Robust Classification of Graph-Based Data

Carlos M. Alaíz [*], Michaël Fanuel [†], and Johan A. K. Suykens [‡]

KU Leuven, ESAT, STADIUS Center. B-3001 Leuven, Belgium.

December 21, 2016

A graph-based classification method is proposed both for semi-supervised learning in the case of Euclidean data and for classification in the case of graph data. Our manifold learning technique is based on a convex optimization problem involving a convex regularization term and a concave loss function with a trade-off parameter carefully chosen so that the objective function remains convex. As shown experimentally, the advantage of considering a concave loss function is that the learning problem becomes more robust in the presence of noisy labels. Furthermore, the loss function considered is then more similar to a classification loss while several other methods treat graph-based classification problems as regression problems.

1 Introduction

Nowadays there is an increasing interest in the study of graph-based data, either because the information is directly available as a network or a graph, or because the data points are assumed to be sampled on a low dimensional manifold whose structure is estimated by constructing a weighted graph with the data points as vertices. Moreover, fitting a function of the nodes of a graph, as a regression or a classification problem, can be a useful tool for example to cluster the nodes using some partial knowledge about the partition and the structure of the graph itself.

In this paper, given some labelled data points and several other unlabelled ones, we consider the problem of predicting the label class of the latter. Following the manifold learning framework, the data are supposed to be positioned on a manifold that is embedded in a high dimensional space, or to constitute a graph by themselves. In the first case, the usual assumption is that the classes are separated by low density regions, whereas in the second case is that the connectivity is weaker between classes than inside each of them [1]. On the other side, the robustness of semi-supervised learning methods and their behaviour in the presence of noise, in this case just wrongly labelled data, has been recently discussed in [2], where a robustification method was introduced.

We propose here a different optimization problem, based on a concave error function, which is specially well-suited when the number of available labels is small and which can deal with flipped labels naturally. The major contributions of our work are:

(i) We propose a manifold learning method phrased as an optimization problem which is robust to label noise. While many other graph-based methods involve a regression-like loss function, our loss function intuitively corresponds to a classification loss akin to the well-known hinge loss used in Support Vector Machines.

(ii) We prove that, although the loss function is concave, the optimization problem remains convex provided that the positive trade-off parameter is smaller than the second least eigenvalue of the normalized combinatorial Laplacian of the graph.

(iii) Computationally, the solution of the classification problem is simply given by solving a linear system.

(iv) We introduce a heuristic method to automatically set the parameter in order to get a parameter-free approach.

Let us also emphasize that the method proposed in this paper can be naturally phrased in the framework of kernel methods, as a function estimation in a Reproducing Kernel Hilbert Space. Indeed, the corresponding kernel is then given by the Moore-Penrose pseudo-inverse of the normalized Laplacian. In this sense, this work can be seen as a continuation of [3].

The paper is structured as follows. Section 2 introduces the context of the classification task and it reviews two state-of-the-art methods for solving it. In Section 3 we introduce our proposed robust approach, which is numerically compared with the others in Section 4. The paper ends with some conclusions in Section 5.

2 Classification of Graph-Based Data

2.1 Preliminaries

The datasets analysed in this paper constitute the nodes $\mathcal{V}$ of a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the undirected edges $\mathcal{E}$ are given as a symmetric weight matrix W with non-negative entries. This graph can be obtained in different settings:

- Given a set of data point $\{x_i\}_{i=1}^N$, with $x_i \in \mathbb{R}^d$ and a positive kernel $k(x, y) \geq 0$, the graph weights can be defined as $w_{ij} = k(x_i, x_j)$.
- Given a set of data point $\{x_i\}_{i=1}^N$, with $x_i \in \mathbb{R}^d$, the weights are constructed as follows: $w_{ij} = 1$ if j is among the k nearest neighbours of i for the ℓ_2-norm, and $w_{ij} = 0$ otherwise. Then, the weight matrix W is symmetrized as $(W + W^\intercal)/2$.
- The dataset is already given as a weighted undirected graph.

Some nodes are labelled by ±1 and we denote by $\mathcal{V}_\mathrm{L} \subset \mathcal{V}$ the set of labelled nodes. For simplicity, we identify $\mathcal{V}_\mathrm{L}$ with $\{1, \dots, s\}$ and $\mathcal{V}$ with $\{1, \dots, N\}$, with $s < N$ the number of available labels. Any labelled node $i \in \mathcal{V}_\mathrm{L}$ has a class label $c_i = \pm 1$. We denote by y the label vector defined as follows

$$y_i = \begin{cases} c_i & \text{if } i \in \mathcal{V}_\mathrm{L}, \\ 0 & \text{if } i \in \mathcal{V} \setminus \mathcal{V}_\mathrm{L}. \end{cases}$$

[*] Email: cmalaiz@esat.kuleuven.be.
[†] Email: michael.fanuel@esat.kuleuven.be.
[‡] Email: johan.suykens@esat.kuleuven.be.

The methods discussed in this paper are formulated in the framework of manifold learning. Indeed, the classification of unlabelled data points relies on the definition of a Laplacian matrix, which can be seen as a discrete Laplace-Beltrami operator on a manifold [4].

Let $L = D - W$ be the matrix of the combinatorial Laplacian, where $D = \operatorname{diag}(d)$ and where the degree vector is $d = W\mathbb{1}$, e.g., $d_i = \sum_{j=1}^{N} w_{ij}$. We will write $i \sim j$ iff $w_{ij} \neq 0$. The normalized Laplacian, defined by $L^{\mathrm{N}} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$, accounts for a non-trivial sampling distribution of the data points on the manifold. The normalized Laplacian has an orthonormal basis of eigenvectors $\{v_\ell\}_{\ell=0}^{N-1}$, with $v_k^\intercal v_\ell = \delta_{k\ell}$, associated to non-negative eigenvalues $0 = \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{N-1} \leq 2$. Noticeably, the zero eigenvector of L^{N} is simply specified by the node degrees, i.e., we have $v_{0,i} \propto \sqrt{d_i}$ for all $i = 1, \ldots, N$. Notice that the Laplacian can be expressed in this basis according to the lemma below.

Lemma 1. *The normalized Laplacian admits the following spectral decomposition, which also gives a resolution of the identity matrix $I \in \mathbb{R}^{N \times N}$:*

$$L^{\mathrm{N}} = \sum_{\ell=1}^{N-1} \lambda_\ell v_\ell v_\ell^\intercal, \quad I = \sum_{\ell=0}^{N-1} v_\ell v_\ell^\intercal.$$

Proof. See [5]. □

For simplicity, we assume here that each eigenvalue is associated to a one-dimensional eigenspace. The general case can be phrased in a straightforward manner.

Following Belkin and Niyogi [6], we introduce the smoothing functional associated to the normalized Laplacian:

$$S_{\mathcal{G}}(f) = \frac{1}{2} f^\intercal L^{\mathrm{N}} f = \frac{1}{2} \sum_{i,j \mid i \sim j} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2, \qquad (1)$$

where f_i denotes the i-th component of f.

Remark 1. The smoothest vector according to the smoothness functional (1) is the eigenvector v_0, which corresponds to a 0 value, $S_{\mathcal{G}}(v_0) = 0$.

2.2 Belkin–Niyogi Approach

In [6], a semi-supervised classification problem is phrased as the estimation of a (discrete) function written as a sum of the first p smoothest functions, that is, the first p eigenvectors of the combinatorial Laplacian. The classification problem is defined by

$$\min_{a \in \mathbb{R}^p} \sum_{i=1}^{s} \left\| c_i - \sum_{\ell=0}^{p-1} a_\ell v_{\ell,i} \right\|_2^2, \qquad (2)$$

where $a_0, \ldots, a_{p-1}$ are real coefficients. The solution of Problem 2, $a^\star$, is obtained by solving a linear system. The predicted vector is then

$$f^\star = \sum_{\ell=1}^{p} a_\ell^\star v_\ell.$$

Finally, the classification of an unlabelled node $i \in \mathcal{V} \setminus \mathcal{V}_{\mathrm{L}}$ is given by $\operatorname{sign}(f_i^\star)$. Indeed, Problem 2 is minimizing a sum of errors of a regression-like problem involving only the labelled data points. The information known about the position of the unlabelled data points is included in the eigenvectors v_ℓ of the Laplacian (Fourier modes), which is the Laplacian of the full graph, including the unlabelled nodes. Only a small number p of eigenvectors is used in order to approximate the label function. This number p is a tuning parameter of the model.

We will denote this model as Belkin–Niyogi Graph Classification (BelkGC).

2.3 Zhou *et al.* Approach

In [7], the following regularized semi-supervised classification problem is proposed:

$$\min_{f \in \mathbb{R}^N} \; \frac{1}{2} f^\intercal L^{\mathrm{N}} f + \frac{\gamma}{2} \|f - y\|_2^2, \qquad (3)$$

where $\gamma > 0$ is a regularization parameter which has to be selected. We notice that the second term in the objective function of Problem 3, involving the ℓ_2-norm of the label vector, can be interpreted as the error term of a least-squares regression problem. Intuitively, Problem 3 will have a solution $f^\star \in \mathbb{R}^N$ such that $f_i^\star \approx 0$ if $i \in \mathcal{V} \setminus \mathcal{V}_{\mathrm{L}}$ (unlabelled nodes), that is, it will try to fit zeroes. Naturally, we will have $f_i^\star \approx c_i$ for all the labelled nodes $i \in \mathcal{V}_{\mathrm{L}}$. Finally, the prediction of the unlabelled node class is given by calculating $\operatorname{sign}(f_i^\star)$ for $i \in \mathcal{V} \setminus \mathcal{V}_{\mathrm{L}}$. The key ingredient is the regularization term which will make the solution smoother by increasing the bias.

Notice that the original algorithm solves Problem 3 once per each class, using as target the indicator vector of the nodes labelled as that class, and then classifying the unlabelled nodes according to the maximum prediction between all the classes. Nevertheless, in this work we consider only binary problems, in which both formulations (using two binary target vectors and predicting with the maximum, or using a single target vector with ±1 and zero values and predicting with the sign) are equivalent. We will denote this model as Zhou *et al.* Graph Classification (ZhouGC).

In the recent work [2], it is emphasized that this method is implicitly robust in the presence of graph noise, since the prediction decays towards zero preventing the errors in far regions of the network from propagating to other areas. Moreover, a modification of this algorithm is proposed to add an additional ℓ_1 penalization, so that the prediction decays faster according to an additional regularization parameter. However, the resultant method is still qualitatively similar to ZhouGC since the loss term is still the one of a regression problem, with the additional disadvantage of having an extra tuning parameter.

2.4 Related Methods

Other semi-supervised learning methods impose the label values as constraints [8, 9]. The main drawback is that, as discussed in [2], the rigid way of including the labelled information makes them more sensible to noise, specially in the case of mislabelled nodes.

On the other side, there are techniques with completely different approaches as Laplacian SVM [10], a manifold learning model for semi-supervised learning based on an ordinary Support Vector Machine (SVM) classifier supplemented with an additional manifold regularization term. This method was originally designed for Euclidian data, hence its scope is different from the previous models. In order to apply this method to graph data, an embedding of the graph has to be performed, what requires the computation of the inverse of a dense Gram matrix entering in the definition of an SVM problem. Hence, the training involves both a matrix inversion of the size of the labelled and unlabelled training data set and a quadratic problem of the same size. In order to reduce the computational cost, a training procedure in the primal was proposed in [11] where the use of a preconditioned conjugate gradient algorithm with an early stopping criterion is suggested. However, these methods still require the choice of two regularization parameters besides the kernel bandwidth. This selection requires a cross-validation procedure which is especially difficult if the number of known labels is small.

3 Robust Method

The two methods presented in Sections 2.2 and 2.3 can be interpreted as regression problems, which intuitively estimate a smooth function $f^\star$ such that its value is approximately the class label, i.e., $f_i^\star \approx c_i$ for all the labelled nodes $i \in \mathcal{V}_{\mathrm{L}}$. We will propose in this section a new method based on a concave loss function and a convex regularization term, which is best suited for classification tasks. Moreover, with the proper constraints, the resulting problem if convex and can be solved using a dual formulation.

We keep as a main ingredient the first term of Problem 3, $\frac{1}{2} f^\intercal L^{\mathrm{N}} f$, which is a well-known regularization term requiring a maximal smoothness of the solution on the (sampled) manifold. However, if the smooth solution is $f^\star$, we emphasize that we have to favour $\operatorname{sign}(f_i^\star) = c_i$ instead of imposing $f_i^\star \approx c_i$ for all $i \in \mathcal{V}_{\mathrm{L}}$. Hence, for $\gamma > 0$, we propose the minimization problem

$$\begin{cases} \min\limits_{f \in \mathbb{R}^N} \ \dfrac{1}{2} f^\intercal L^{\mathrm{N}} f - \dfrac{\gamma}{2} \displaystyle\sum_{i=1}^{N} (y_i + f_i)^2 \\ \text{s.t. } \ f^\intercal v_0 = 0, \end{cases} \tag{4}$$

where γ has to be bounded from above as stated in Theorem 1. The constraint means that we do not want the solution to have a component directed along the vector v_0, since its components all have the same sign (an additional justification is given in Remark 2). We will denote our model as Robust Graph Classification (RobustGC).

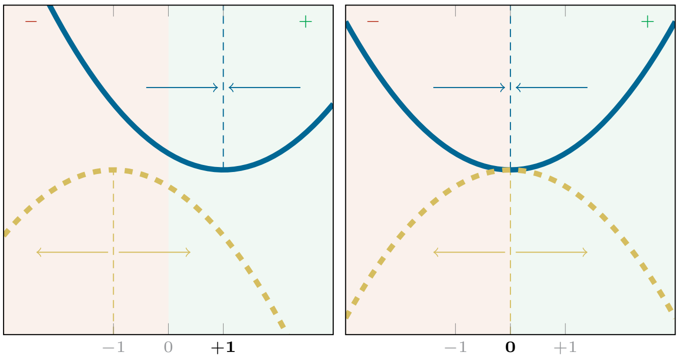
Notice that Problem 4, corresponding to RobustGC, can be written as Problem 3, corresponding to ZhouGC, by doing the following changes: $\gamma \to -\gamma$, $y \to -y$, and by supplementing the problem with the constraint $f^T v_0 = 0$. Both problems can be compared by analysing the error term in both formulations. In ZhouGC this term simply corresponds to the Squared Error (SE), namely $(f_i - y_i)^2$. In RobustGC, a Concave Error (CE) is used instead, $-(f_i + y_i)^2$. As illustrated in Fig. 1, this means that ZhouGC tries to fit the target, both if it is a known label ±1, or if it is zero. On the other side, RobustGC tries to have predictions far from 0, biased towards the direction marked by the label for labelled points. Nevertheless, as shown in Fig. 1a, the model is also able to minimize the CE in the opposite direction to the one indicated by the label, what provides robustness with respect to label noise. Finally, if the label is unknown, the CE only favours large predictions in absolute value. As an additional remark, let us stress that the interplay of the Laplacian-based regularization and the error term, which are both quadratic functions, is yet of fundamental importance. As a matter of fact, in the absence of the regularization term, the minimization of the unbounded error term is meaningless.

RobustGC can be further studied by splitting the error term to get the following equivalent problem:

$$\begin{cases} \min\limits_{f \in \mathbb{R}^N} \ \dfrac{1}{2} f^\intercal L^{\mathrm{N}} f + \gamma \displaystyle\sum_{i=1}^{N} (-y_i f_i) + \gamma \sum_{i=1}^{N} \left(-\dfrac{f_i^2}{2} \right) \\ \text{s.t. } \ f^\intercal v_0 = 0, \end{cases}$$

where the two error terms have the following meaning.

- The first error term is a penalization term involving a sum of loss functions $\mathrm{L}(f_i) = -y_i f_i$. This unbounded loss function term is reminiscent of the hinge loss in Support Vector Machines: $\max(0, 1 - y_i f_i)$. Indeed, for each labelled node $i \in \mathcal{V}_{\mathrm{L}}$, this terms favours values of f_i which have the sign of y_i. However, for each unlabelled node $i \in \mathcal{V} \setminus \mathcal{V}_{\mathrm{L}}$, the corresponding term $\mathrm{L}(f_i) = 0$ vanishes. This motivates the presence of the other error term.



(a) Positive label. **(b)** Unknown label.

Figure 1: Comparison of the Squared Error and the proposed Concave Error, both for a labelled node with $c_i = 1$ (the case $c_i = -1$ is just a reflection of this one) and for an unlabelled point.
Legend: [▬] SE; [▪▪] CE.

- The second error term is a penalization term forcing the value f_i to take a non-zero value in order to minimize $-f_i^2/2$. In particular, if i is unlabelled, this terms favours f_i to take a non-zero value which will be dictated by the neighbours of i in the graph.

The connection between our method and kernel methods based on a function estimation problem in a Reproducing Kernel Hilbert Space (RKHS) is explained in the following remark.

Remark 2. The additional condition $f^\intercal v_0 = 0$ in Problem 4 can also be justified as follows. The Hilbert space $H_K = \{ f \in \mathbb{R}^N \text{ s.t. } f^\intercal v_0 = 0 \}$ is an RKHS endowed with the inner product $\langle f | f' \rangle_K = f^\intercal L^{\mathrm{N}} f'$ and with the reproducing kernel given by the Moore–Penrose pseudo-inverse $K = \left(L^{\mathrm{N}}\right)^\dagger$. More explicitly, we can define $K_i = \left(L^{\mathrm{N}}\right)^\dagger e_i \in \mathbb{R}^N$, where e_i is the canonical basis element given by a vector of zeros with a 1 at the i-th component. Furthermore, the kernel evaluated at any nodes i and j is given by $K(i,j) = e_i^\intercal \left(L^{\mathrm{N}}\right)^\dagger e_j$. As a consequence, the reproducing property is merely [12]

$$\langle K_i | f \rangle_K = \left(\left(L^{\mathrm{N}} \right)^\dagger e_i \right)^\intercal L^{\mathrm{N}} f = f_i,$$

for any $f \in H_K$. As a result, the first term of Problem 4 is equal to $\|f\|_K^2/2$ and the problem becomes a function estimation problem in an RKHS.

Notice that the objective function involves the difference of two convex functions and therefore, it is not always bounded from below. The following theorem states the values of the regularization parameter such that the objective is bounded from below on the feasible set and so that the optimization problem is convex.

Theorem 1. *Let $\gamma > 0$ be a regularization parameter. The optimization problem*

$$\min_{f \in \mathbb{R}^N} \ \frac{1}{2} f^\intercal L^{\mathrm{N}} f - \frac{\gamma}{2} \|f + y\|_2^2 \quad \textit{s.t. } \ f^\intercal v_0 = 0,$$

is strictly convex if and only if $\gamma < \lambda_1$ (the second smallest eigenvalue of L^{N}). In that case, the unique solution is given by the vector:

$$f^\star = \left(\frac{L^{\mathrm{N}}}{\gamma} - I \right)^{-1} \mathrm{p}_0(y),$$

with $\mathrm{p}_0(y) = y - v_0(v_0^\intercal y)$.

Algorithm 1 Algorithm of RobustGC.

```
Input:
    · Graph $\mathcal{G}$ given by the weight matrix $W$ ;
    · Regularization parameter $0 < \eta < 1$ ;
Output:
    · Predicted labels $\hat{y}$ ;
$d_{ii} \leftarrow \sum_j W_{ij}$ ;
$S \leftarrow D^{-1/2} W D^{-1/2}$ ;
$L^{\mathrm{N}} \leftarrow I - S$ ;
$(v_0)_i \leftarrow \sqrt{d_{ii}}$ ;
$v_0 \leftarrow v_0 / \|v_0\|$ ;
Compute $\lambda_1$, second smallest eigenvalue of $L^{\mathrm{N}}$, or, alternatively,
    largest eigenvalue of $S - v_0 v_0^\intercal$ ;
$\gamma \leftarrow \eta \lambda_1$ ;
$f \leftarrow \left(L^{\mathrm{N}}/\gamma - I\right)^{-1} (y - v_0(v_0^\intercal y))$ ;
return $\hat{y} \leftarrow \operatorname{sign}(f)$ ;
```

Proof. Using Lemma 1, any vector satisfying the constraint $f^\intercal v_0 = 0$ can be written as $f = \sum_{\ell=1}^{N-1} \tilde{f}_\ell v_\ell$, where $\tilde{f}_\ell = v_\ell^\intercal f \in \mathbb{R}$ is the projection of f over v_ℓ. Furthermore, we also expand the label vector in the basis of eigenvectors $y = \sum_{\ell=0}^{N-1} \tilde{y}_\ell v_\ell$, with $\tilde{y}_\ell = v_\ell^\intercal y$. Then, the objective function is the finite sum

$$F\left(\tilde{f}_1, \ldots, \tilde{f}_{N-1}\right) = \sum_{\ell=1}^{N-1} \left(\frac{\lambda_\ell - \gamma}{2} \tilde{f}_\ell^2 - \gamma \tilde{y}_\ell \tilde{f}_\ell \right) - \frac{\gamma}{2} \|y\|_2^2,$$

where we emphasize that the term $\ell = 0$ is missing. As a result, the function $F\left(\tilde{f}_1, \ldots, \tilde{f}_{N-1}\right)$ is clearly strictly convex if and only if $\gamma < \lambda_\ell$ for all $\ell = 1, \ldots, N-1$, that is, iff $\gamma < \lambda_1$. Since the objective F is quadratic, its minimum is merely given by $\left(\tilde{f}_1^\star, \ldots, \tilde{f}_{N-1}^\star\right)$, with

$$\tilde{f}_\ell^\star = \frac{\tilde{y}_\ell}{\frac{\lambda_\ell}{\gamma} - 1}, \qquad (5)$$

for $\ell = 1, \ldots, N-1$. Then, the solution of the minimization problem is given by

$$\begin{aligned} f^\star &= \sum_{\ell=1}^{N-1} \tilde{f}_\ell^\star v_\ell = \sum_{\ell=1}^{N-1} \frac{\tilde{y}_\ell}{\frac{\lambda_\ell}{\gamma} - 1} v_\ell \\ &= \left(\frac{L^{\mathrm{N}}}{\gamma} - I \right)^{-1} (y - v_0(v_0^\intercal y)), \end{aligned}$$

which is obtained by using $y - v_0(v_0^\intercal y) = \sum_{\ell=1}^{N-1} \tilde{y}_\ell v_\ell$. This completes the proof. □

By examining the form of the solution of Problem 4 given in (5) as a function of the regularization constant $0 < \gamma < \lambda_1$, we see that taking γ close to the second eigenvalue λ_1 will give more weight to the first eigenvector, while the importance of the next eigenvectors decreases as $1/\lambda_\ell$. Regarding the selection of γ in practice, as shown experimentally just fixing a value of $\gamma = 0.9\lambda_1$ leads to a parameter-free version of RobustGC (denoted PF-RobustGC) that keeps a considerable accuracy.

The complete procedure to apply this robust approach is summarized in Algorithm 1, where γ is set as a percentage η of λ_1 to make it problem independent. Notice that, apart from building the needed matrices and vectors, the algorithm only requires to compute the largest eigenvalue of a matrix and to solve a well-posed linear system.

Illustrative Example

A comparison of ZhouGC, BelkGC and RobustGC is shown in Fig. 2, where the three methods are applied over a very simple graph: a chain with strong links between the first ten nodes, strong links between the last ten nodes, and a weak link connecting the tenth and the eleventh nodes (with a weight ten times smaller). This structure clearly suggests to split the graph in two halves.

In Fig. 2a one node of each cluster receives a label, whereas in Fig. 2b one node of the positive class and four of the negative are labelled, with a flipped label in the negative class. The predicted values of $f^\star$ show that ZhouGC (with $\gamma = 1$) is truly a regression model, fitting the known labels (even the flipped one) and pushing towards zero the unknown ones. BelkGC (with two eigenvectors, $p = 2$) fits much better the unknown labels for nodes far from the labelled ones, although the flipped label push the prediction towards zero in the second example for the negative class. Finally, RobustGC (with $\eta = 0.5$) clearly splits the graph in two for the first example, where the prediction is almost a step function, and it is only slightly affected by the flipped label of the second example. Of course, this experiment is only illustrative, since tuning the parameters of the different models could affect significantly the results.

4 Experiments

In this section we will show empirically how the proposed robust method RobustGC can be successfully applied to the problem of classifying nodes over different graphs, and we will also illustrate the robustness of our method with respect to labelling noise.

The following four models will be compared:

ZhouGC It corresponds to Problem 3, where the parameter γ is selected from a grid of 51 points in logarithmic scale in the interval $\left[10^{-5}, 10^{5}\right]$.

BelkGC It corresponds to Problem 2. The number p of eigenvectors used is chosen between 1 and 51.

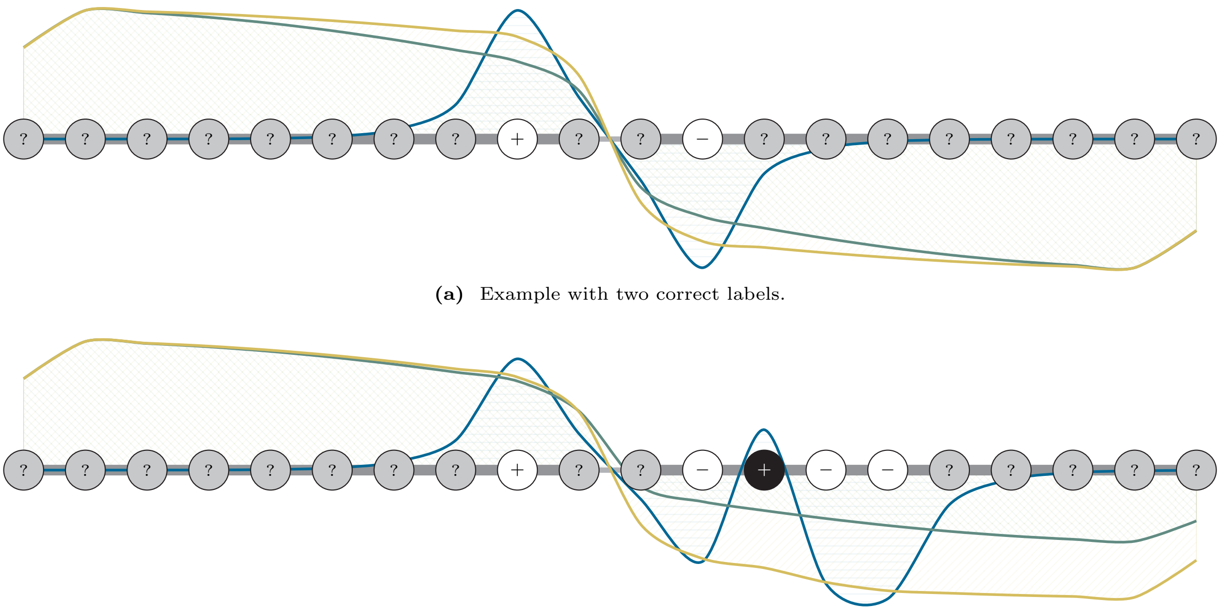
RobustGC It corresponds to Problem 4, where the parameter γ is selected from a grid of 51 points in linear scale between 0 and λ_1.

PF-RobustGC It corresponds to Problem 4, where γ is fixed as $\gamma = 0.9\lambda_1$, so it is a parameter-free method. As shown in Fig. 3, the stability of the prediction with respect to γ suggests to use such a fixed value.

Regarding the selection of the tuning parameters, these models are divided in two groups:

- For ZhouGC, BelkGC and RobustGC, a perfect validation criterion is assumed, so that the best parameter is selected according to the test error. Although this approach prevents from estimating the true generalization error, it is applied to the three models so that the comparison between them should still be fair, and this way we avoid the crucial selection of the parameter, which can be particularly difficult for the small sizes of labelled set considered here. Obviously, any validation procedure will give results at best as good as these ones.

- PF-RobustGC does not require to set any tuning parameter, hence its results are more realistic than those of the previous group, and it is in disadvantage with respect to them. This means that, *if this model outperforms the others in the experiments, it is expected to do it in a real context*, where the parameters of the previous methods have to be set without using test information.



(a) Example with two correct labels.

(b) Example with four correct labels and a flipped one.

Figure 2: Comparison of the different methods over a chain with two clearly separable clusters, where the link between the two middle nodes is ten times smaller than the other links.
Legend: [▭] ZhouGC; [▨] BelkGC; [▨] RobustGC.

4.1 Accuracy of the Classification

The first set of experiments consist in predicting the label of the nodes over the following six supervised datasets:

`digits49-s` and `digits49-w` The task is to distinguish between the handwritten digits 4 and 9 from the USPS dataset [13]; the suffix `-s` denotes that the weight matrix is binary and sparse corresponding to the symmetrized 20-Nearest Neighbours graph, whereas the suffix `-w` corresponds to a non-sparse weight matrix built upon a Gaussian kernel with $\sigma = 1.25$. The total number of nodes is 250 (125 of each class).

`karate` This dataset corresponds to a social network of 34 people of a karate club, with two communities of sizes 18 and 16 [14].

`polblogs` A symmetrized network of hyperlinks between weblogs on US politics from 2005 [15]; there are 1222 nodes, with two clusters of 636 and 586 elements.

`polbooks` A network of books about US politics around 2004 presidential election, with 92 nodes and two classes of 49 and 43 elements.

`synth` This dataset is composed by three clusters of 100 points with a connectivity of 30% inside each cluster and 5% between clusters; the positive class is composed by one cluster and the negative by the other two.

For each dataset, 6 different training sizes (or number of labelled nodes) are considered, corresponding to 1%, 2%, 5%, 10%, 20% and 50% the total number of nodes, provided that this number is larger than two, since at least one sample of each class is randomly selected. Moreover, each experiment is repeated 20 times varying the labelled nodes in order to average the result and check if the differences between them are significant. In order to compare the models we use the accuracy over the non-labelled samples.

The results are included in Table 1, where the significant differences[1] are given by the colours (the darker, the better). We can see that the proposed RobustGC method outperforms both ZhouGC and BelkGC at least for the smallest training sizes, and for all the sizes in the cases of `karate`, `polblogs` (the largest one) and `polbooks`. In the case of `digits49-s` and `digits49-w` RobustGC beats the other methods for the three first sizes, being then beaten by BelkGC in the former and ZhouGC in the latter. Finally, for `synth` the robust RobustGC is the best model for the smallest training size, but it is then outperformed by BelkGC until the largest training size, where both of them solve the problem perfectly. Notice that this dataset is fairly simple, and a spectral clustering approach over the graph (without any labels) could be near a correct partition; BelkGC can benefit for this partition just regressing over the first eigenvectors to get a perfect classification with a very small number of labels. Turning our attention to the parameter-free heuristic approach PF-RobustGC, it is comparable to the approach with perfect parameter selection RobustGC in 3 out of the 6 datasets. In `digits49-s`, `digits49-w` and `synth`, PF-RobustGC is comparable to RobustGC for the experiments with a small number of labels, although it works slightly worse when the number of labels is increased. Nevertheless, the results show that the proposed heuristic performs quite well in practice.

Dependence on the Tuning Parameter

As mentioned before, for the smallest training sets used here, some of them composed by only two labelled nodes, it is impossible to perform a validation procedure. To analyse the dependence of ZhouGC, BelkGC and RobustGC on their tuning parameters, Fig. 3 shows the evolution of the average test accuracy, both for the smallest and largest training sizes. The proposed RobustGC has the most stable behaviour, although as expected it sometimes drops near the critical value $\gamma = \lambda_1$. Nevertheless, this should be the easiest model to tune. ZhouGC shows also a quite smooth dependence, but with a sigmoid shape, where the

[1] Using a Wilcoxon signed rank test for zero median, with a significance level of 5%.

Table 1: Accuracy of the Classification

Data	Labs.	ZhouGC	BelkGC	RobustGC	PF-RobustGC
digits49-s	2	76.6 ± 14.7	74.5 ± 19.6	79.1 ± 16.4	77.4 ± 20.1
	5	80.1 ± 9.4	81.6 ± 11.5	86.9 ± 4.7	85.7 ± 1.9
	12	85.8 ± 4.0	88.2 ± 2.4	88.7 ± 2.7	85.0 ± 1.6
	25	89.3 ± 2.6	91.1 ± 4.5	89.2 ± 2.2	85.0 ± 1.0
	50	92.3 ± 2.4	94.5 ± 2.6	89.7 ± 1.9	84.8 ± 1.4
	125	94.8 ± 2.0	98.1 ± 1.0	90.1 ± 1.9	84.5 ± 2.0
digits49-w	2	70.1 ± 13.4	74.4 ± 9.9	75.5 ± 13.3	75.1 ± 14.0
	5	81.6 ± 9.8	70.6 ± 15.7	82.7 ± 7.4	81.4 ± 8.7
	12	87.9 ± 4.7	85.4 ± 9.4	85.5 ± 5.1	84.4 ± 5.2
	25	93.9 ± 2.1	89.3 ± 5.9	90.1 ± 4.7	89.1 ± 4.0
	50	95.7 ± 1.3	91.9 ± 2.6	92.5 ± 2.8	89.7 ± 3.5
	125	96.9 ± 1.3	95.4 ± 1.7	94.5 ± 2.6	89.6 ± 2.9
karate	—	—	—	—	—
	—	—	—	—	—
	2	90.3 ± 12.2	95.5 ± 7.3	98.9 ± 1.5	98.9 ± 1.5
	3	89.4 ± 8.2	92.7 ± 6.5	98.4 ± 1.7	98.2 ± 1.6
	6	85.5 ± 8.6	96.2 ± 5.2	99.1 ± 1.6	97.9 ± 1.8
	17	96.5 ± 4.8	99.4 ± 1.8	99.4 ± 1.8	98.2 ± 2.8
polblogs	12	92.3 ± 3.1	92.0 ± 4.3	95.6 ± 0.2	95.5 ± 0.2
	24	93.1 ± 1.8	94.1 ± 1.4	95.6 ± 0.2	95.5 ± 0.2
	61	94.5 ± 0.9	94.7 ± 0.6	95.6 ± 0.2	95.5 ± 0.2
	122	94.6 ± 0.7	95.1 ± 0.6	95.6 ± 0.2	95.6 ± 0.2
	244	94.8 ± 0.5	95.2 ± 0.5	95.6 ± 0.3	95.6 ± 0.3
	611	95.3 ± 0.6	95.6 ± 0.7	95.8 ± 0.8	95.7 ± 0.7
polbooks	—	—	—	—	—
	2	97.0 ± 2.0	97.8 ± 0.8	97.8 ± 0.0	97.8 ± 0.0
	4	97.8 ± 1.0	97.4 ± 1.0	97.7 ± 0.0	97.7 ± 0.0
	9	97.5 ± 1.7	97.5 ± 0.7	97.7 ± 0.3	97.7 ± 0.3
	18	97.8 ± 1.3	97.4 ± 0.6	97.5 ± 0.5	97.5 ± 0.5
	46	97.8 ± 1.7	97.4 ± 1.7	97.4 ± 1.7	97.4 ± 1.7
synth	3	79.7 ± 13.5	86.4 ± 11.8	87.0 ± 12.9	85.5 ± 12.6
	6	81.8 ± 9.2	100.0 ± 0.0	91.3 ± 11.3	90.8 ± 11.7
	15	88.2 ± 8.5	100.0 ± 0.0	94.3 ± 8.9	92.1 ± 10.2
	30	93.4 ± 5.3	100.0 ± 0.0	98.0 ± 4.2	96.1 ± 6.8
	60	97.9 ± 1.8	100.0 ± 0.0	99.6 ± 0.6	98.7 ± 2.7
	150	99.6 ± 0.5	100.0 ± 0.0	100.0 ± 0.1	99.5 ± 0.5

maximum tends to be located in a narrow region at the middle. Finally, BelkGC (the model comparable to RobustGC in terms of accuracy) presents the sharpest plot with large changes in the first steps, and hence it is expected to be more difficult to tune.

4.2 Robustness of the Classification with respect to Label Noise

The second set of experiments aims to test the robustness of the classification of the different models with respect to label noise. In particular, a very simple graph of 200 nodes with two clusters is generated with an intra-cluster connectivity of 70%, whereas the connectivity between clusters is either 30% (a well-separated problem) or 50% (a more difficult problem). For each of these two datasets, the performance of the models is compared for different numbers of labels and different levels of noise, which correspond to the percentage of flipped labels. Each configuration is repeated 50 times varying the labelled nodes to average the accuracies.

The results are included in Figs. 4 and 5, where the solid lines represent the average accuracy, and the striped regions the areas between the minimum and maximum accuracies. In the case of the low inter-cluster connectivity dataset of Fig. 4, RobustGC is able to perfectly classify all the points independently of the noise level. Moreover, PF-RobustGC is almost as good as RobustGC, and only slightly worse when the noise is the highest and the number of labels is small. These two models outperform BelkGC, and also ZhouGC, which is clearly the worse of the four approaches. Regarding the high inter-cluster connectivity dataset of Fig. 5, for this more difficult problem RobustGC still gets a perfect classification except when the noise level is very high, where the accuracy drops a little when the number of labels is small. BelkGC is again worse than RobustGC, and the difference is more noticeable when the noise increases. On the other side, the heuristic PF-RobustGC is in this case worse than BelkGC (the selection of γ is clearly not optimal) but it still outperforms ZhouGC.

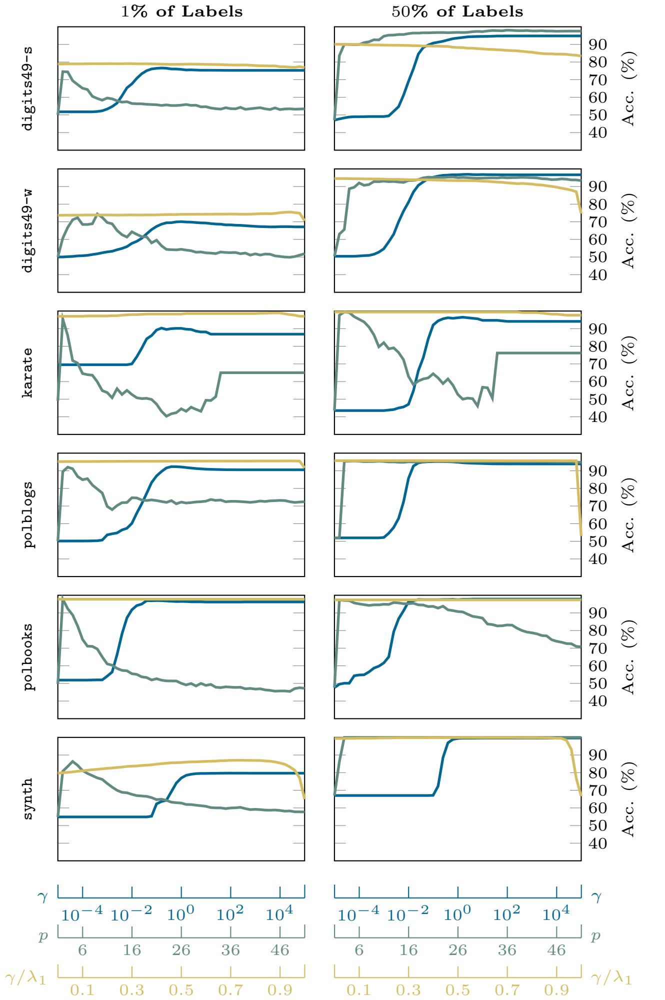


Figure 3: Comparison of the accuracy with respect to the different tuning parameters, for the smallest and largest training sets, and for the six datasets.
Legend: [—] ZhouGC; [—] BelkGC; [—] RobustGC.

5 Conclusions

Starting from basic spectral graph theory, a novel graph-based classification method applicable to semi-supervised classification and graph data classification has been derived in the framework of manifold learning, namely Robust Graph Classification (RobustGC). The method has a clear interpretation in terms of loss functions and regularization. Noticeably, even though the loss function is concave, we have stated the conditions so that the optimization problem is convex. A simple algorithm to solve this problem has been proposed, which only requires to solve a linear system. The results of the method on artificial and real data show that RobustGC is indeed more robust to the presence of wrongly labelled data points, and it is also particularly well-suited when the number of available labels is small.

As further work, we intend to study with more detail the possibilities of the concave loss functions in supervised problems, bounding the solutions using either regularization terms or other alternative mechanisms. Regarding the selection of γ, according to our results the predictions of RobustGC are quite stable with respect to changes in γ in an interval containing the best parameter value. Hence, it seems that a stability criterion could be useful to tune γ.

Acknowledgments

The authors would like to thank the following organizations. • EU: The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC AdG A-DATADRIVE-B (290923). This paper reflects only the authors' views, the Union is not liable for any use that may be made of the contained information. • Research Council KUL: GOA/10/09 MaNet, CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants. • Flemish Government: – FWO: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity); PhD/Postdoc grants. – IWT: SBO POM (100031); PhD/Postdoc grants. • iMinds Medical Information Technologies SBO 2014. • Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012-2017).

References

[1] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-Supervised Learning.* The MIT Press, 1st edition, 2010.

[2] David F. Gleich and Michael W. Mahoney. Using local spectral methods to robustify graph-based learning algorithms. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 359–368, New York, NY, USA, 2015. ACM.

[3] Carlos M Alaíz, Michaël Fanuel, and Johan A K Suykens. Convex formulation for kernel PCA and its use in semi-supervised learning. *arXiv preprint arXiv:1610.06811*, 2016.

[4] Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5 – 30, 2006.

[5] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

[6] Mikhail Belkin and Partha Niyogi. Semi-Supervised Learning on Riemannian Manifolds. *Machine Learning*, 56(1):209–239, 2004.

[7] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pages 321–328. MIT Press, 2004.

[8] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919, 2003.

[9] Thorsten Joachims. Transductive learning via spectral graph partitioning. In *ICML*, volume 3, pages 290–297, 2003.

[10] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov):2399–2434, 2006.

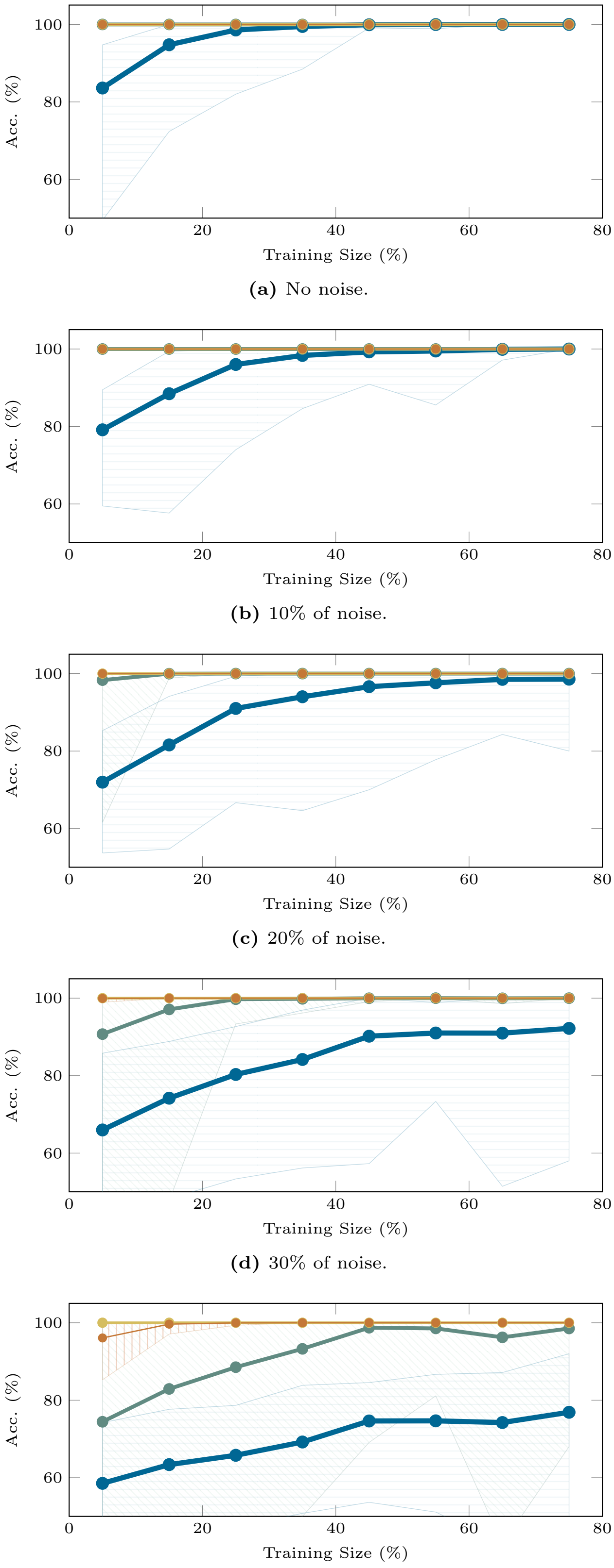
[11] Stefano Melacci and Mikhail Belkin. Laplacian support vector machines trained in the primal. *Journal of Machine Learning Research*, 12(Mar):1149–1184, 2011.

[12] Xueyuan Zhou, Mikhail Belkin, and Nathan Srebro. An iterated graph laplacian approach for ranking on manifolds. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 877–885. ACM, 2011.

[13] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

[14] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, pages 452–473, 1977.

[15] Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 US election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43. ACM, 2005.



(a) No noise.

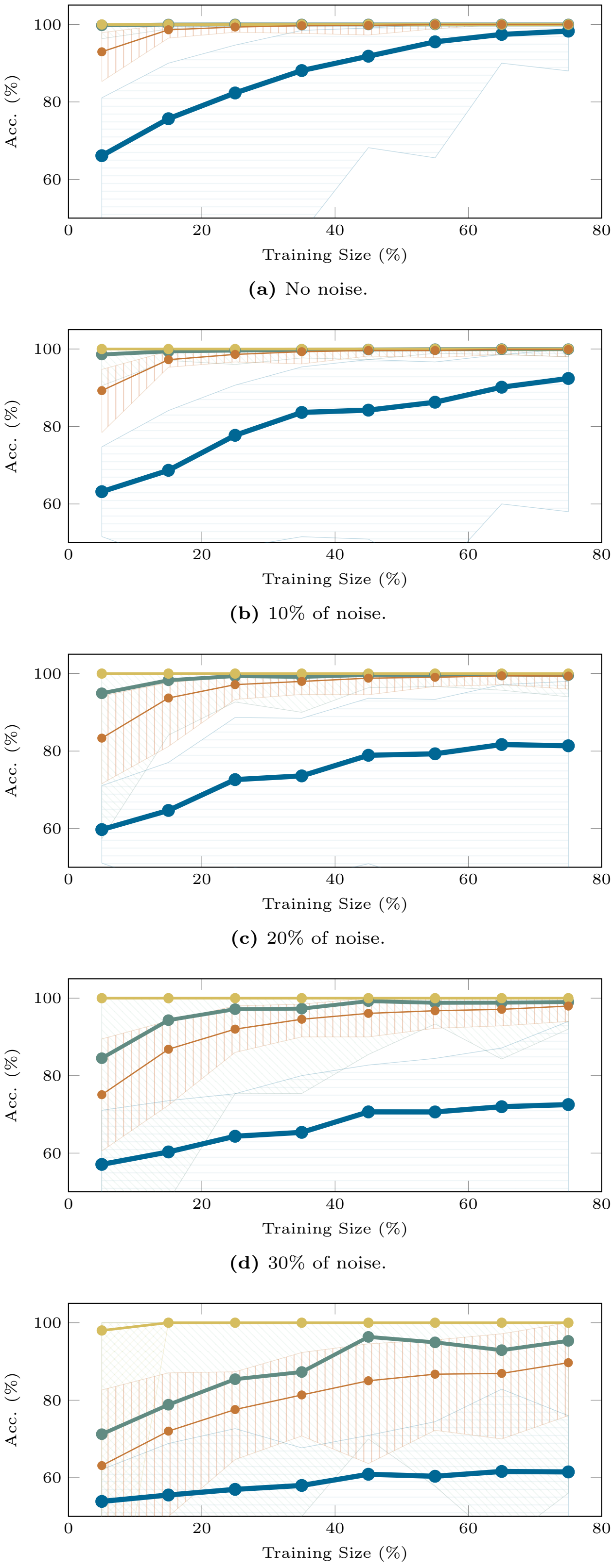
(b) 10% of noise.

(c) 20% of noise.

(d) 30% of noise.

(e) 40% of noise.

Figure 4: Robust comparison for the graph with low inter-cluster connectivity.
Legend:
ZhouGC; BelkGC; RobustGC; PF-RobustGC.



(a) No noise.

(b) 10% of noise.

(c) 20% of noise.

(d) 30% of noise.

(e) 40% of noise.

Figure 5: Robust comparison for the graph with high inter-cluster connectivity.
Legend:
ZhouGC; BelkGC; RobustGC; PF-RobustGC.