Graph Diffusion & PCA Framework for Semi-supervised Learning

Konstantin Avrachenkov, Aurélie Boisbunon, Mikhail Kamalov

Graph diffusion & PCA framework for semi-supervised learning*

Konstantin Avrachenkov[1], Aurélie Boisbunon[2], and Mikhail Kamalov[1**]

[1] INRIA, Sophia Antipolis, France
{konstantin.avratchenkov, mikhail.kamalov}@inria.fr
[2] MyDataModels, Sophia Antipolis, France
abb@mydatamodels.com

Abstract. A novel framework called Graph diffusion & PCA (GDPCA) is proposed in the context of semi-supervised learning on graph structured data. It combines a modified Principal Component Analysis with the classical supervised loss and Laplacian regularization, thus handling the case where the adjacency matrix is *Sparse* and avoiding the *Curse of dimensionality*. Our framework can be applied to non-graph datasets as well, such as images by constructing similarity graph. GDPCA improves node classification by enriching the local graph structure by node covariance. We demonstrate the performance of GDPCA in experiments on citation networks and images, and we show that GDPCA compares favourably with the best state-of-the-art algorithms and has significantly lower computational complexity.

Keywords: Semi-supervised learning, Principal Component Analysis, Citation networks.

1 Introduction

The area of graph-based semi-supervised learning (GB-SSL) focuses on the classification of nodes in a graph where there is an extremely low number of labeled nodes. It is useful in applications such as paper classification to help researchers find articles in a topic, where the data is represented through a citation network, and it is especially beneficial for the classification of medical studies, where collecting labeled nodes is an expensive procedure. In particular, we prepared a real dataset for our experiments which consists of paper abstracts with clinical trials[3] regarding the coronavirus (COVID) topic. Also, GB-SSL is applicable for post labelling in social networks and for detecting protein functions in different biological protein-protein interactions [7].

In GB-SSL, the data consists of the feature matrix $X = [X_i]_{i=1}^n$, where $X_i = (X_{i,j})_{j=1}^d$ lies in a d-dimensional feature space (e.g. from bag-of-words

* Supported by MyDataModels company. This is the author version of the paper accepted at LION 2021, Springer LNCS 12931, pp. 25-39.

** corresponding author

[3] https://clinicaltrials.gov

[15]), and of the label matrix $Y = [Y_{i,j}]_{i,j=1}^{n,k}$ such that $Y_{i,j} = 1$ if $X_i \in \mathcal{C}_j$ and $Y_{i,j} = 0$ otherwise, $\{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$ being a set of k classes. The aim of semi-supervised learning is to estimate Y by a classification result $Z = [Z_{i,j}]_{i,j=1}^{n,k}$ when there is a low number of labels available, while X contains information for both labeled and unlabeled observations. We also assume that the dataset (X, Y) can be represented through the undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $n = |\mathcal{V}|$ the number of nodes with features (e.g. papers) and $e = |\mathcal{E}|$ is the number of edges (e.g. citations). Let $A = [A_{i,j}]_{i,j=1}^{n,n}$ denote the adjacency matrix associated with the $\mathcal{G}$, and $D = \mathrm{diag}(D_{i,i})$ be a diagonal matrix with $D_{i,i} = \sum_{j=1}^{n} A_{i,j}$.

Many works in GB-SSL [1, 31, 32] consider the following minimization problem:

$$\min_{Z \in \mathbb{R}^{n \times k}} \Big\{ \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} ||Z_i - Z_j||_2^2 + \mu \sum_{i=1}^{n} ||Z_i - Y_i||_2^2 \Big\}, \tag{1}$$

where μ is a Lagrangian multiplier, n is the number of nodes, $A = [A_{i,j}]_{i,j=1}^{n,n}$ is an adjacency matrix, $Z = [Z_i]_{i=1}^{n}$ is a classification result and $Y = [Y_i]_{i=1}^{n}$ is a matrix that represents labels. The first part of the objective in (1) is a Laplacian regularization, which penalizes nodes connected from different classes, while the second part is a supervised loss. For the specific problem of paper classification in citation graphs, Problem (1) has the following particular issues:

1. *Sparse A*: the Laplacian regularization cannot estimate classification results Z for graphs with an extremely small number of edges [18] (e.g. citations). Moreover, binary weights ($A_{i,j} = 0$ or $A_{i,j} = 1$) are a poor reflection of node similarity which can lead to a weak estimation of the Laplacian regularization;
2. *Curse of dimensionality*: it arises when A is replaced by a similarity matrix $W = [h(X_i, X_j)]_{i,j=1}^{n} \in \mathbb{R}^{n \times n}$ with a positive definite kernel $h(\cdot)$ and $d \to \infty$ where $X = [X_i]_{i=1}^{n}$ is a matrix of node features and d is the node features space. This replacement is made to avoid the sparsity of A. This issue is especially noticeable in the case of paper classification, for example, based on the Heaps law [2], the d-space of features (bag-of-words[15]) is increasing with respect to the number and length of papers.

The first issue is resolved by Graph Convolution Network (GCN)[19] and Planetoid [32] by prediction of edges, however these solutions are limited in their ability to generalize predicted edge structure. The second issue is treated in GCN as well as in Semi-supervised embedding (SemiEmb) [31], and Least-squares kernel PCA (LS-KPCA) [30], but they require high computation complexity.

In this work, we propose to add a reorganized principal component analysis (PCA) loss to Problem (1) and denote our framework as Graph diffusion & PCA (GDPCA). Not only does it address the aforementioned issues, but we also prove that there exists an explicit solution to the corresponding problem. We apply it to real datasets, and show that GDPCA is the best among GB-SSL state-of-the-art linear algorithms, and that it has comparable performance with GB-SSL neural network algorithms with significantly lower computational complexity.

Finally, we show that GDPCA can also be applied to datasets with no explicit graph structure such as images, and that it outperforms both linear and neural network algorithms for GB-SSL on this type of datasets.

2 Graph-based semi-supervised learning

The recent advances in GB-SSL can be classified into the following rapidly growing directions: **(1) classical linear graph diffusion** algorithms which apply the graph structure for spreading the information of labelled nodes through it, such as Label Propagation (LP) [33], PageRank SSL (PRSSL) [1], or manifold regularization (ManiReg) [4]; and **(2) graph-convolution based neural network** algorithms. The latter category can be further seperated into *(i) nonlinear graph diffusion* algorithms, which apply convolution on the graph's adjacency matrix A with node features, such as Graph Convolution Network (GCN) [19], approximated Personalized graph neural network (APPNP) [20], Planetoid [32], or DeepWalk [23]; and *(ii) graph convolution deep generative* models, focusing on the application of nonlinear graph convolution algorithms with respect to the latent representation of nodes/edges: GenPR [18], Graphite [13].

Linear graph diffusion models are interesting because of their simplicity, but they suffer greatly from the curse of dimensionality. On the other contrary, graph-convolution based neural networks outperform classical linear graph diffusion algorithms and solve the *Curse of dimensionality* issue [19] [32]. However, they are oriented only on computations on small, sparse graphs, leading to the *Sparse A* issue. Furthermore, they do not provide a transparent solution of the classification result Z.

In this work, we present the novel **Graph diffusion** & **PCA** (GDPCA) framework aiming at solving both the *Curse of dimensionality* and *Sparse A* issues while maintaining a low computational complexity. Moreover, our framework provides an explicit solution of the combination of (1) with a reorganized PCA loss. Also we show that GDPCA outperforms the main state-of-the-art GB-SSL classical linear algorithms on various datasets. Our framework also has comparable performance with state-of-the-art GB-SSL neural network algorithms and significantly lower computational complexity.

3 Graph diffusion with reorganized PCA loss

This work is motivated by the idea that principal component analysis (PCA) can solve at least the *Curse of dimensionality* issue. Different works [3, 17, 25, 27, 30] consider a transformation of X by principal components $XU^T = Z$ to the classification results, where $U \in \mathbb{R}^{d\times k}$ is a matrix of principal component vectors from PCA. Instead, we consider principal components which are straightforwardly related to the classification result ($U \in \mathbb{R}^{k\times n}$, $U^T = Z$), as explained in the sequel.

One of the main ideas of this work is that the nodes from different classes have high covariance. This idea lies under the hood of Linear Discriminant Analysis

(LDA) [10] , which was developed for supervised learning. We extend this idea so that it can also be applied in both unsupervised (PCA-BC) and semi-supervised learning (GDPCA).

3.1 PCA for binary clustering (PCA-BC)

In this section, we restrict the setting to the case where no labels are available, and where the nodes come from two clusters. Let us assume that the feature matrix X is sampled from the Gaussian distribution:

$$X_1, \ldots, X_{\frac{n}{2}} \sim \mathcal{N}(\mu_1, C) \text{ and } X_{\frac{n}{2}+1}, \ldots, X_n \sim \mathcal{N}(\mu_2, C), \tag{2}$$

where C is the covariance matrix and μ_1, μ_2 are the expectations of classes $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively. Furthermore, let $||C||_2 = O(1)$, $||\mu_1 - \mu_2||_2 = O(1)$, and the ratio $c_0 = n/d$ be bounded away from zero for large d.

Remark 1. The assumptions $||C||_2 = O(1)$ and $||\mu_1 - \mu_2||_2 = O(1)$ are needed to save the essential variations in d linearly independent directions and define a non-trivial classification case for extremely large d. In particular, this assumption allows us to work with bag-of-words [15] where the d-space is increasing with respect to the number and the length of papers, which leads to the *Curse of dimensionality* issue.

Based on the proof of Theorem 2.2 in [9] and the above restrictions on X, there exists a connection between the binary clustering problem and the PCA maximization objective given by:

$$\max_{U \in \mathbb{R}^{k \times n}} ||\bar{X}U^T||_2^2, \ s.\ t.\ U^T U = 1 \tag{3}$$

where $\bar{X} = [\bar{X}_i^T]_{i=1}^d \in \mathbb{R}^{d \times n}$ with $\bar{X}_i^T = X_i^T - \frac{1}{d}\sum_{j=1}^d X_j^T$; $U = [U_i]_{i=1}^k \in \mathbb{R}^{k \times n}$ is a matrix of principal component vectors. Moreover, $U_{i=1} = U_1 = (U_{1,j})_{j=1}^n$ is the direction of maximum variance, and it can be considered as clustering results in the following way: if $U_{1,j} \geqslant median(U_1)$ then $X_j \in \mathcal{C}_1$ otherwise $X_j \in \mathcal{C}_2$. Figure 1 illustrates the idea that the covariance between nodes from different classes is high. We further demonstrate the applicability of PCA on the binary clustering task with a small numerical experiment. We generated several synthetic datasets (2) with various ratios c_0 and fixed values for expectation ($\mu_1 = (0.5, \ldots, 0)$; $\mu_2 = (0.1, \ldots, 0)$;) and covariance matrix ($C = diag(0.1)$) with $\frac{n}{2}$ the number of nodes in each class: $n = 100$, $d = 1000$, $c_0 = 0.1$; $n = 1000$, $d = 100$, $c_0 = 10$. The code of these experiments is publicly available through a GitHub repository [4]. Figure 2 shows examples of how U_1 discriminates the two classes, even for large d-spaces.

[4] https://github.com/KamalovMikhail/GDPCA

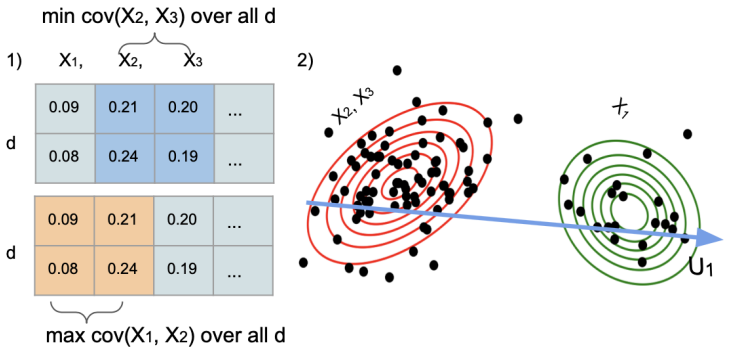

Fig. 1. The intuition behind PCA-BC: 1) Transpose X and visualise the nodes with the maximum and minimum covariance ($cov(\cdot)$) in between; 2) Normalize transposed X and find the direction of maximum covariance by PCA.

3.2 Generalization of PCA-BC for GB-SSL

We propose to modify Problem (1) by adding the reorganized PCA loss (the minus sign being necessary to account for the maximization of the covariance between classes). The optimization problem thus consists in:

$$\min_{Z\in\mathbb{R}^{n\times k}} \left\{ \sum_{i=1}^{n}\sum_{j=1}^{n} A_{i,j}||D_{ii}^{\sigma-1}Z_i - D_{jj}^{\sigma-1}Z_j||_2^2 \right. \\ \left. +\mu\sum_{i=1}^{n} D_{ii}^{2\sigma-1}||Z_i - Y_i||_2^2 - 2\delta||\bar{X}Z||_2^2 \right\} \tag{4}$$

where δ is a penalty multiplier and σ is the parameter controlling the contribution of node degree. We control the contribution of a node degree through the diagonal matrix D to the power in Problem (4) based on the work in [1]. It should be noticed that in Problem (4) we do not require the orthogonality condition $Z^TZ = 1$ as in (3). An interesting feature of Problem (4) is that there exists an explicit solution given by the following proposition.

Proposition 1. *When Problem* (4) *is convex, the explicit solution is given by:*

$$Z = \left(I - \alpha\left(D^{\sigma-1}AD^{-\sigma} + \delta SD^{-2\sigma+1}\right)\right)^{-1}(1-\alpha)Y, \tag{5}$$

where $\alpha = 2/(2+\mu)$, $I \in \mathbb{R}^{n\times n}$ *is the identity matrix and* $S = \frac{\bar{X}^T\bar{X}}{(d-1)} \in \mathbb{R}^{n\times n}$ *is the sample covariance matrix.*

Proof. See Appendix A.

Remark 2. Proposition 1 provides the global minimum of Problem (4) in cases where it is convex, which occurs when the matrix

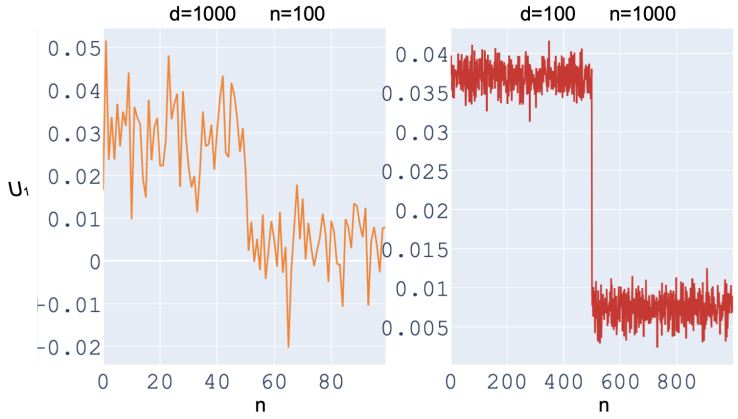

Fig. 2. Mean value of U_1 (the direction of maximum variance in the PCA) on 100 sets of random synthetic data.

$I - \alpha \left(D^{\sigma-1}AD^{-\sigma} + \delta SD^{-2\sigma+1}\right)$ has positive eigenvalues (Theorem 1 in [12]). This condition can be achieved by values of δ such that the sum in brackets will not be upper then 1 and α always less than 1.

Direct matrix inversion in Eq. (5) can be avoided thanks to efficient iterative methods such as the PowerIteration (PI) or the Generalized minimal residual (GMRES) [28] methods. PI consists in iterative matrix multiplications[5] and can be applied when the spectral radius verifies $\rho(\alpha(D^{\sigma-1}AD^{-\sigma} + \delta SD^{-2\sigma+1})) < 1$. GMRES consists in approximating the vectors' solution in Krylov subspace instead of explicit matrix inversion. In practice, PI is more convenient for the computation of Eq. (5) as it converges faster to the best classification accuracy and it can be computed in a distributed regime over nodes [6, p. 135]. The accuracy is computed by comparing maximum values per row between label matrix Y and classification results Z. Furthermore, instead of explicitly computing the spectral radius mentioned above, we can use the following proposition.

Proposition 2. *Suppose that $SD^{-2\sigma+1}$ has only real eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$. Then the inequality $\rho\left(\alpha\left(D^{\sigma-1}AD^{-\sigma} + \delta SD^{-2\sigma+1}\right)\right) < 1$ can be transformed into a simpler one:*

$$1 + \delta\gamma < 1/\alpha \tag{6}$$

where γ is the maximum singular value of $SD^{-2\sigma+1}$ and δ is the penalty multiplier in Eq. (5).

Proof. See Appendix B.

Remark 3. In order to speed up the computation of singular values, we can use the randomized Singular Value Decomposition (SVD) [14]. Inequality (6) can

[5] $Z = \alpha\left(D^{\sigma-1}AD^{-\sigma} + \delta SD^{-2\sigma+1}\right)Z + (1-\alpha)Y$

then be rewritten as $1+\delta(\gamma+\epsilon) < 1/\alpha$, where ϵ is the tolerance of the randomized SVD. The computational complexity of the randomized SVD is $C+O(n)$, where C is the cost of matrix-vector multiplications.

Algorithm 1 gives the outline of our novel Graph diffusion & PCA (GDPCA) framework derived from Propositions 1 and 2. GDPCA uses the following setup: $\mathcal{I}$ is the number of iterations, τ is the tolerance in GMRES, δ is a Lagrangian multiplier, σ is the parameter controlling the contribution of node degree and ϵ is the tolerance in randomized SVD.

Algorithm 1: GDPCA (Graph diffusion & PCA)

INPUT: X, A, Y, σ, α, δ, $\mathcal{I}$, τ, ϵ;
INITIALIZE:
$\bar{X}_i^T = X_i^T - \frac{1}{d}\sum_j^d X_j^T \forall i \in (1,\ldots,n)$; $S = \frac{\bar{X}^T\bar{X}}{d-1}$
$\gamma = randomizedSVD(SD^{-2\sigma+1})$
IF: $1+\delta(\gamma-\epsilon) < 1/\alpha$:
$Z = PI(\alpha(D^{\sigma-1}AD^{-\sigma} + \delta SD^{-2\sigma+1}), (1-\alpha)Y, \mathcal{I})$
ELSE:
$Z = GMRES((I - \alpha(D^{\sigma-1}AD^{-\sigma} + \delta SD^{-2\sigma+1})), (1-\alpha)Y, \tau, \mathcal{I})$

Note also that Proposition 1 simplifies to the known results of PRSSL[1] for the value $\delta = 0$. GDPCA can thus be seen as a generalization of PRSSL enriching the default random walk matrix $D^{\sigma-1}AD^{-\sigma}$ thanks to the sample covariance matrix S. Notice that S is retrieved from PCA loss in Problem (4) (see Appendix A). This enrichment of the binary weights ($A_{i,j} = 0$ or $A_{i,j} = 1$) by node covariance allows bypassing the *Sparse A* issue. Similarly, we assume that our framework solves the *Curse of dimensionality* issue thanks to the use of PCA loss.

4 Experiments

4.1 Datasets description

In the experimental part of this work, we consider two types of datasets: datasets with an underlying graph structure, and datasets that are non-graph based. The latter allow us to test the flexibility of our framework.

Graph-based datasets. We consider the citation networks datasets of Cora, Citeseer, and Pubmed [29]. These datasets have bag-of-words[15] representation for each node (paper) features and a citation network between papers. The citation links are considered as edges in the adjacency matrix A. Each paper has a class label ($X_i \in \mathcal{C}_j$).

Non-graph based datasets. *Images.* We consider the standard MNIST image dataset [21] composed of square 28×28 pixel grayscale images of handwritten digits from 0 to 9. Besides, we flattened square pixels in 784 d-space

features for this dataset. *Text data. Covid clinical trials (CCT) crawled dataset.* We consider a second non-graph based dataset which we prepared and processed from the ClinicalTrials resource[6] from summaries of evidence-based [22] clinical trials on COVID. This dataset is particularly important given the current need from medical experts on this topic. We analyzed 1001 xml files as follows:

1. the feature matrix X was generated from a bag-of-words model based on the descriptive fields "official_title", "brief_summary", "detailed_description", "eligibility";
2. the label matrix Y was generated from the field "masking", which takes values in (*Open, Blind*)[7], as it is one of the essential parameters of evidence-based medicine EBM [8]. The type of masking corresponds to the way of conducting clinical trials: the *Open* way is a less expensive and complicated procedure than the *Blind* one.

Note that the CCT dataset could be useful to other researchers who wish to improve even further the labeling of COVID clinical trials. The registration procedure of clinical trial is useful when authors forget to create masking tag for their work. Particularly after analyzing 1001 xml files, we found that from 3557 clinical trials 1518 of them do not have a masking tag.

As the non-graph based datasets do not have a predefined graph structure, we apply the K-nearest neighbours (KNN)[11] algorithm to generate the adjacency matrix. In Appendix C, we show on validation sets of MNIST and CCT datasets how the choice of distances and number of neighbours for the generation of the adjacency matrix by KNN influence GDPCA. We followed the strategy for train/validation/test splitting as in [32] for Pubmed, Citeseer, Cora and CCT, and as in [24] for MNIST.

The above datasets and code with GDPCA are available through a GitHub repository[8]. Table 1 provides a description of these datasets, where $LR = n_l/n$ is the learning rate with n_l the number of labeled nodes.

Table 1. Dataset statistic.

	Citeseer	Cora	Pubmed	CCT	MNIST
n	3327	2708	19717	2039	50000
e	4732	5492	44338	–	–
k	6	7	3	2	10
d	3703	1433	500	7408	784
LR	0.036	0.052	0.003	0.019	0.002
c_0	0.898	1.889	39.43	0.275	63.77

[6] https://clinicaltrials.gov/ct2/resources/download#DownloadMultipleRecords

[7] In order to simplify the labeling process, we replaced the long description of masking by a shorter version (e.g. Single Blind (Participant, Investigator) by *Blind*).

[8] https://github.com/KamalovMikhail/GDPCA

4.2 State-of-the-art (SOTA) algorithms

As some of the SOTA algorithms cannot be applied to all types of datasets, we consider specific SOTA algorithms depending on the datasets. For the graph-structured Citeseer, Cora and Pubmed datasets, we compare GDPCA to the LP [34] and ManiReg [4] linear graph diffusion algorithms and to the SemiEmb [31], Planetoid [32], GCN [19] and DeepWalk [23] graph convolution-based neural networks. For MNIST, we compared it to the transductive SVM (TSVM) [16] and KNN[11] linear algorithms, and to the GCN neural network. Finally, for CCT, we compared it to the linear LP [34], KNN [11], and PRSSL[1], and to GCN.

Accuracy for non-reproduced benchmarks Since for training and estimation of the GDPCA framework, we use the train/validation/test split strategy for Citeseer, Pubmed, Cora and CCT datasets as in [32] we can use the accuracy of SOTA algorithms from work [32]. In particular, we can take the accuracy of LP[34], ManiReg[4], TSVM[16], SemiEmb[31], Planetoid[32] algorithms from work [32], and the GCN [19], DeepWalk [23] algorithm's accuracy from work[19]. Since for MNIST dataset we use the train/validation/test split strategy as in [24] we can use the value of accuracy of KNN[11] and TSVM[16] algorithms from work [24].

Algorithm parameters for reproduced benchmarks We trained LP, PRSSL, KNN and GCN on CCT and MNIST datasets with the best hyper-parameters defined in the articles describing these algorithms: LP[34] $RBF(\cdot)$ kernel function; GCN [19] 0.5 dropout rate, $5 \cdot 10^{-4}$ L2 regularization, 16 hidden units and 200 epochs; KNN parameters selected by Randomized Search[5] for Cora, Citeseer, Pubmed and CCT datasets.

For a fair model comparison between GDPCA, PRSSL and GCN, we replaced A by $A+I$ as was done in [19, 32]. Also, for GDPCA and PRSSL we fixed $\alpha = 0.9$ and $\sigma = 1$ on all datasets as it was shown in [1] that these parameters provide the best accuracy result for PRSSL. We trained GDPCA on Cora, Citeseer and CCT with $\delta = 1$, $\mathcal{I} = 10$, $\tau = 10^{-3}$, $\epsilon = 10^{-3}$, and the same for MNIST and Pubmed but changing the value of δ to 10^{-3}. We selected these specific $\mathcal{I}$, ϵ, τ parameters by Random Search algorithm [5] as a trade-off between fast computation with GMRES and PowerIteration and accuracy on the validation set. Moreover, for MNIST and CCT we generated a synthetic adjacency matrix A by KNN with respect to the results from Appendix C. In particular, we generated synthetic adjacency matrices based on the following parameters of KNN for datasets: for CCT - *Dice* distance and 7 nearest neighbours; for MNIST - *Cosine* distance and 7 nearest neighbours. We used these synthetic adjacency matrices for the training of GDPCA, PRSSL and GCN algorithms.

4.3 Results

Accuracy results The aforementioned comparisons in terms of accuracy (%) are presented in Table 2 and Table 3. Table 2 shows that GDPCA outperforms

Table 2. Classification accuracy (%) comparison with linear algorithms.

Dataset	Cora	Citeseer	Pubmed	CCT	MNIST
TSVM[16]	57.5	64.0	62.2	–	83.2
KNN[11]	43.9	47.4	63.8	57.1	74.2
LP[34]	68.0	45.3	63.0	53.5	34.2
ManiReg[4]	59.5	60.1	70.7	–	–
PRSSL [1]	69.3	45.9	68.4	55.8	87.2
GDPCA	**77.7**	**73.1**	**76.1**	**61.1**	**88.4**

other SOTA linear algorithms, especially it is significantly better on the Cora, Citeseer and Pubmed, where it outperforms the others by 8%, 9% and 5% respectively. Moreover, Table 3 shows that our linear GDPCA framework provides performance that is close to the best neural network algorithms results. Note that GDPCA has a fixed explicit solution (5) as opposed to the neural network algorithms, which depend on the layer's weights initialization process. Furthermore, Table 2 and Table 3 show that GDPCA has a good performance on standard Cora, Citeseer, Pubmed and MNIST as well as on real dataset CCT.

Table 3. Classification accuracy (%) comparison with neural network algorithms.

Dataset	Cora	Citeseer	Pubmed	CCT	MNIST
SemiEmb[31]	59.0	59.6	71.1	–	–
DeepWalk [23]	67.2	43.2	65.3	–	–
Planetoid[32]	75.7	64.7	77.2	–	–
GCN[19]	**81.5**	70.3	**79.0**	55.2	81.4
GDPCA	77.7	**73.1**	76.1	**61.1**	**88.4**

Computational complexity We finish this experiment section by comparing the computational complexity of GDPCA with the SOTA algorithms that obtained the most similar performance, namely GCN[9] and Planetoid[10]. The algorithmic complexity of GDPCA is $\mathcal{O}(\mathcal{I}nk)$ in the case of PowerIteration, where e' is the number of non-zero elements in matrix $(D^{\sigma-1}AD^{-\sigma} + \delta SD^{-2\sigma+1})$, and $\mathcal{O}(\mathcal{I}nk)$ in the case of GMRES. Note that PowerIteration can be computed in the distributed over node regime [6, p. 135], and GMRES can be distributed over classes. The comparison of GDPCA framework with GCN and Planetoid algorithms in big-$\mathcal{O}$ notation is presented in Table 4. Figure 3 provides the time (in seconds) of 50 completed trainings on CPU(1.4GHz quad-core Intel Core i5)

[9] https://github.com/tkipf/gcn
[10] https://github.com/kimiyoung/planetoid

Table 4. Comparison of computational complexity, where l is the number of layers, n is the number of nodes, d is the number of features, r is the number sampled neighbors per node, k is the batch size; ϕ is the number of random walks; p is the walk length; w is the window size; m is a representation size; k is the number of classes.

ALGORITHM	GCN	GDPCA	PLANETOID
TIME	$O(led + lndm)$	$O(\mathcal{I}nk)$	$O(\phi npw(m + m \log n))$
MEMORY	$O(lnd + ld^2)$	$O(e')$	$O(nld^2)$

for each algorithms. It shows a clear advantage of GDPCA over the GCN and Planetoid especially with GMRES, in terms of computational time.

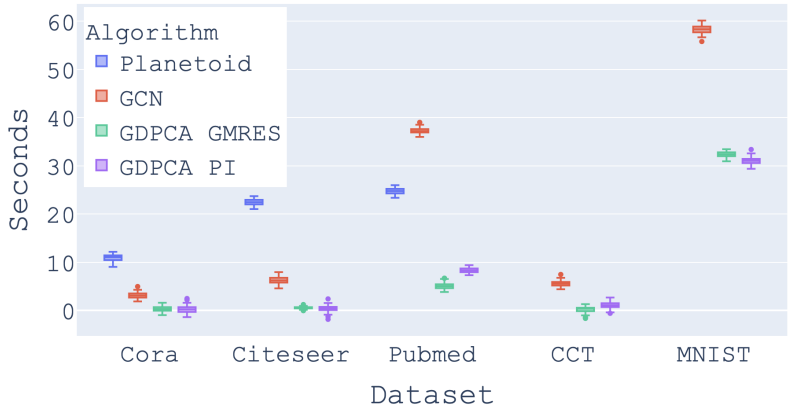

Fig. 3. Computational time of 50 completed trainings on CPU.

Significance of the covariance matrix In this experiment, the aim is to verify that the use of the covariance matrix S actually leads to an improvement. In order to do so, we compare GDPCA with PRSSL ($\delta = 0$) and other values of δ, as well as with variants of GDPCA where S is replaced with the following efficient similarity matrices: $W_{COS} = \frac{[COS(X_i,X_j)]_{i,j=1}^{n,n}}{d-1}$ and $W_{RBF} = \frac{[RBF(X_i,X_j)]_{i,j=1}^{n,n}}{d-1}$. Table 5 displays the average accuracies of each variant along with their statistical significance evaluated with t-tests. It shows that using S in GDPCA is significantly better on the Cora, Citeseer and Pubmed datasets, where it outperforms the others at least by 7%, 8% and 3% respectively. Notice that Table 2 and Table 3 contain accuracy on a test set of fixed dataset splits: as in [32] for Citeseer, Cora, Pubmed and CCT datasets; as in [24] for MNIST dataset, and

Table 5 has accuracy on test sets averaged over 50 random splits. All experiments mentioned above are available through a GitHub repository[11].

Table 5. Average accuracy (%), ▲ denotes the statistical significance for $p < 0.05$.

Dataset	GDPCA $\delta = 1$ (S)	GDPCA $\delta = 10^{-3}$ (S)	PRSSL $\delta = 0$	GDPCA $\delta = 1$ (W_{COS})	GDPCA $\delta = 1$ (W_{RBF})
Cora	77.3 ▲	71.8	69.8	70.1	68.3
Citeseer	73.0 ▲	65.1	44.8	64.8	44.5
Pubmed	68.7	75.8 ▲	67.9	72.6	71.1
CCT	60.4 ▲	54.5	55.6	54.2	56.2
MNIST	62.5	85.3▲	82.6	60.6	59.2

5 Conclusion

In this work, we proposed a novel minimization problem for semi-supervised learning that can be applied to both graph-structured and non-graph based datasets. We provided an explicit solution to the problem, leading to a new linear framework called *Graph diffusion & PCA*. This framework allows to overcome the *Curse of dimensionality*, through the use of reorganized PCA, and the *sparsity of the adjacency matrix*, by considering the covariance matrix, which are both common issues in graph-based semi-supervised learning. We demonstrated the impact of these improvements in experiments on several datasets with and without an underlying graph structure. We also compared it to state-of-the art algorithms and showed that GDPCA clearly outperforms the other linear graph-based diffusion ones. As for the comparison with neural networks, the experiments showed that the performance are similar, while GDPCA has a significantly lower computational time in addition to providing an explicit solution. In future works, we plan to generalize GDPCA to a nonlinear case keeping the low computational complexity and improving classification performance. Also, we want to avoid the bottleneck that arises in the dense covariance matrix S, which can lead to high memory consumptions. Particularly, by distributed PI regimes [6, p. 135] and GMRES, we can directly compute covariance between nodes for a small distributed portion of nodes. This preserves the space consumption as opposed to the precomputed (S).

A Proof of Proposition 1

Proof. This proof uses the same strategy as the proof of Proposition 2 in [1]. Rewriting Problem (4) in matrix form with the standard Laplacian $L = D - A$

[11] https://github.com/KamalovMikhail/GDPCA

and with $Z_{.i}, Y_{.i} \in \mathbb{R}^{n\times 1}$:

$$Q(Z) = 2\sum_{i=1}^{k} Z_{.i}^T D^{\sigma-1} L D^{\sigma-1} Z_{.i} + \mu \sum_{i=1}^{k} (Z_{.i} - Y_{.i})^T D^{2\sigma-1} (Z_{.i} - Y_{.i}) - \delta \sum_{i=1}^{k} Z_{.i} S Z_{.i}^T$$

where $S = \bar{X}^T \bar{X}/(d-1) \in \mathbb{R}^{n\times n}$. Considering $\frac{Q(Z)}{\partial Z} = 0$:

$$2Z^T(D^{\sigma-1} L D^{\sigma-1} + D^{\sigma-1} L^T D^{\sigma-1}) + 2\mu(Z-Y)^T D^{2\sigma-1} - \delta Z^T (S + S^T) = 0$$

Multiplying by $D^{-2\sigma+1}$ and replacing $L = D - A$ results in:

$$Z^T(2I - 2D^{\sigma-1} A D^{-\sigma} + \mu I - 2\delta S D^{-2\sigma+1}) - \mu Y^T = 0$$

Taking out the μ over the parentheses and transposing the equation:

$$Z = \frac{\mu}{(2+\mu)}(I - \frac{2}{(2+\mu)}(D^{\sigma-1} A D^{-\sigma} + \delta S D^{-2\sigma+1}))^{-1} Y$$

Finally, the desired result is obtained with $\alpha = 2/(2+\mu)$. □

B Proof of Proposition 2

Proof. Apply Theorem 1 of sums of spectral radii [35] for the following inequality:

$$\rho(D^{\sigma-1} A D^{-\sigma} + \delta S D^{-2\sigma+1}) \leq \rho\left(D^{\sigma-1} A D^{-\sigma}\right) + \rho(\delta S D^{-2\sigma+1}) < 1/\alpha$$

based on the fact that spectral radius of a matrix similar to the stochastic matrix is equal to 1 (Gershgorin bounds):

$$1 + \delta\rho(S D^{-2\sigma+1}) < 1/\alpha$$

apply the Theorem 7 [26] for replacing $\rho(S D^{-2\sigma+1})$ by the γ maximum singular value of $S D^{-2\sigma+1}$ we obtain the desired result in (6). □

C Generation of synthetic adjacency matrix

For selecting the best synthetic adjacency matrix for GDPCA, we have considered three standard distances, such as *Cosine*, *Minkowski*, *Dice* and the number of neighbours from 1 till 14 for KNN algorithm. The accuracy of GDPCA on above parameters on the validation set for MNIST and CCT datasets are shown in Figure 4. Figure 4 shows that the best GDPCA accuracy on the validation set is obtained with the use of 7 neighbours and *Dice* distance for the CCT dataset is obtained with the use of 7 neighbours and *Cosine* distance for the MNIST dataset.

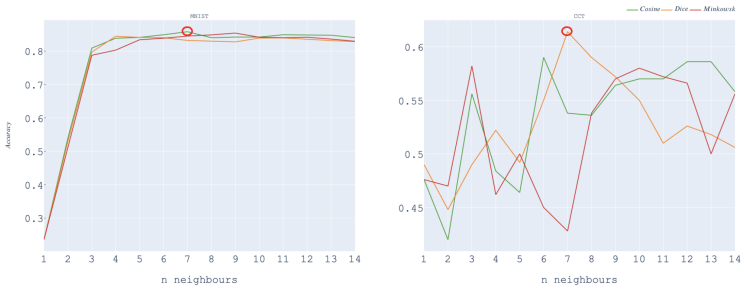

Fig. 4. Estimate different adjacency matrix for GDPCA.

References

1. Avrachenkov, K., Mishenin, A., Gonçalves, P., Sokol, M.: Generalized optimization framework for graph-based semi-supervised learning. In: Proceedings of the 2012 SIAM International Conference on Data Mining. pp. 966–974. SIAM (2012)
2. Baeza-Yates, R., Navarro, G.: Block addressing indices for approximate text retrieval. Journal of the American Society for Information Science **51**(1), 69–82 (2000)
3. Bair, E., Hastie, T., Paul, D., Tibshirani, R.: Prediction by supervised principal components. Journal of the American Statistical Association **101**(473), 119–137 (2006)
4. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of machine learning research **7**(Nov), 2399–2434 (2006)
5. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of machine learning research **13**(2) (2012)
6. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and distributed computation: numerical methods, vol. 23. Prentice hall Englewood Cliffs, NJ (1989)
7. Chapelle, O., Scholkopf, B., Zien, A.: Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. IEEE Transactions on Neural Networks **20**(3), 542–542 (2009)
8. Day, S.J., Altman, D.G.: Blinding in clinical trials and other studies. Bmj **321**(7259), 504 (2000)
9. Ding, C., He, X.: K-means clustering via principal component analysis. In: Proceedings of the twenty-first international conference on Machine learning. p. 29 (2004)
10. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of eugenics **7**(2), 179–188 (1936)
11. Fix, E.: Discriminatory analysis: nonparametric discrimination, consistency properties. USAF school of Aviation Medicine (1951)
12. Freund, R.M.: Quadratic functions, optimization, and quadratic forms (2004)
13. Grover, A., Zweig, A., Ermon, S.: Graphite: Iterative generative modeling of graphs. In: International conference on machine learning. pp. 2434–2444. PMLR (2019)
14. Halko, N., Martinsson, P.G., Tropp, J.A.: Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. SIAM review **53**(2), 217–288 (2011)

15. Harris, Z.S.: Distributional structure. Word **10**(2-3), 146–162 (1954)
16. Joachims, T.: Transductive inference for text classification using support vector machines. In: Icml. vol. 99, pp. 200–209 (1999)
17. Johnson, R., Zhang, T.: Graph-based semi-supervised learning and spectral kernel design. IEEE Transactions on Information Theory **54**(1), 275–288 (2008)
18. Kamalov, M., Avrachenkov, K.: Genpr: Generative pagerank framework for semi-supervised learning on citation graphs. In: Conference on Artificial Intelligence and Natural Language. pp. 158–165. Springer (2020)
19. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations. ICLR (2017)
20. Klicpera, J., Bojchevski, A., Günnemann, S.: Predict then propagate: Graph neural networks meet personalized pagerank. arXiv preprint arXiv:1810.05997 (2018)
21. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
22. Masic, I., Miokovic, M., Muhamedagic, B.: Evidence based medicine–new approaches and challenges. Acta Informatica Medica **16**(4), 219 (2008)
23. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710 (2014)
24. Rifai, S., Dauphin, Y.N., Vincent, P., Bengio, Y., Muller, X.: The manifold tangent classifier. Advances in neural information processing systems **24**, 2294–2302 (2011)
25. Ritchie, A., Scott, C., Balzano, L., Kessler, D., Sripada, C.S.: Supervised principal component analysis via manifold optimization. In: 2019 IEEE Data Science Workshop (DSW). pp. 6–10. IEEE (2019)
26. Rojo, O., Soto, R., Rojo, H.: Bounds for the spectral radius and the largest singular value. Computers & Mathematics with Applications **36**(1), 41–50 (1998)
27. Roli, F., Marcialis, G.L.: Semi-supervised pca-based face recognition using self-training. In: Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR). pp. 560–568. Springer (2006)
28. Saad, Y., Schultz, M.H.: Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM Journal on scientific and statistical computing **7**(3), 856–869 (1986)
29. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. AI magazine **29**(3), 93–93 (2008)
30. Walder, C., Henao, R., Mørup, M., Hansen, L.: Semi-Supervised Kernel PCA. IMM-Technical Report-2010-10, Technical University of Denmark, DTU Informatics, Building 321 (2010)
31. Weston, J., Ratle, F., Mobahi, H., Collobert, R.: Deep learning via semi-supervised embedding. In: Neural networks: Tricks of the trade, pp. 639–655. Springer (2012)
32. Yang, Z., Cohen, W., Salakhudinov, R.: Revisiting semi-supervised learning with graph embeddings. Proceedings of Machine Learning Research, vol. 48, pp. 40–48. PMLR, New York, New York, USA (20–22 Jun 2016)
33. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation (2002)
34. Zhu, X., Ghahramani, Z., Lafferty, J.D.: Semi-supervised learning using gaussian fields and harmonic functions. In: Proceedings of the 20th International conference on Machine learning (ICML-03). pp. 912–919 (2003)
35. Zima, M.: A theorem on the spectral radius of the sum of two operators and its application. Bulletin of the Australian Mathematical Society **48**(3), 427–434 (1993)