Accepted version of paper published in 2018 International Joint Conference on Neural Networks (IJCNN), 2018.
DOI: https://doi.org/10.1109/IJCNN.2018.8489487

Manifold Correlation Graph for Semi-Supervised Learning

Lucas Pascotti Valem, Daniel C. G. Pedronette, Fabricio Breve, Ivan Rizzo Guilherme
Department of Statistics, Applied Mathmatics and Computing (DEMAC),
São Paulo State University (UNESP), Rio Claro, Brazil
lucasvalem@rc.unesp.br, daniel@rc.unesp.br, fabricio@rc.unesp.br, ivan@rc.unesp.br

***Abstract*—Due to the growing availability of unlabeled data and the difficulties in obtaining labeled data, the use of semi-supervised learning approaches becomes even more promising. The capacity of taking into account the dataset structure is of crucial relevance for effectively considering the unlabeled data. In this paper, a novel classifier is proposed through a manifold learning approach. The graph is constructed based on a new hybrid similarity measure which encodes both supervised and unsupervised information. Next, strongly connected components are computed and used to analyze the dataset manifold. The classification is performed through a voting scheme based on primary (labeled) and secondary (unlabeled) voters. An experimental evaluation is conducted, considering various datasets, diverse situations of training/test dataset sizes and comparison with baselines. The proposed method achieved positive results in most of situations.**

I. Introduction

The capacity of automatically inferring a function from labeled data turned up supervised learning methods into indispensable tools present in a wide range of applications. The possibility of discovering patterns encoded in training sets through robust mathematical and computing models supported the development of many methods present in everyday applications. However, despite of the huge success of many supervised learning applications, some limitations still represent relevant challenges. The most important concerns the necessity of sufficient amount of training data for achieving high-accuracy classification rates. The challenge increases as the number of labeled samples used to train the classifier reduces in comparison to the number of unlabeled samples [1].

In this scenario, semi-supervised learning approaches have been established as a promising solution, attracting growing attention from the machine learning research community. Semi-supervised learning (SSL) is halfway between supervised and unsupervised learning, where the main motivation of such approaches consists in exploiting the information encoded also on unlabeled data, in addition to the labeled training data. The capacity of dealing with relatively few labeled training data, but a large number of unlabeled samples makes the semi-supervised learning methods of great importance in many practical applications, specially in the big data era [2].

Many different semi-supervised methods have been proposed based on diverse models developed in branches of machine learning, from kernel methods to Bayesian techniques [2]. Other techniques, designed for exploiting specific characteristics from semi-supervised scenarios, as self-training and co-training have also been developed [3]. Graph-based methods represents another relevant trend in semi-supervised methods, being one of the most active research areas [4], [5], [6], [2]. In general, data samples are represented by the nodes and the edges are defined according to pairwise similarity between samples or k-neighborhood relationships. Despite the common framework provided by the graph modelling, many different approaches have been employed, as graph-cuts [4], random walks [7], label propagation [8] and particles competition and cooperation [6].

Besides semi-supervised learning, other research area from machine learning which have been attracting growing interest is unsupervised learning [9], [10], [11], [12]. In scenarios where the unavailability of training data is extreme, the capacity of understanding and considering the underlying structure of the data is of crucial importance for learning and retrieval tasks. Pioneer unsupervised methods [9] are inspired by semi-supervised approaches, as label propagation [8]. Graph Transduction [9] uses label propagation in retrieval scenarios, with the case of only one known class, which is the class of the query object.

Among various approaches, manifold learning methods [13], [14], [15], [10] have presented remarkable results on both semi-supervised and unsupervised scenarios. In general, one of the basic ideas from manifold learning approaches consists in discovering and understating the underlying dataset manifold, performing similarity measurements capable of considering the intrinsic structures of datasets. As a result, tasks which can be attached to similarity measures, as learning and retrieval, can be performed achieving more effective results.

In this paper, a novel classifier is proposed through a manifold learning approach based on a correlation graph. The method, named as Correlation Graph Classifier (CGC) uses a manifold learning approach for discovering and representing information encoded in the dataset structure. The method is inspired by recent advances on unsupervised manifold learning techniques [10], which uses the correlation graph to model the dataset similarity. However, the proposed classifier present significant novelties in order to represent both labeled and unlabeled data for semi-supervised learning tasks. A new hybrid similarity measure is derived, considering supervised and unsupervised information. Next, strongly connected components are computed and used to analyze the dataset manifold.

The classification is performed through a voting scheme based on primary (labeled) and secondary (unlabeled) voters, which have their weight computed by an effectiveness estimation. The proposed method is experimentally evaluated on various datasets, considering diverse situations of training/test dataset sizes. The method is also evaluated in comparison with baselines, achieving positive results in most of situations.

The remainder of this paper is organized as follows. Section II formally describe the problem addressed. Section III presents the proposed method and Section IV describes the conducted experimental evaluation. Finally, Section V concludes the paper and discusses possible future research directions.

II. Formal Problem Definition

Let $\mathfrak{X} = \{x_1, x_2, \ldots, x_L, x_{L+1}, \ldots, x_N\}$ be a data collection, where each element x_i denotes a data item. The collection $\mathfrak{X}$ can be defined as a partially labeled data set, where $\mathfrak{X}_L = \{x_i\}_{i=1}^{L}$ is the labeled data items subset and $\mathfrak{X}_U = \{x_i\}_{i=L+1}^{N}$ is the unlabeled data items subset.

Let v_{x_i} be a feature vector defined in $\mathbb{R}^n$, which represents the data item x_i. Let d: $\mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ be a distance function (as the Euclidean distance), which computes the distance between two data items according to their corresponding feature vectors. Formally, the distance between two data items x_i and x_j is given by $d(v_{x_i}, v_{x_j})$. For readability purposes, the notation $d(i, j)$ is used along the paper.

Based on the distance function d, a ranked list τ_q can be computed for obtaining the most similar data items to a given data item x_q. The ranked list τ_q=(x_1, x_2, …, x_N) can be formally defined as a permutation of the collection $\mathfrak{X}$. For a permutation τ_q, we interpret $\tau_q(i)$ as the position (or rank) of the data item x_i in the ranked list τ_q. We can say that, if x_i is ranked before x_j in the ranked list of x_q, that is, $\tau_q(i) < \tau_q(j)$, then $d(q, i) \leq d(q, j)$. A ranked list τ_i can be computed for every data item $x_i \in \mathfrak{X}$, in order to obtain a set $\mathcal{T}$ = $\{\tau_1, \tau_2, \ldots, \tau_N\}$ of ranked lists.

Let $\mathfrak{L} = \{1, \ldots, C\}$ be a set which contains the labels of the dataset. Let $y : \mathfrak{X} \rightarrow \mathfrak{L}$ be a function which associates each $x_i \in \mathfrak{X}$ to its label $y(x_i)$ in the final classification results. The semi-supervised learning procedure can be formally defined as the estimation of function $y(x_i)$ for each unlabeled data item $x_i \in \mathfrak{X}_U$.

III. Proposed Method

In this section, we present the classification method mainly based on graphs and correlation measures for semi-supervised learning tasks. The method is deterministic, offers a probability of each element belong to a certain class and can be used in different pattern recognition applications. The main steps of the algorithm are summarized in the Algorithm 1 and detailed discussed in the next subsections.

Algorithm 1 Correlation Graph Classifier

Require: A dataset and set of training data
Ensure: A set of classification arrays
1: Compute the similarity among the top-T elements for each ranked list.
2: Compute the effectiveness estimations for the elements in the training set.
3: **while** $th_c \geq th_{end}$ **do**
4: Built the graph considering edges with weights $\geq th_c$.
5: Find the strongly connected components.
6: Voting step
7: $th_c \leftarrow th_c - th_{step}$
8: **end while**
9: Compute the remaining classification arrays.
10: Normalization and count the votes.

A. Semi-Supervised Similarity Measure

Let x_i and x_j be data samples of a given dataset and let k denotes the neighborhood size. We proposed a semi-supervised similarity measure which is used as basis for the construction of the Correlation Graph. The motivation consists in exploiting both the unlabeled information encoded in the dataset manifold as the labeled available. In this way, the similarity between x_i and x_j can be computed according to the Equation 1:

$$\rho(x_i, x_j) = \frac{\rho_u(x_i, x_j) + \rho_s(x_i, x_j)}{2}, \tag{1}$$

where ρ_u is an unsupervised measure and ρ_s is computed based on labeled data.

The unsupervised measure $\rho_u(x_i, x_j, k)$ can be computed by any correlation measure, considering distances or rank information. The correlation measures are expected to compute values in the interval $[0, 1]$ (higher values, higher correlation). Most of our experimental evaluation considered the Jaccard index as correlation measure. Let $\mathcal{N}_k(i)$ be the set of k-nearest neighbors to the data sample x_i, the measure ρ_u can be defined by Jaccard [1] as follows:

$$\rho_u(x_i, x_j) = \frac{|\mathcal{N}_k(x_i) \cap \mathcal{N}_k(x_j)|}{|\mathcal{N}_k(x_i) \cup \mathcal{N}_k(x_j)|}, \tag{2}$$

While ρ_u defines a contextual similarity measure completely unsupervised, it is possible to estimate the similarity between x_i and x_j based on supervised information if only the class of x_i is known, e.g, if $x_i \in \mathfrak{X}_L$ and $x_j \in \mathfrak{X}_U$. In this way, the objective of supervised measure $\rho_s(x_i, x_j, k)$ is to count the number of elements of the same class of x_i among the neighborhood of x_j. The measure can formally defined according to the Equation 3:

$$\rho_s(x_i, x_j) = \frac{\sum_{l=0}^{k} f_l(x_i, \tau_j(l))}{|\mathcal{N}_k(x_j) \cap \mathfrak{X}_L|} \tag{3}$$

where $f_l(x_i, x_j)$ is a function which verify if two elements have the same label (belong to the same class), as defined by Equation 4:

[1] Besides Jaccard index, the RBO rank correlation measure and the Pearson correlation coefficient were also considered in the experimental evaluation. The Pearson coefficient considered the correlation among distances to k-nearest neighbors.

$$f_l(x_i, x_j) = \begin{cases} 1, & \text{if } x_i, x_j \in \mathfrak{X}_L \wedge l(x_i) = l(x_j); \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

The function $l(x_i)$ denotes the label (or class) of x_i.

B. Effectiveness Estimation

The classification is performed in the proposed method through a voting scheme. In this scenario, we propose a strategy which aims at estimating the authority (or weight) of each voter in the classification process. Our conjecture is that high-accurate voters are positioned in a dense region in the space, with a high number of elements of the same label.

Therefore, it is expected that the similarity to such elements are also high. We propose an effectiveness estimation of a given object x_i, which is computed by the Equation 5:

$$w(x_i) = \frac{\sum_{l=1}^{k} \rho(x_i, \tau_i(l))}{k}. \quad (5)$$

Consequently, an element which presents a straight similarity relationship with its neighbors have an elevated effectiveness estimation and, therefore, an high authority in the voting procedure.

C. Correlation Graph

The Correlation Graph (CG) is the main foundation of the proposed method, in which is based the voting procedure and consequently the classification. The graph is constructed based on different levels of confidence, considering different thresholds of similarity for edges creation. High thresholds produces a sparse graph, but extremely reliable which present a high weight in the voting scheme.

Formally, the graph can be defined as a directed graph $G = (V, E)$, where each note corresponds to an element in the dataset, such that $V = \mathfrak{X}$. The set of edges is defined considering the similarity among the objects in the top-T positions of the ranked lists and the current threshold, such that:

$$E = \{(x_q, x_j) \mid \tau_q(j) \leqslant T \wedge \rho(x_q, x_j) \geqslant t_c\}, \quad (6)$$

where t_c denotes the current threshold at a given moment of execution of the algorithm.

In order to reduce to drawbacks associated to the sparsity of the graph and, at same time, exploiting the dataset manifold, the Strongly Connected Components (SCC) are computed. The strongly connected components of a directed graph are defined by subgraphs that are themselves strongly connected, i.e., where every vertex is reachable from every other vertex. The SCCs expand the similarity neighborhood respecting the geometry of the dataset manifold. The Tarjan [16] algorithm is used to compute the SCCs.

D. Voting Strategy

The Correlation Graph encodes the similarity information obtained from the datatset manifold, considering both unlabeled and labeled data. However, its necessary to condense such information in order to perform classification tasks. A voting strategy is employed aiming at classifying and offer a probability of each element belonging to a certain class.

All the elements, including those from both the labeled and unlabeled sets, are considered voters. However, they are divided into two categories: primary and secondary voters, as discussed in the following.

1) Primary Voters: All the labeled elements, e.g., those which belong to the training set are immediately used as primary voters. A primary voter x_i votes in all the adjacent nodes until a depth $d_p(x_i)$, which is defined as follows.

$$d_p(x_i) = \lceil d_b \times w(x_i) \times t_c \rceil \quad (7)$$

where d_b (depth base) is a parameter. The idea consists and allow deeper votes in the graph to nodes which presents a high effectiveness estimation.

Let V_j be a voting vector of a given data sample x_j. The vote from a primary voter x_i to a data sample x_j in favor of x_i label ($l(x_i)$) is defined according to Equation 8:

$$V_j[l(x_i)] = \frac{w(x_i)}{(1 + d(x_i, x_j))} \times (t_c + 1), \quad (8)$$

where $d(x_i, x_j)$ is the depth, or distance in the graph given by the number of nodes between x_i and x_j.

2) Secondary Voters: In order to maximize the information obtained from the unlabeled data, the elements which has the major probability of belonging to a certain class are allowed to vote in favor of that class. In this sense, we can say that the secondary voters are elected by the primary voters. A node requires to meet two requirements for being elected as a secondary voter: have at least $\alpha\%$ of votes to a certain class with a minimum of β effectiveness estimation.

Since the original class of the secondary voter is unknown, the vote is done according to the class that has received the majority of the votes for that element. A secondary voter votes only in the adjacent vertices from a given depth ($d = 1$). The vote of a secondary voter x_i to a given data sample x_j is defined according to Equation 9:

$$V_j[e_l(x_i)] = w(x_i) \times (t_c + 1), \quad (9)$$

where $e_l(x_i)$ denotes the expected label of x_i, e.g., the class with the highest probability.

E. SCC Votes

A Strongly Connected Component is defined as a set of elements. Each element in the SCC votes to all other elements in the same SCC. Formally, let $\mathcal{S}$ be a strongly connected component, each pair (x_i, x_j) such that $x_i, x_j \in \mathcal{S}$ demand a vote. The elements are distinguished as primary or secondary

voters, depending if they belong to the labeled or unlabeled set. The votes are computed according to Equation 10 to the primary voters:

$$V_j[l(x_i)] = w(x_i) \times (t_c + 1), \tag{10}$$

The votes from secondary voters are computed in the same way as defined by Equation 9. This step is performed only when the current threshold (th_c) is lower than th_{scc}.

F. Final Steps

The iteration is interrupted when all of the nodes have received at least one vote or when the current threshold (th_c) reaches th_{end}. Once the votes depends on the adjacency of the Correlation Graph, it is possible that the algorithm stops iterating before all the elements have received at least one vote. Aiming at solving this problem, a k-NN analysis is executed to compute an classification array for the remaining elements.

G. Classification

When the voting procedure is finished, a voting vector is obtained for all elements from the unlabeled set. All the vectors are normalized, given rise to a probabilistic vector and the class assigned is that with the highest probability. Formally, the CGC classification for a given data sample x_j is computed as:

$$y(x_j) = \arg\max_{l \in \mathfrak{L}} V_j[l], \tag{11}$$

where $y(x_j)$ represents the function which associates the label to a data sample, as defined in Section II.

IV. Experimental Evaluation

This section presents the experimental evaluation conducted in order to assess the effectiveness of the proposed approach. Section IV-A discusses the experimental protocol considered; Section IV-B presents an experimental analysis of the algorithm and Section IV-C presents the obtained results.

A. Experimental Protocol

In order to measure the performance of the proposed method, we have applied it to the well-known Iris dataset [2] and other 5 standard datasets commonly used for evaluation of semi-supervised methods[3]. Some information about these datasets are available at Table I. For detailed description about each of them, one can refer to [2].

For each dataset, there are 10 or 100 labeled data points, and for each case 12 random splits are performed in order to partition the dataset into labeled and unlabeled points. It is ensured that each split contains at least one point from each class. For comparison purpose, we have included the classification results of 13 semi-supervised learning methods presented in [2]. The Nearest Neighbor (1-NN) and Linear SVM (SVM) [17] are used as the base line algorithms. The other 11 algorithms are those that presented the best performance on their respective category and they are presented in the top 11 lines of Table II. The detailed configurations of each method are described in [2]. Besides of these 13 techniques, we have also included other 3 semi-supervised learning graph-based methods in the comparison, which can be considered as the techniques of the same sub-category of the proposed method. These are presented at the last 3 lines of Table II.

[2]Available at https://archive.ics.uci.edu/ml/datasets/iris

[3]Available at http://www.kyb.tuebingen.mpg.de/ssl-book/benchmarks.html

TABLE I: Benchmark datasets used in the experiments.

Dataset	Classes	Dimension	Points	Type
g241c	2	241	1500	artificial
Digit1	2	241	1500	artificial
USPS	2	241	1500	imbalanced
COIL	6	241	1500	
BCI	2	117	400	

TABLE II: Semi-Supervised Learning Methods used for Performance Comparison

Abbreviation	**Method**	**References**
MVU + 1-NN	Maximum Variance Unfolding	[18], [19]
LEM + 1-NN	Laplacian Eigenmaps	[20]
QC + CMN	Quadratic Criterion and Class Mass Regularization	[5], [21]
Discrete Reg.	Discrete Regularization	[22]
TSVM	Transductive Support Vector Machines	[23], [24]
SGT	Spectral Graph Transducer	[24]
Cluster-Kernel	Cluster Kernels	[25]
Data-Dep. Reg.	Data-Dependent Regularization	[26]
LDS	Low-Density Separation	[23]
Laplacian RLS	Laplacian Regularized Least Squares	[27]
CHM (normed)	Conditional Harmonic Mixing	[28]
LGC	Local and Global Consistency	[29]
LP	Label Propagation	[30]
LNP	Linear Neighborhood Propagation	[31]
PCC	Particle Competition and Cooperation	[32], [33]
CUSSL	Comb. Unsup. and Semi-Supervised Learning	[33]

The parameters used in most of the conducted experiments are described in Table III.

TABLE III: Parameters settings.

Parameter	**Default Value**
Vote depth base (d_b)	5
Iteration Start (th_{start})	1
Iteration Step (th_{step})	0.01
Iteration End (th_{end})	0.1
Minimum threshold for SCC increments (th_{scc})	0.8
Min. vote percentage for secondary voters (α)	95%
Min. effect. estimation for secondary voters (β)	0.05
Correlation Measure	Jaccard
Ranked list size (T)	Dataset size (points)

B. Experimental Analysis

This section aims at presenting an experimental analysis of parameters and characteristics of the proposed method. As discussed in the Section III-A, the proposed approach can rely on different correlation measures to compute the classification results. Table IV presents a comparison of results on the Iris dataset for three different correlation measures (RBO, Pearson,

and Jaccard). Different sizes of training sets are considering, ranging from 2% to 10%. In this case, the Jaccard index showed the best results and so we decided to use it as the default correlation measures for the remaining experiments.

TABLE IV: Accuracy for different correlation measures.

	2%	4%	6%	8%	10%
RBO	85.97%	87.39%	90.05%	90.81%	91.86%
Pearson	85.33%	88.70%	91.64%	92.68%	93.17%
Jaccard	**93.28%**	**93.17%**	**93.80%**	**93.57%**	**93.89%**

Other aspect evaluated consists in the impact of the neighborhood size on the accuracy of the method. Figure 1 presents the impact of the neighborhood variation in three different datasets considering 10 labeled data points.

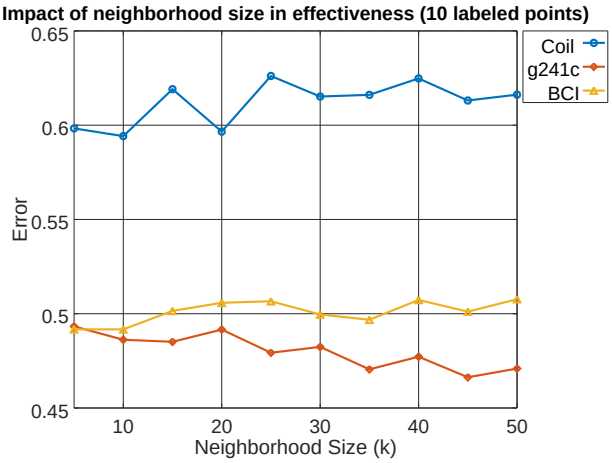

Fig. 1: Impact of variation of k on effectiveness (10 labeled training labels).

The proposed Effectiveness Estimation approach, discussed in Section III-B, is also experimentally analyzed. It is expected that a good effectiveness estimation to provide higher values when the real accuracy is also high. Figure 2 compares the effectiveness estimation with the Average Precision (AP) measure in the Iris dataset. As can be seen, there is a high correlation between the values indicating that the estimation is accurate.

C. Experimental Results

Table V shows the classification accuracy when the proposed method is applied to the Iris dataset. For comparison purposes, PCC and CUSSL are considered as baselines [33]. The best accuracy results for each configuration are highlighted. For each graph configuration, 2% to 10% data items are randomly chosen to compose the labeled subset, which data items are presented to the algorithm with their respective labels. The remaining data items are presented to the algorithm without their labels, so it can classify them. For each graph configuration and labeled subset size, the experiment is repeated 1,000 times with different labeled subsets, so each value in these tables is the average of the 1,000 executions.

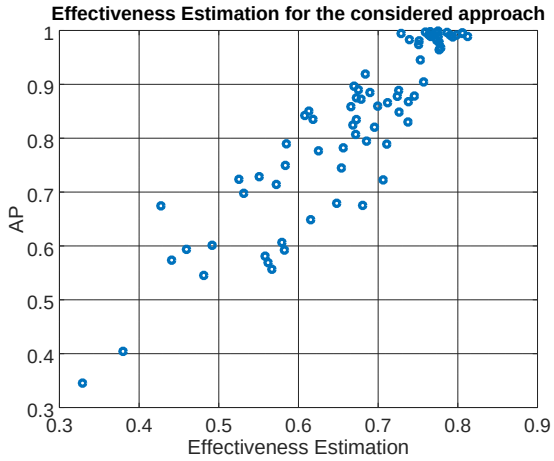

Fig. 2: Evaluation of the proposed Effectiveness Estimation.

These results are plotted on Figure 3. As can be observed, the CGC achieved the best results in all the cases.

TABLE V: Test accuracy results for Iris Dataset.

	2%	4%	6%	8%	10%
PCC	90.44%	89.77%	90.52%	91.23%	91.77%
CUSSL	92.07%	91.42%	91.43%	92.17%	92.89%
CGC	**93.28%**	**93.17%**	**93.80%**	**93.57%**	**93.89%**

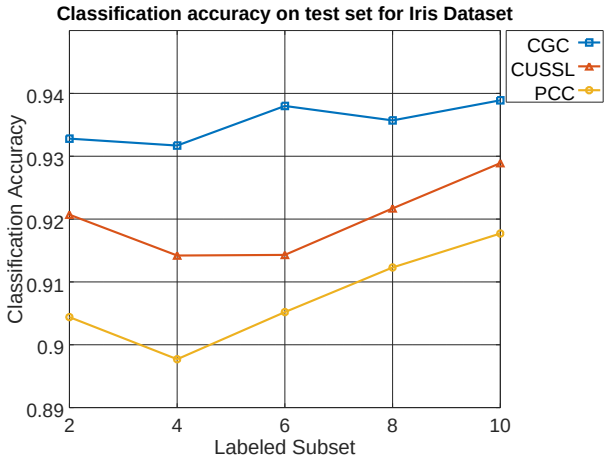

Fig. 3: Accuracy results for the Iris dataset.

Tables VI and VII report the average test errors of the 18 methods and the proposed method applied to the datasets presented in Table I. They show test errors (%) with 10 and 100 labeled training points respectively. The best value of k was used for each dataset, performing a search in the interval [0,50] in steps of 5. For 10 labeled points the values of k considered are 45, 10, 25, 10, 5 for g241c, Digit1, USPS, COIL, and BCI, respectively. For 100 labeled points the values of k considered are 45, 10, 5, 5, 20 for g241c, Digit1, USPS, COIL, and BCI, respectively.

Tables VIII and IX show the rank position obtained by each method on different datasets for 10 and 100 labeled training

TABLE VI: Test Errors (%) with 10 Labeled Training Points

	g241c	Digit1	USPS	COIL	BCI	Mean
1-NN	47.88	13.65	16.66	63.36	49.00	38.11
SVM	47.32	30.60	20.03	68.36	49.85	43.23
MVU + 1-NN	47.15	14.42	23.34	62.62	47.95	39.10
LEM + 1-NN	44.05	23.47	19.82	65.91	48.74	40.40
QC + CMN	39.96	9.80	13.61	59.63	50.36	34.67
Discrete Reg.	49.59	12.64	16.07	63.38	49.51	38.24
TSVM	24.71	17.77	25.20	67.50	49.15	36.87
SGT	22.76	8.92	25.36	–	49.59	26.66
Cluster-Kernel	48.28	18.73	19.41	67.32	48.31	40.41
Data-Dep. Reg.	41.25	12.49	17.96	63.65	50.21	37.11
LDS	28.85	15.63	17.57	61.90	49.27	34.64
Laplacian RLS	43.95	5.44	18.99	54.54	48.97	34.38
CHM (normed)	39.03	14.86	20.53	–	46.90	30.33
LGC	45.82	9.89	9.03	63.45	47.09	35.06
LP	42.61	11.31	14.83	55.82	46.37	34.19
LNP	47.82	8.58	17.87	55.50	47.65	35.48
PCC	37.57	9.94	17.44	58.65	47.66	34.25
CUSSL	40.49	12.30	17.19	59.24	–	32.31
Proposed Method	46.63	12.23	11.99	59.42	49.17	35.89

TABLE VII: Test Errors (%) with 100 Labeled Training Points

	g241c	Digit1	USPS	COIL	BCI	Mean
1-NN	43.93	3.89	5.81	17.35	48.67	23.93
SVM	23.11	5.53	9.75	22.93	34.31	19.13
MVU + 1-NN	43.01	2.83	6.50	28.71	47.89	25.79
LEM + 1-NN	40.28	6.12	7.64	23.27	44.83	24.43
QC + CMN	22.05	3.15	6.36	10.03	46.22	17.56
Discrete Reg.	43.65	2.77	4.68	9.61	47.67	21.68
TSVM	18.46	6.15	9.77	25.80	33.25	18.69
SGT	17.41	2.61	6.80	–	45.03	17.96
Cluster-Kernel	13.49	3.79	9.68	21.99	35.17	16.82
Data-Dep. Reg.	20.31	2.44	5.10	11.46	47.47	17.36
LDS	18.04	3.46	4.96	13.72	43.97	16.83
Laplacian RLS	24.36	2.92	4.68	11.92	31.36	15.05
CHM (normed)	24.82	3.79	7.65	–	36.03	18.07
LGC	41.64	2.72	3.68	45.55	43.50	27.42
LP	30.39	3.05	6.98	11.14	42.69	18.85
LNP	44.13	3.27	17.22	11.01	46.22	24.37
PCC	24.20	2.65	4.65	14.85	44.38	18.15
CUSSL	26.15	2.52	4.81	23.13	–	14.15
Proposed Method	36.16	3.2	4.47	11.41	47.39	20.53

TABLE VIII: Ranking of semi-supervised methods with 10 labeled training points

	g241c	Digit1	USPS	COIL	BCI	Mean
1-NN	17	12	6	10	10	**11**
SVM	15	19	15	17	16	**16.4**
MVU + 1-NN	14	13	17	9	6	**11.8**
LEM + 1-NN	11	18	14	14	8	**13**
QC + CMN	6	4	3	7	18	**7.6**
Discrete Reg.	19	11	5	11	14	**12**
TSVM	2	16	18	16	11	**12.6**
SGT	1	3	19	–	15	**9.5**
Cluster-Kernel	18	17	13	15	7	**14**
Data-Dep. Reg.	8	10	11	13	17	**11.8**
LDS	3	15	9	8	13	**9.6**
Laplacian RLS	10	1	12	1	9	**6.6**
CHM (normed)	5	14	16	–	2	**9.25**
LGC	12	5	1	12	3	**6.6**
LP	9	7	4	3	1	**4.8**
LNP	16	2	10	2	4	**6.8**
PCC	4	6	8	4	5	**5.4**
CUSSL	7	9	7	5	–	**7**
Proposed Method	13	8	2	6	12	**8.2**

TABLE IX: Ranking of semi-supervised methods with 100 labeled training points

	g241c	Digit1	USPS	COIL	BCI	Mean
1-NN	18	16	9	10	18	**14.2**
SVM	7	17	17	12	3	**11.2**
MVU + 1-NN	16	7	11	16	17	**13.4**
LEM + 1-NN	14	18	14	14	10	**14**
QC + CMN	6	10	10	2	12	**8**
Discrete Reg.	17	6	4	1	16	**8.8**
TSVM	4	19	18	15	2	**11.6**
SGT	2	3	12	–	11	**7**
Cluster-Kernel	1	14	16	11	4	**9.2**
Data-Dep. Reg.	5	1	8	6	15	**7**
LDS	3	13	7	8	8	**7.8**
Laplacian RLS	9	8	5	7	1	**6**
CHM (normed)	10	15	15	–	5	**11.25**
LGC	15	5	1	17	7	**9**
LP	12	9	13	4	6	**8.8**
LNP	19	12	19	3	13	**13.2**
PCC	8	4	3	9	9	**6.6**
CUSSL	11	2	6	13	–	**8**
Proposed Method	13	11	2	5	14	**9**

points, respectively. As we can observe, the CGC achieved an average result of 8.2 and 9 among 19 methods.

V. Conclusions

The huge growth of unlabeled data associated with the costly procedures required to obtain labeled data has made semi-supervised learning approaches a promising solution in many applications. At the same time, unsupervised techniques also have been attracting a lot of attention of research community. In this scenario, taking advantage of the recent developments in unsupervised approaches for making new advances in semi-supervised learning tasks can represent a great research venue.

In this paper, we exploit this investigation line by proposing a novel semi-supervised classifier based on a manifold learning approach and a correlation graph, inspired by a recent unsupervised approach [10]. A new semi-supervised similarity measure is proposed and a voting strategy based on the graph structure is used to perform classification tasks. The method was evaluated considering various datasets and comparisons with diverse supervised and semi-supervised methods. The results are positive in most of datasets, achieving error rates comparable to baseline methods and some remarkable results, as the high-accuracy rate of 93.28% with only 2% of training data on the well-known Iris dataset.

We believe that the work represents an important contribution regarding the possibility of exploiting the recent advances obtained by graph and rank-based approaches [34], [35], [11] in unsupervised scenarios for semi-supervised learning tasks. As future work, we intend to investigate other correlation measures and different voting strategies.

Acknowledgment

The authors are grateful to the São Paulo Research Foundation - FAPESP (grants 2017/02091-4, 2016/05669-4, and 2013/08645-0).

REFERENCES

[1] W. P. Amorim, A. X. Falcão, J. P. Papa, and M. H. Carvalho, "Improving semi-supervised learning through optimum connectivity," *Pattern Recognition*, vol. 60, pp. 72 – 85, 2016.

[2] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*, ser. Adaptive Computation and Machine Learning. Cambridge, MA: The MIT Press, 2006.

[3] X. Zhu, "Semi-supervised learning literature survey," Computer Sciences, University of Wisconsin-Madison, Tech. Rep. 1530, 2005.

[4] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proceedings of the Eighteenth International Conference on Machine Learning*. San Francisco: Morgan Kaufmann, 2001, pp. 19–26.

[5] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and semisupervised learning on large graphs," in *Conference on Learning Theory*. Springer, 2004, pp. 624–638.

[6] F. Breve, L. Zhao, M. Quiles, W. Pedrycz, and J. Liu, "Particle competition and cooperation in networks for semi-supervised learning," vol. 24, no. 9, pp. 1686 –1698, sept. 2012.

[7] T. H. Cupertino and L. Zhao, "Semi-supervised learning using random walk limiting probabilities," in *Advances in Neural Networks – ISNN 2013*, 2013, pp. 395–404.

[8] X. Zhu, "Semi-supervised learning with graphs," Ph.D. dissertation, Pittsburgh, PA, USA, 2005, aAI3179046.

[9] X. Yang, X. Bai, L. J. Latecki, and Z. Tu, "Improving shape retrieval by learning graph transduction," in *ECCV*, vol. 4, 2008, pp. 788–801.

[10] D. C. G. Pedronette and R. da S. Torres, "A correlation graph approach for unsupervised manifold learning in image retrieval tasks," *Neurocomputing*, vol. 208, no. Sup C, pp. 66 – 79, 2016.

[11] D. C. G. Pedronette, F. M. F. Gonçalves, and I. R. Guilherme, "Unsupervised manifold learning through reciprocal knn graph and connected components for image retrieval tasks," *Pattern Recognition*, vol. 75, pp. 161 – 174, 2018.

[12] W. He, X. Cheng, R. Hu, Y. Zhu, and G. Wen, "Feature self-representation based hypergraph unsupervised feature selection via low-rank representation," *Neurocomputing*, vol. 253, no. Sup. C, pp. 127 – 134, 2017.

[13] D. C. G. Pedronette, O. A. Penatti, and R. da S. Torres, "Unsupervised manifold learning using reciprocal knn graphs in image re-ranking and rank aggregation tasks," *Image and Vision Computing*, vol. 32, no. 2, pp. 120 – 130, 2014.

[14] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum, "Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[15] L. Ma, H. Li, F. Meng, Q. Wu, and L. Xu, "Manifold-ranking embedded order preserving hashing for image semantic retrieval," *Journal of Visual Communication and Image Representation*, vol. 44, no. Sup C, pp. 29 – 39, 2017.

[16] R. Tarjan, "Depth first search and linear graph algorithms," *SIAM Journal on Computing*, 1972.

[17] V. N. Vapnik, *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[18] K. Q. Weinberger and L. K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, Washington D.C., 2004, pp. 988–995.

[19] J. Sun, S. Boyd, L. Xiao, and P. Diaconis, "The fastest mixing markov process on a graph and a connection to a maximum variance unfolding problem," *SIAM Review*, vol. 48, no. 4, pp. 681–699, 2006.

[20] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computuation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[21] O. Delalleau, Y. Bengio, and N. L. Roux, "Efficient non-parametric function induction in semi-supervised learning," in *Artificial Intelligence and Statistics*, 2005, pp. 96–103.

[22] D. Zhou and B. Schölkopf, *Semi-supervised Learning*. Cambridge, MA: MIT Press, 2006, ch. Discrete Regularization, pp. 237–250.

[23] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Tenth International Workshop on Artificial Intelligence and Statistics*, 2005, pp. 57–64.

[24] T. Joachims, "Transductive learning via spectral graph partitioning," in *Proceedings of International Conference on Machine Learning*. AAAI Press, 2003, pp. 290–297.

[25] O. Chapelle, J. Weston, and B. Schölkopf, "Cluster kernels for semi-supervised learning," in *Advances in Neural Information Processing Systems*, vol. 15, 2003.

[26] A. Corduneanu and T. Jaakkola, *Semi-supervised Learning*. Cambridge, MA: MIT Press, 2006, ch. Data-Dependent Regularization, pp. 163–190.

[27] V. Sindhwani, P. Niyogi, and M. Belkin, "Beyond the point cloud: from transductive to semi-supervised learning," in *ICML '05: Proceedings of the 22nd international conference on Machine learning*. New York, NY, USA: ACM, 2005, pp. 824–831.

[28] C. J. C. Burges and J. C. Platt, *Semi-supervised Learning*. Cambridge, MA: MIT Press, 2006, ch. Semi-Supervised Learning with Conditional Harmonic Mixing, pp. 251–273.

[29] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems*, vol. 16. MIT Press, 2004, pp. 321–328. [Online]. Available: http://www.kyb.tuebingen.mpg.de/bs/people/weston/localglobal.pdf

[30] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," Carnegie Mellon University, Pittsburgh, Tech. Rep. CMU-CALD-02-107, 2002.

[31] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 55–67, Jan. 2008.

[32] F. Breve, L. Zhao, M. Quiles, W. Pedrycz, and J. Liu, "Particle competition and cooperation in networks for semi-supervised learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 9, pp. 1686–1698, Sept 2012.

[33] F. A. Breve and D. C. G. Pedronette, "Combined unsupervised and semi-supervised learning for data classification," in *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, Sept 2016, pp. 1–6.

[34] X. Bai, S. Bai, and X. Wang, "Beyond diffusion process: Neighbor set similarity for fast re-ranking," *Information Sciences*, vol. 325, pp. 342 – 354, 2015.

[35] S. Bai and X. Bai, "Sparse contextual activation for efficient visual re-ranking," *IEEE Trans. on Image Processing (TIP)*, vol. 25, no. 3, pp. 1056–1069, 2016.