges, and our goal is to introduce of the notion of data depth into the analysis of complex networks.

In this paper, we present a novel spectral clustering algorithm for supervised community detection in networks, based on the nonparametric notion of data depth and Data Depth vs. Depth (DD) classifiers. The new DD(G)-community detection algorithm is completely data-driven and requires no prior information on the underlying network degree distributions. Since the DD(G) approach is inherently geometric, we can simultaneously evaluate and, in the case of two clusters, visually assess network communities and outliers. We find that the new community detection DD(G) method is robust to network sparsity as well as outliers in the form of weak and small communities.

While the current DD(G) approach operates within a supervised learning framework, this is just one of the *first* steps toward integration of the powerful robust methodology on data depths into network sciences. Indeed, the notion of data depth can be used in a much broader context in analysis of complex random networks. For instance, below we outline just a number of open questions and future research directions on fusion of data depths and networks. First, we can extend the data depth approach to an unsupervised community detection, that is, instead of the classical loss functions of K-means/medians, we can consider loss functions based on depths. Our preliminary results (Tian and Gel, 2017; Dey et al., 2017) show that this route may be particularly useful for sparser networks, but the choice of appropriate depth and its optimality remains an open problem. For instance, it will be interesting to expand the depth concept to unstructured or heterogeneous communities as well as models where probability of an edge existence depends on various multi-attributes. Second, similarly to the approach of Cuesta-Albertos et al. (2017) for functional data analysis, we can study a combination of depth functions, or even a *meta-depth*. Third, it appears intuitive that the data depth analysis of networks can be useful for initializing cluster centers in conventional K-means. Furthermore, the two-step supervised approach adopted in the current paper can be modified to a semi-supervised framework, for instance, if the dissimilarity measure is selected using the depth function (Zhu et al., 2003; Dhillon et al., 2012). Fourth, data depth can be employed for identifying an optimal number of communities and community validation, particularly in the case of a growing number of classes (Rohe et al., 2011; Choi et al., 2012). Fifth, following (Li and Daniels, 2015; Li et al., 2016), we can extend utility of data-depth methods for better understanding of structural nodes and significance of a detected social community. Sixth, assessment of stability and robustness of the recovered partitions is an important and yet largely under-explored research direction in the analysis of complex networks (Carissimo et al., 2018). We plan to explore stability of the recovered depth-based network partitions and its comparison vs. conventional approaches. Finally, we believe that depth functions developed *directly* for inference of graph-structured data and augmented by the current knowledge on network structural organization such as, for instance, centrality, k-core, and k-shells, will likely become the next "hot" direction at the intersection of network sciences and nonparametric statistics.

While certainly there exist more questions than answers at the current stage, it is certain that the data depth methodology opens new horizons for more robust, data-driven and systematic analysis of complex random networks.

Acknowledgments

The authors thank Robert Serfling, Ricardo Fraiman and Vyacheslav Lyubchich for guidance and highly stimulating and motivating discussions throughout the project. The authors are also grateful to Iliyan Iliev on his help with analysis and interpretation of terrorist networks and cosponsorship in the Italian Parliament. This work was made possible by the facilities of the Shared Hierarchical Academic Research Computing Network of Canada.

References

Abbe, E., 2016. Community detection and the stochastic block model. IEEE Inf. Theory Soc. Newsletter 66 (1), 3–12.

Adamic, L.A., Glance, N., 2005. The political blogosphere and the 2004 U.S. election: Divided they blog. In: Proceedings of the 3rd International Workshop on Link Discovery. pp. 36–43.

Aggarwal, C.C., 2013. Outlier detection in graphs and networks. In: Outlier Analysis. Springer, pp. 343–371.

Amini, A.A., Chen, A., Bickel, P.J., Levina, E., 2013. Pseudo-likelihood methods for community detection in large sparse networks. Ann. Statist. 41, 2097–2122.

Athreya, A., Lyzinski, V., Priebe, C.E., Sussman, D.L., Tang, M., Marchette, D., 2016. A limit theorem for scaled eigenvectors of random dot product graphs. Sankhya A 78 (1), 1–18.

Avrachenkov, K., Cottatellucci, L., Kadavankandy, A., 2015. Spectral properties of random matrices for stochastic block model. In: Proc. of PHYSCOMNET 2015.

Bai, Z., Silverstein, J.W., 2010. Spectral Analysis of Large Dimensional Random Matrices. Springer.

Bande, M.F., de la Fuente, M.O., Galeano, P., Nieto, A., Garcia-Portugues, E., 2016. fda.usc: Functional data analysis and utilities for statistical computing. R package https://cran.r-project.org/package=fda.usc.

Briatte, F., 2016. Network patterns of legislative collaboration in twenty parliaments. Netw. Sci. 4 (2), 266–271.

Cai, T.T., Li, X., 2015. Robust and computationally feasible community detection in the presence of arbitrary outlier nodes. Ann. Statist. 43 (3), 1027–1059.

Campbell, W., Dagli, C., Weinstein, C., 2013. Social network analysis with content and graphs. Linc. Lab. J. 20 (1), 62–81.

Carissimo, A., Cutillo, L., Defeis, I., 2018. Validation of community robustness. Comput. Statist. Data Anal. 120, 1–24.

Charles, C.A., Maras, M.-H., 2015. Strengthening counterterrorism from the information of a successful terrorist attack and failed missions in the United States. J. Appl. Secur. Res. 10 (2), 155–180.

Chaudhuri, K., Chung, F., Tsiatas, A., 2012. Spectral clustering of graphs with general degrees in the extended planted partition model. J. Mach. Learn. Res. 35.1–35.23.

Choi, D., Wolfe, P., Airoldi, E., 2012. Stochastic blockmodels with a growing number of classes. Biometrika 99 (2), 273–284.
Cuesta-Albertos, J.A., Febrero-Bandey, M., de la Fuente, M.O., 2017. The DD^g-classifier in the functional setting. Test 26 (1), 119–142.
Cuesta-Albertos, J.A., Nieto-Reyes, A., 2008. The random Tukey depth. Comput. Statist. Data Anal. 52 (11), 4979–4988.
Cuevas, A., Febrero, M., Fraiman, R., 2007. Robust estimation and classification for functional data via projection-based depth functions. Comput. Statist. 22, 481–496.
Dey, A.K., Gel, Y.R., Poor, H.V., 2017. Intentional islanding of power grids with data depth. In: IEEE Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP2017). pp. 1–5.
Dhillon, P.S., Talukdar, P., Crammer, K., 2012. Metric learning for graph-based domain adaptation. In: Proceedings of the 24th International Conference on Computational Linguistics. COLING'12.
Dyckerhoff, R., Mozharovskyi, P., 2016a. Exact computation of the halfspace depth. Comput. Statist. Data Anal. 98, 19–31.
Dyckerhoff, R., Mozharovskyi, P., 2016b. Exact computation of the halfspace depth. Comput. Statist. Data Anal. 98, 19–30.
Estrada, E., Knight, P.A., 2015. A First Course in Network Theory. Oxford University Press, Oxford.
Everton, S., 2012. Disrupting Dark Networks. Cambridge university press Cambridge.
Fan, J., Wang, W., 2015. Asymptotics of empirical eigen-structure for ultra-high dimensional spiked covariance model. https://arxiv.org/pdf/1502.04733.pdf.
Fortunato, S., 2010. Community detection in graphs. Phys. Rep..
Fortunato, S., Barthelemy, M., 2007. Resolution limit in community detection. Proc. Natl. Acad. Sci. 104 (1), 36–41.
Fraiman, D., Fraiman, F., Fraiman, R., 2015. Statistics of dynamic random networks: A depth function approach. arXiv:1408.3584v3.
Gao, J., Liang, F., Fan, W., Wang, C., Sun, Y., Han, J., 2010. On community outliers and their efficient detection in information networks. In: Proceedings of the 16th ACM SIGKDD. pp. 813–822.
Goldenberg, A., Zheng, A.X., Fienberg, S.E., Airoldi, E.M., 2010. A survey of statistical network models. Found. Trends Mach. Learn. 2 (2), 129–233.
Harenberg, S., Bello, G., Gjeltema, L., Ranshous, S., Harlalka, J., Seay, R., Padmanabhan, K., Samatova, N., 2014. Community detection in large-scale networks: A survey and empirical evaluation. Wiley Interdiscip. Rev. Comput. Stat. 6, 426–439.
Holland, P., Laskey, K.B., Leinhardt, S., 1983. Stochastic blockmodels: First steps. Social Networks 5 (2), 109–137.
Hubert, M., Rousseeuw, P.J., Van Aelst, S., 2008. High-breakdown robust multivariate methods. Statist. Sci. 23 (1), 92–119.
Hyndman, R.J., Shang, H.L., 2010. Rainbow plots, bagplots, and boxplots for functional data. J. Comput. Graph. Statist. 19, 29–45.
Ieraci, G., 2008. Governments and Parties in Italy: Parliamentary Debates, Investiture Votes and Policy Positions (1994–2006). Troubador Publishing Ltd.
Jörnsten, R., 2004. Clustering and classification based on the l_1 data depth. J. Multivariate Anal. 90 (1), 67–89.
Joseph, A., Yu, B., 2016. Impact of regularization on spectral clustering. Ann. Statist..
Kadavankandy, A., Cottatellucci, L., Avrachenkov, K., 2015. Characterization of random matrix eigenvectors for stochastic block model. In: Proc. of the 49th Asilomar Conference on Signals, Systems and Computers. pp. 861–865.
Kumpula, J.M., Saramäki, J., Kaski, K., Kertész, J., 2007. Limited resolution in complex network community detection with Potts model approach. Eur. Phys. J. B 56 (1), 41–45.
Le, C.M., Vershynin, R., 2015. Concentration and regularization of random graphs. arXiv preprint arXiv:1506.00669.
Ledoit, O., Péché, S., 2011. Eigenvectors of some large sample covariance matrix ensembles. Probab. Theory Related Fields 151 (1), 233–264.
Leskovec, J., Lang, K.J., Mahoney, M.W., 2010. Empirical comparison of algorithm for network community detection. In: Proc. of the 19th International Conference on World Wide Web. pp. 631–640.
Li, H.-J., Bu, Z., Li, A., Liu, Z., Shi, Y., 2016. Fast and accurate mining the community structure: Integrating center locating and membership optimization. IEEE Trans. Knowl. Data Eng. 28 (9), 2349–2362.
Li, J., Cuesta-Albertos, J., Liu, R.Y., 2012. DD-classifier: Nonparametric classification procedure based on DD-plot. J. Amer. Statist. Assoc. 107 (498), 737–753.
Li, H.-J., Daniels, J.J., 2015. Social significance of community structure: Statistical view. Phys. Rev. E 91 (1), 012801.
Li, H.-J., Wang, H., Chen, L., 2015. Measuring robustness of community structure in complex networks. Europhys. Lett. 108 (6), 68009.
Li, H.-J., Zhang, X.-S., 2013. Analysis of stability of community structure across multiple hierarchical levels. Europhys. Lett. 103 (5), 58002.
Liu, R.Y., Parelius, J., Singh, K., 1999. Multivariate analysis by data depth: Descriptive statistics, graphics and inference. Ann. Statist. 27 (3), 783–858.
López-Pintado, S., Romo, J., 2009. On the concept of depth for functional data. J. Amer. Statist. Assoc. 104, 718–734.
Manning, C.D., Raghavan, P., Schütze, H., 2008. Introduction to Information Retrieval. Cambridge university press Cambridge.
Mosler, K., Mozharovskyi, P., 2015. Fast DD-classifier of functional data. http://dx.doi.org/10.1007/s00362-015-0738-3.
Newman, M., Clauset, A., 2016. Structure and inference in annotated networks. Nature Commun. 7, 11863.
Nieto-Reyes, A., Battey, H., 2016. A topologically valid definition of depth for functional data. preprint. Statist. Sci. 31 (1), 61–79.
Paindaveine, D., Šiman, M., 2012. Computing multiple-output regression quantile regions. Comput. Statist. Data Anal. 56 (4), 840–853.
Perozzi, B., Akoglu, L., Iglesias Sánchez, P., Müller, E., 2014. Focused clustering and outlier detection in large attributed graphs. In: Proc. of the 20th ACM SIGKDD. ACM, pp. 1346–1355.
Plantié, M., Crampes, M., 2013. Survey on social community detection. In: Ramzan, N., van Zwol, R., Lee, J.-S., Clüver, K., Hua, X.-S. (Eds.), Social Media Retrieval. Springer, London, pp. 65–85.
Povoledo, E., 2015. Italy: Legislative electoral reform (italicum). Global Legal Monitor. Library of Congress 6.
Povoledo, E., 2016. Matteo renzi resigns, ending Italy's 63rd government in 70 years. New York Times 12.
Radcliffe, M., Young, S.J., 2014. The spectra of multiplicative attribute graphs. Linear Algebra Appl. 462, 39–58.
Rohe, K., Chatterjee, S., Yu, B., 2011. Spectral clustering and the high-dimensional stochastic blockmodel. Ann. Statist. 39 (4), 1878–1915.
Rousseeuw, P.J., Ruts, I., 1996. Algorithm AS 307: Bivariate location depth. J. R. Stat. Soc. Ser. C. Appl. Stat. 45 (4), 516–526.
Ruppert, D., 2014. Multivariate Transformations. Wiley StatsRef: Statistics Reference Online.
Scott, J., Carrington, P., 2011. The SAGE Handbook of Social Network Analysis. SAGE.
Signorelli, M., Wit, E.C., 2016. A penalized inference approach to stochastic blockmodelling of community structure in the Italian Parliament. arXiv preprint arXiv:1607.08743.
START, 2016. The National Consortium for the Study of Terrorism and Responses to Terrorism (START). A Department of Homeland Security Center of Excellence led by the University of Maryland, https://www.start.umd.edu.
Subbian, K., Sharma, D., Wen, Z., Srivastava, J., 2014. Finding influencers in networks using social capital. Soc. Netw. Anal. Min. 4 (1), 1–13.
Tang, M., Priebe, C., 2016. Limit theorems for eigenvectors of the normalized Laplacian for random graphs. Ann. Statist. (in press).
Tian, Y., Gel, Y.R., 2017. Fast community detection in complex networks with a K-depths classifier. In: Ahmed, S.E. (Ed.), Big and Complex Data Analysis: Methodologies and Applications. Springer, pp. 139–157.
van Laarhoven, T., Marchiori, E., 2013. Network community detection with edge classifiers trained on LFR graphs, In: Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN).
Wilson, J.D., Wang, S., Mucha, P.J., Bhamidi, S., Nobel, A.B., 2014. A testing based extraction algorithm for identifying significant communities in networks. Ann. Appl. Stat. 8 (3), 1853–1891. http://dx.doi.org/10.1214/14-AOAS760.
Yang, J., Leskovec, J., 2012. Community-affiliation graph model for overlapping network community detection. In: Proceedings of ICDM2012. pp. 1170–1175.
Zhu, X., Ghahramani, Z., Lafferty, J., 2003. Semi-supervised learning using Gaussian fields and harmonic functions. In: Proceedings of ICML 2003. vol. 3. pp. 912–919.
Zuo, Y., Serfling, R., 2000. General notions of statistical depth function. Ann. Statist. 28, 461–482.