COMBINED UNSUPERVISED AND SEMI-SUPERVISED LEARNING FOR DATA CLASSIFICATION

Fabricio Aparecido Breve, Daniel Carlos Guimarães Pedronette

Department of Statistics, Applied Mathematics and Computing (DEMAC)
State University of São Paulo (UNESP)
Rio Claro - Brazil

ABSTRACT

Semi-supervised learning methods exploit both labeled and unlabeled data items in their training process, requiring only a small subset of labeled items. Although capable of drastically reducing the costs of labeling process, such methods are directly dependent on the effectiveness of distance measures used for building the kNN graph. On the other hand, unsupervised distance learning approaches aims at capturing and exploiting the dataset structure in order to compute a more effective distance measure, without the need of any labeled data. In this paper, we propose a combined approach which employs both unsupervised and semi-supervised learning paradigms. An unsupervised distance learning procedure is performed as a pre-processing step for improving the kNN graph effectiveness. Based on the more effective graph, a semi-supervised learning method is used for classification. The proposed Combined Unsupervised and Semi-Supervised Learning (CUSSL) approach is based on very recent methods. The Reciprocal kNN Distance is used for unsupervised distance learning tasks and the semi-supervised learning classification is performed by Particle Competition and Cooperation (PCC). Experimental results conducted in six public datasets demonstrated that the combined approach can achieve effective results, boosting the accuracy of classification tasks.

Index Terms— Semi-Supervised Learning, Unsupervised Learning, Data Classification

1. INTRODUCTION

Semi-Supervised learning is a class of machine learning techniques which is receiving increasing interest in the last decade. These techniques combine both labeled and unlabeled data items in their training process. Therefore, they are usually applied in data sets in which only a small subset of data items may be effectively labeled, due to the high costs and time required in the labeling process [1–3].

The authors would like to thank the São Paulo Research Foundation - FAPESP (grants 2011/17396-9 and 2013/08645-0) and the National Counsel of Technological and Scientific Development - CNPq (grants 475717/2013-9) for the financial support.

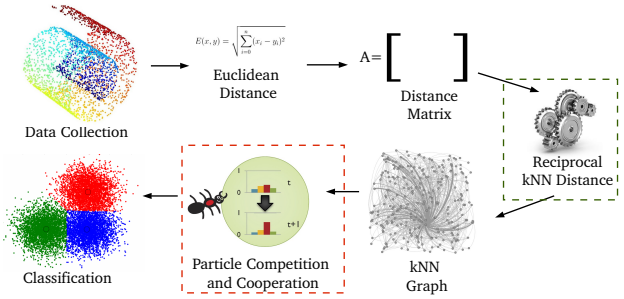

Fig. 1: Overall work-flow of combined method: in green dashed box, the unsupervised distance learning method (Reciprocal kNN Distance); in red dashed box, the semi-supervised learning method used for classification (Particle Competition and Cooperation).

Recently, a new semi-supervised learning approach, known as *Particle Competition and Cooperation* (PCC) was proposed [4]. Particles, which correspond to labeled data, walk in non-weighted and undirected graphs, where each node corresponds to a data item and edges are created between nodes corresponding to similar data items. Particles representing the same class cooperate with each other, at the same time that they compete against particles representing other classes. They aim to dominate the unlabeled nodes, spreading their respective label, and preventing invasion by particles representing other labels. At the end of the process, each unlabeled node is labeled with the label of the particles group that has dominated it.

PCC graphs are usually built by connecting each node to its k-nearest neighbors, according to the Euclidean distance between the corresponding data items [4–6]. However, the graph formation step has major impact in the final classification accuracy in many graph-based approaches, including the PCC. Recently, it has been shown that different distance measures affects the PCC classification results, and the more appropriate distance measure depends on the input data set [7]. Therefore, in order to achieve better classification accuracy, it is indispensable to put research efforts on methods for building a better k-NN graph. An effective approach to build a more effective graph consists in using an unsupervised distance learning step, which do not require any extra training

978-1-5090-0746-2/16/$31.00

information.

Thus, this paper proposes a novel approach named *Combined Unsupervised and Semi-Supervised Learning* (CUSSL), which aims at exploiting both unsupervised and semi-supervised paradigms for a better classification. A recent unsupervised distance learning approach based on the *Reciprocal kNN Distance* [8] is used to compute a more effective graph. Next, the computed graph is used by a semi-supervised PCC method in classification tasks. To the best of our knowledge, this is the first attempt to combine semi-supervised classification methods with unsupervised distance learning approaches. Computer simulations are performed on some artificial and real-world data sets. The high classification accuracies obtained demonstrate that the proposed approach can improve the effectiveness of the regular *Particle Competition and Cooperation* (PCC) method.

The paper is organized as follows. Section 2 presents the proposed combined approach. Section 3 describes the Reciprocal kNN Distance used for unsupervised distance learning tasks. Section 4 describes the PCC method used on semi-supervised learning step classification. Section 5 presents the experimental evaluation and the results of computer simulations. Finally, Section 6 discusses the conclusions.

2. COMBINED UNSUPERVISED AND SEMI-SUPERVISED LEARNING (CUSSL)

In the proposed approach, an unsupervised distance learning procedure is performed as a pre-processing step for the graph construction. Diverse methods have been proposed in order to improve the effectiveness of distance measures in an unsupervised manner [9–17]. The main motivation of such initiatives consists in to exploit the intrinsic dataset structure for computing a more effective distance measure among collection objects, without the need of any training data. Generally, the classic distance measures are replaced by more global measures in order to obtain the k nearest neighbors. In fact, data samples are often modeled as high dimensional points in an Euclidean space. However, the data samples often live in a much lower-dimensional intrinsic space. Consequently, capturing and exploiting the manifold structure constitute a central problem for effective distance computation [12].

Recently, an unsupervised distance learning approach, known as *Reciprocal kNN Distance*, was proposed to provide a more effective distance measure in image retrieval scenarios [8]. It takes into account the intrinsic dataset structure by analyzing the reciprocal references at top rank positions. The modelling in terms of rank information enables its use in many other scenarios, specially in cases which require the the computation of k nearest neighbors, as the PCC.

Figure 1 illustrates the overall work-flow of the *Combined Unsupervised and Semi-Supervised Learning* approach. The *Reciprocal kNN Distance* consider the Euclidean distance as input and computes a new and more effective distance measure between pair of data items. In this step, no labeled data is required. In the following, the new computed distance is used to build a graph in which the PCC based algorithm is then applied for classification through semi-supervised learning. In this way, although independent from each other, the two methods are combined to achieve better classification accuracy. A formal definition of learning models considered by the proposed approach is presented in next section.

2.1. Unsupervised and Semi-Supervised Learning Models

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.

The objective of the *unsupervised distance learning* step consists in redefining the distance d by computing a more effective distance function d_r. In a math notation: d_r: $\mathfrak{X} \times \mathfrak{X} \times \mathcal{T} \rightarrow \mathbb{R}$ is a distance function between two data items $x_i, x_j \in \mathfrak{X}$ that considers the rank information encoded in the set of ranked lists $\mathcal{T}$. For the unsupervised distance learning procedure, all data items from the collection $\mathfrak{X}$ are considered, both labeled and unlabeled items. However, no label information is exploited.

The *semi-supervised learning* step uses the information encoded in the distance function d_r for estimating the label of unlabeled data items. 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 $x_i \in \mathfrak{X}_U$.

The next sections describe the Reciprocal kNN Distance and the PCC method, used for the unsupervised and semi-supervised steps, respectively.

3. RECIPROCAL KNN DISTANCE

The Reciprocal kNN Distance [8] models a data collection in terms of ranking information, defining a more effective distance measure by analyzing the reciprocal references at top ranked positions. The ranked lists define relationships not only between pairs of objects as distance functions, but considering among all objects found in a ranked list [8, 18].

Additionally, the k-reciprocal nearest neighborhood relationship is a much stronger indicator of similarity than the unidirectional nearest neighborhood, mitigating the risk of false positives at top positions of ranked lists [8, 19]. The *Reciprocal kNN Distance* is formally described in follow sections.

3.1. Reciprocal Neighborhood

Given a data item x_q, we can define a neighborhood set that contains the k most similar data items to x_q as $\mathcal{N}(q,k)$. For the k-nearest neighbor query, we have $|\mathcal{N}(q,k)| = k$, which is formally defined as follows:

$$\mathcal{N}(q,k) = \{\mathcal{S} \subseteq \mathfrak{X}, |\mathcal{S}| = k \wedge \forall x_i \in \mathcal{S}, x_j \in \mathfrak{X} - \mathcal{S} : \tau_q(i) < \tau_q(j)\}. \quad (1)$$

The nearest neighbor relationships are not symmetric [17, 19], i.e., $x_i \in \mathcal{N}(q,k)$ does not imply $x_q \in \mathcal{N}(i,k)$. The k-reciprocal neighborhood set of a data item x_q can be defined [19] as:

$$\mathcal{N}_r(q,k) = \{x_i \in \mathcal{N}(q,k) \wedge x_q \in \mathcal{N}(i,k)\}. \quad (2)$$

Based on the reciprocal neighborhood set $\mathcal{N}_r(q,k)$, a binary function $f_r : \mathfrak{X} \times \mathfrak{X} \rightarrow \{0,1\}$ is defined for determining if two data items $x_q, x_i \in \mathfrak{X}$ are reciprocal neighbors:

$$f_r(q,i) = |\mathcal{N}_r(q,k) \cap \{x_i\}|. \quad (3)$$

The function f_r is defined as 1 if data items x_q and x_i are reciprocal neighbors, and 0 otherwise.

3.2. Reciprocal kNN Distance

The *Reciprocal kNN Distance* between two data items $x_q, x_i \in \mathfrak{X}$ is computed based on the number of reciprocal neighbors at top positions of ranked lists $\tau_q, \tau_i \in \mathcal{T}$. Additionally, a weight for each pair of reciprocal neighbors is considered, proportionally to their position in the ranked lists τ_q and τ_i. In this way, the incidence of reciprocal neighbors at top positions of ranked lists is considered more relevant. The score based on the number of reciprocal neighbors and its respectively weights are given by the function $n_r(q,i)$, defined as follows:

$$n_r(q,i) = \frac{\sum_{j \in \mathcal{N}(q,k)} \sum_{l \in \mathcal{N}(i,k)} f_r(j,l) \times w_r(q,j) \times w_r(i,l)}{k^4}, \quad (4)$$

While the function f_r determines if a pair of items (o_j, o_l) are reciprocal neighbors, the weight is computed based on position of these items in the ranked lists τ_q and τ_i, according to the function w_r, defined as follows:

$$w_r(q,j) = k + 1 - \tau_q(j). \quad (5)$$

The value of w_r is linearly decreasing, ranging from k assigned to the first position to 1, at the kth position. The divisor k^4 in Equation 4 is defined considering the maximum value of reciprocal neighbors (k^2) and the maximum values of w_r. The *Reciprocal kNN Distance* is defined as the inverse of the number of reciprocal neighbors n_r, as follows:

$$d_r(q,i) = \frac{1}{1 + n_r(q,i)}. \quad (6)$$

A constant L is introduced aiming at limiting the computational complexity of the distance learning procedure. Once the top positions of ranked lists are expected to contain the most relevant data items, the distance learning can be performed considering only the beginning of the ranked lists. Therefore, the ranked lists are processed only until the L position, keeping its complexity of $O(N)$. In this way, the parameter L establishes a trade-off between effectiveness and efficiency, specially for large scale datasets. The function d_r is redefined for consider the value of L as follows:

$$d_r(q,i) = \begin{cases} \frac{1}{1+n_r(q,i)}, & \textit{if } \tau_q(i) \leq L, \\ \tau_q(i), & \textit{otherwise}. \end{cases} \quad (7)$$

Finally, a new set of k-nearest neighbors $\mathcal{N}_{rd}(q,k)$ can be obtained based on the *Reciprocal kNN Distance*. The set is formally defined as follows:

$$\mathcal{N}_{rd}(q,k) = \{\mathcal{S} \subseteq \mathfrak{X}, |\mathcal{S}| = k \wedge \forall x_i \in \mathcal{S}, x_j \in \mathfrak{X} - \mathcal{S} : d_r(q,i) \leqslant d_r(q,j)\}. \quad (8)$$

The set is obtained aiming at computing the k-NN graph used by the semi-supervised learning step. An undirected graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ is generated from the data collection $\mathfrak{X}$, in which $\mathbf{V} = \{v_1, v_2, \ldots, v_N\}$ is the set of nodes, and $\mathbf{E}$ is the set of edges (v_i, v_j). Each node v_i corresponds to a data item x_i. Two nodes v_i and v_j are connected if $v_j \in \mathcal{N}_{rd}(i,k)$ or $v_i \in \mathcal{N}_{rd}(j,k)$.

4. PARTICLE COMPETITION AND COOPERATION

Overall, the semi-supervised learning particle competition and cooperation approach [4] may be described as follows. Each graph node in an undirected and non-weighted graph represents a data item. Edges are created between nodes representing similar data items. A particle is created for each labeled node. Particles will walk through the nodes, following a random-greedy rule to select the next node to visit among the neighbors of the current node. Particles corresponding to nodes with the same label belong to the same team, thus they cooperate with each other to dominate the unlabeled nodes. Particles in different teams compete against each other for the possession of the nodes. Each node has a set of domination levels, each of them representing a team of particles. Particles increase their team domination level in the nodes they visit, at the same time that they decrease other teams domination levels. At the end of the iterative process, each node is labeled accordingly to the class of the team which has the highest domination level on it.

Formally, for each node $v_i \in \{v_1, v_2, \ldots, v_L\}$ of the graph $\mathbf{G}$, a particle ρ_i is generated and its initial position is set to v_i. Each particle ρ_j has a variable $\rho_j^{\omega}(t) \in [0,1]$ equivalent to the particle strength, which defines how much

a particle impacts a node it is currently visiting. Particles always begin with their maximum strength, $\rho_j^{\omega}(0) = 1$. Each particle ρ_j also has a distance table, $\rho_{\mathbf{j}}^{\mathbf{d}}(\mathbf{t}) = \{\rho_j^{d_1}(t), \rho_j^{d_2}(t), \ldots, \rho_j^{d_n}(t)\}$, where each element $\rho_j^{d_i}(t) \in [0, \quad n-1]$ corresponds to the distance measured between the particle's initial node v_j and any node v_i. The distance table is dynamically updated as the corresponding particle walks.

Each node v_i has a domination vector $\mathbf{v}_{\mathbf{i}}^{\omega}(\mathbf{t})$, where each element $v_i^{\omega_c}(t) \in [0, 1]$ corresponds to the domination level from the team/class c over the node v_i. The sum of the domination levels in each node is always constant, $\sum_{c=1}^{C} v_i^{\omega_c} = 1$.

In the nodes corresponding to labeled data items, the domination levels are fixed. They are set to have complete domination by the corresponding class and none for the others. On the other hand, nodes corresponding to unlabeled data items have variable domination levels. They begin with all classes domination levels set equally, but particles change these levels when they pay a visit. Thus, for each node v_i, the domination vector $\mathbf{v}_{\mathbf{i}}^{\omega}$ is set as follows:

$$v_i^{\omega_c}(0) = \begin{cases} 1 & \text{if } x_i \text{ is labeled and } y(x_i) = c \\ 0 & \text{if } x_i \text{ is labeled and } y(x_i) \neq c \\ \frac{1}{C} & \text{if } x_i \text{ is unlabeled} \end{cases} . \quad (9)$$

Every time a particle ρ_j visits any unlabeled node v_i, the node domination levels are updated as follows:

$$v_i^{\omega_c}(t+1) = \begin{cases} \max\{0, v_i^{\omega_c}(t) - \frac{\Delta_v \rho_j^{\omega}(t)}{C-1}\} \\ \quad \text{if } c \neq y(\rho_j) \\ v_i^{\omega_c}(t) + \sum_{r \neq c} v_i^{\omega_r}(t) - v_i^{\omega_r}(t+1) \\ \quad \text{if } c = y(\rho_j) \end{cases} , \quad (10)$$

where $0 < \Delta_v \leq 1$ is a parameter to control the changing rate, and $y(\rho_j)$ represents the class of particle ρ_j. A particle ρ_j changes the node v_i it is visiting by increasing the domination level of its class ($v_i^{\omega_c}$, $c = y(\rho_j)$) at the same time that it decreases the domination levels of other classes ($v_i^{\omega_c}$, $c \neq y(\rho_j)$). Since labeled nodes have fixed domination levels, (10) does not apply to them.

A particle may get stronger or weaker according to the domination level of its class in the node it is currently visiting. At each iteration, the particle strength is updated, $\rho_j^{\omega}(t) = v_i^{\omega_c}(t)$, where v_i is the node being visited, and $c = y(\rho_j)$.

At each iteration, a particle ρ_j chooses a node v_i to visit among the neighbors of its current node. The probability of choosing a node v_i is given by: a) the particle class domination on it, $v_i^{\omega_c}$, and b) the inverse of its distance, $\rho_j^{d_i}$, as follows:

$$p(v_i|\rho_j) = (1 - p_{\text{grd}}) \frac{W_{qi}}{\sum_{\mu=1}^{n} W_{q\mu}} + p_{\text{grd}} \frac{W_{qi} v_i^{\omega_c} (1 + \rho_j^{d_i})^{-2}}{\sum_{\mu=1}^{n} W_{q\mu} v_{\mu}^{\omega_c} (1 + \rho_j^{d_\mu})^{-2}}, \quad (11)$$

where q is the index of the node being visited by particle ρ_j, c is the class label of particle ρ_j, $W_{qi} = 1$ if there is an edge between the current node and the node v_i, and $W_{qi} = 0$ otherwise. A particle stays on the chosen node only if, after applying (10), its class domination level is the largest on that node; otherwise, a shock happens and the particle goes back to the previous node and stays there until the next iteration. $0 \leq p_{\text{grd}} \leq 1$ controls the balance between randomness and greediness in the probabilities.

For further details on the original particle competition and cooperation algorithm, see [4].

Table 1: Classification Accuracy on the Iris data set with 2% to 10% labeled samples

Labeled	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%**

5. COMPUTER SIMULATIONS

In this section, computer simulation using some artificial and real-world data sets are presented in order to show the effectiveness of the proposed method. For each data set, we applied both the original PCC method and the proposed method (CUSSL). For PCC, the parameter k defines the size of k-neighborhood (amount of nearest neighbors) for the graph construction using the Euclidean distance (L2). For CUSSL, two parameters are considered: (*i*) the size of k-neighborhood used by the unsupervised distance learning algorithm, referred in this section as k_r; and (*ii*) the size of k-neighborhood for the graph construction, referred in this section as k_n.

The parameters $p_{\text{grd}} = 0.5$ and $\Delta_v = 0.1$ are fixed in all experiments with both PCC and CUSSL. These values were chosen because they were used in previous PCC applications [5, 6, 20]. Notice that one could still optimize p_{grd} and Δ_v in each experiment to further increase classification accuracy in both PCC and CUSSL. The other parameters were chosen by performing a search over the following grid: $k, k_r, k_n \in \{20, 30, 40, 50, 60, 70\}$. The parameter L was defined as suggested by [8].

Tables 1 and 2 show the classification accuracy when PCC and CUSSL are applied to the Iris data set and to the Wine data set [21], respectively. 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. These results are plotted on Figures 2 and 3.

Tables 3 and 4 show the classification accuracy when PCC and CUSSL are applied to Digit1, COIL, USPS, and g241c

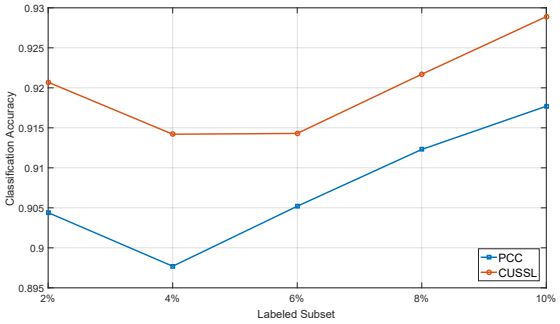

Fig. 2: Classification Accuracy on the Iris data set with 2% to 10% labeled samples

Table 2: Classification Accuracy on the Wine data set with 2% to 10% labeled samples

Labeled	2%	4%	6%	8%	10%
PCC	93.24%	92.54%	93.26%	94.13%	94.89%
CUSSL	**94.17%**	**94.84%**	**95.13%**	**95.39%**	**95.74%**

data sets [1] [2]. For each data set, there are 12 subsets of 10 labeled data points and 12 subsets of 100 labeled data points, which were randomly chosen and provided by [2]. Table 3 shows the results for the 10 labeled data points subsets and Table 4 shows the results for the 100 labeled data points subsets. For each graph configuration, both algorithms are applied using each of the 12 labeled subsets. For each subset, the experiment is repeated 100 times. Therefore, each value shown in these tables is the average of 1200 executions, i.e. 100 in each of the 12 labeled subsets.

Figure 4 shows the classification accuracy achieved by CUSSL on Digit1, COIL, USPS, and g241c data sets with 10 labeled with different combinations of k_r and k_n.

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

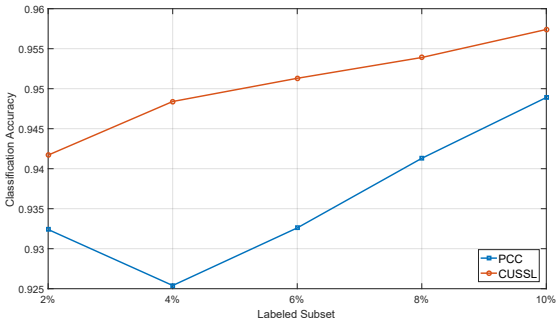

Fig. 3: Classification Accuracy on the Wine data set with 2% to 10% labeled samples

Table 3: Classification Accuracy on Digit1, COIL, USPS, and g241c data sets with 10 labeled samples

Dataset	Digit1	COIL	USPS	g241c	Mean
PCC	86.91%	39.30%	80.06%	56.96%	65.77%
CUSSL	**87.70%**	**40.76%**	**82.81%**	**59.51%**	**67.28%**

Table 4: Classification Accuracy on Digit1, COIL, USPS, and g241c data sets with 100 labeled samples

Dataset	Digit1	COIL	USPS	g241c	Mean
PCC	97.31%	74.21%	93.59%	**73.87%**	84.73%
CUSSL	**97.48%**	**76.87%**	**95.19%**	73.85%	**85.00%**

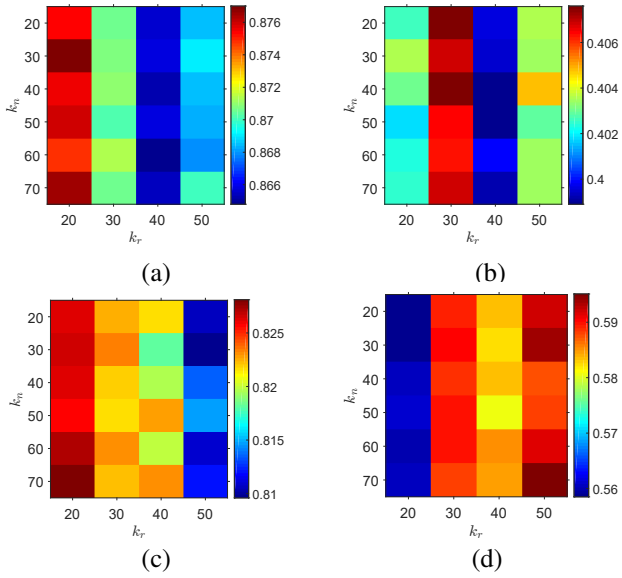

Fig. 4: Classification accuracy achieved by CUSSL with different combinations of k_r and k_n on Digit1, COIL, USPS, and g241c data sets with 10 labeled samples

By analyzing Tables 1 to 4, one can notice that, in most of the covered scenarios, CUSSL achieved higher classification accuracy than the PCC method, demonstrating the potential of the combined approach.

The graph construction step has the highest order of computational complexity, and it is $O(NlogN)$ in both methods, where N is the amount of data items. The iterative steps is only $O(N)$, as shown in [4]. Therefore, PCC and CUSSL have the same computational complexity.

6. CONCLUSION

A combined approach including unsupervised and semi-supervised methods is discussed in this paper. The proposed approach performs an unsupervised distance learning step, without the need of any labeled or training data, through the Reciprocal kNN Distance. The objective consists in exploiting the intrinsic dataset structure for improving the distance among data items. Subsequently, a k-NN graph is computed based on the learned distance and used as input for a semi-supervised learning step. The semi-supervised learning step is based on the Particle Competition and Cooperation approach. It combines both labeled and unlabeled data items in its training process.

An experimental evaluation was conducted considering six public datasets, including artificial and real-world data

sets. The computer simulations also considered various different size of labeled sets used in the training procedure. The vast majority of experimental results demonstrated the benefits of the combined approach, CUSSL, in comparison to the original PCC method.

Future work focuses on: (*i*) the use of other unsupervised distance learning approaches as a pre-processing step for semi-supervised learning classification; and (*ii*) the investigation of feature selection to be applied before the unsupervised distance learning procedure.

7. REFERENCES

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

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

[3] X. Zhu, "Semi-supervised learning," in *Encyclopedia of Machine Learning* (C. Sammut and G. Webb, eds.), pp. 892–897, Springer US, 2010.

[4] F. Breve, L. Zhao, M. Quiles, W. Pedrycz, and J. Liu, "Particle competition and cooperation in networks for semi-supervised learning," *IEEE Trans. Knowl. Data Eng.*, vol. 24, pp. 1686 –1698, sept. 2012.

[5] F. A. Breve, L. Zhao, and M. G. Quiles, "Particle competition and cooperation for semi-supervised learning with label noise," *Neurocomputing*, vol. 160, pp. 63 – 72, 2015.

[6] F. Breve, M. Goncalves Quiles, and L. Zhao, "Interactive image segmentation using particle competition and cooperation," in *Neural Networks (IJCNN), 2015 International Joint Conference on*, pp. 1–8, July 2015.

[7] L. Guerreiro and F. A. Breve, "Analysis of the influence of distance metrics on the semi-supervised algorithm of particle competition and cooperation," in *Proceedings - XI Workshop de Visão Computacional WVC'2015*, pp. 77–82, 2015.

[8] D. C. G. Pedronette, O. A. B. Penatti, R. T. Calumby, and R. da S. Torres, "Unsupervised distance learning by reciprocal knn distance for image retrieval," in *Proceedings of International Conference on Multimedia Retrieval*, ICMR '14, pp. 345–352, 2014.

[9] X. Yang, S. Koknar-Tezel, and L. J. Latecki, "Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval.," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2009)*, pp. 357–364, 2009.

[10] D. C. G. Pedronette and R. da S. Torres, "Unsupervised manifold learning by correlation graph and strongly connected components for image retrieval," in *International Conference on Image Processing (ICIP'2014)*, 2014.

[11] X. Yang, L. Prasad, and L. Latecki, "Affinity learning with diffusion on tensor product graph," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 28–38, 2013.

[12] J. Jiang, B. Wang, and Z. Tu, "Unsupervised metric learning by self-smoothing operator," in *International Conference on Computer Vision (ICCV'2011)*, pp. 794–801, 2011.

[13] D. C. G. Pedronette and R. da S. Torres, "Exploiting pairwise recommendation and clustering strategies for image re-ranking," *Information Sciences*, vol. 207, pp. 19–34, 2012.

[14] X. Yang, X. Bai, L. J. Latecki, and Z. Tu, "Improving shape retrieval by learning graph transduction," in *European Conference on Computer Vision (ECCV'2008)*, vol. 4, pp. 788–801, 2008.

[15] 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.

[16] M. Donoser and H. Bischof, "Diffusion processes for retrieval revisited," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2013)*, pp. 1320–1327, 2013.

[17] H. Jegou, C. Schmid, H. Harzallah, and J. Verbeek, "Accurate image search using the contextual dissimilarity measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 2–11, 2010.

[18] D. C. G. Pedronette and R. da S. Torres, "Image re-ranking and rank aggregation based on similarity of ranked lists," *Pattern Recognition*, vol. 46, no. 8, pp. 2350–2360, 2013.

[19] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. van Gool, "Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2011)*, pp. 777 –784, june 2011.

[20] F. Breve, "Active semi-supervised learning using particle competition and cooperation in networks," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pp. 1–6, Aug 2013.

[21] M. Lichman, "UCI machine learning repository," 2013.