International Journal of Engineering

Journal Homepage: www.ije.ir

A Sample Dependent Decision Fusion Algorithm for Graph-based Semi-supervised Learning

A. Namjoy, A. Bosaghzadeh*

Faculty of Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran

PAPER INFO

Paper history:
Received 15 February 2020
Received in revised form 01 April 2020
Accepted 01 April 2020

Keywords:
Affinity Graph
Decision Fusion
Label Propagation
Multiple Features

ABSTRACT

On many occasions, the evaluation of a phenomenon based on a single feature could not solely be resulted in comprehensive and accurate results. Moreover, even if we have several features, we don't know in advance, which feature offers a better description of the phenomenon. Thus, selecting the best features and especially their combination could lead to better results. An affinity graph is a tool that can describe the relationship between the samples. In this paper, we proposed a graph-based sample-based ranking method that sorts the graphs based on six proposed parameters. The sorting is performed such that the graphs at the top of the list have better performance compared to the graphs at the bottom. Furthermore, we propose a fusion method to merge the information of various features and improve the accuracy of label propagation. Moreover, a method is proposed for parameter optimizations and the ultimate decision fusion. The experimental results indicate that the proposed scheme, apart from correctly ranking the graphs according to their accuracy, in the fusion step, increases the accuracy compared to the use of a single feature.

***doi**: 10.5829/ije.2020.33.05b.35*

1. INTRODUCTION

Recent technologies enabled us to easily gather diverse types of data. To create a big picture out of these different types of data for a specific problem (such as illness, biological process, or face recognition) we need to combine these types of data. One of the solutions is graph-based fusion which is obtained by creating a similarity matrix for each available type of data and then fusing them to have a complete spectrum of data [1]. In recent years, graph-based semi-supervised learning methods draw lots of attention in machine learning mainly, because of their flexibility and straightforward implementations [2–10].

A graph is represented by three components G = (V, **E**, **W**) where V is a set of vertices (nodes) based on a dataset of **X**={x_i, i= 1, …, N} where N is the number of samples, E is a set of edges with the dimension of N × N, and **W** is an N × N undirected weighting affinity matrix which represents the similarity between samples. If the weight of an edge is larger than zero, it indicates that there is an edge between two vertices; otherwise, there is no edge between the two vertices [11].

A constructed graph alone cannot lend itself to an easy assessment and is not the final goal in machine learning. Graph-based label propagation is one of the well-known methods that adopt one or several graphs for the task of classification [2, 9, 12]. It essentially propagates the label of a vertex to its neighboring vertex based on the similarities coded in the affinity graph (i.e., **W**).

The adopted graph plays an important role in the obtained result of label propagation. Indeed, given a real dataset as well as a machine learning task that uses this dataset, very often, knowing in advance the ideal graph for that dataset and for that task is a challenging task, if not unfeasible. Furthermore, graph construction methods have several parameters that need to be tuned. Moreover, some methods attempt to learn different metrics based on different feature descriptors [13–16].

*Corresponding Author Institutional Email: *a.bosaghzadeh@sru.ac.ir* (A. Bosaghzadeh)

Please cite this article as: A. Namjoy, A. Bosaghzadeh, A Sample Dependent Decision Fusion Algorithm for Graph-Based Semi-Supervised Learning, International Journal of Engineering (IJE), IJE TRANSACTIONS B: Applications Vol. 33, No. 5, (May 2020) 1010-1019

Thus, in many cases, the best solution is to construct several graphs and then fuse them such that the resulted graph possesses useful information about each graph.

This fusion can be performed either linearly like Sparse Multiple Graph Integration (SMGI) [17], deep graph fusion [18], and Multi-view Local Global Consistency [19] or non-linearly like Similarity Network Fusion (SNF) [1], Multi-modality Dynamic Label Propagation (MDLP) [11], and Nonlinear Graph Fusion [20].

Wang et al. [1] proposed a method called SNF. This method was utilized to combine the information of DNA, mRNA expression, and microRNA (miRNA) for five datasets of cancer. Their results have shown that their information fusion has significantly better performance than using one simple type of data. In SNF, feature selection and combination are done simultaneously. The combination algorithm transforms the graphs in a way that each strong (weak) connection in many graphs is reinforced (weakened).

Karasuyama and Mamitsuka [17] proposed a method for graph combination called SMGI, forcing sparsity on the graph fusion coefficients. Based on their claim, these sparse coefficients are not reachable by any other means. The basic idea is that among the created graphs, there might be some irrelevant and noisy information which make some graphs less important than others. These cases can be handled by controlling the coefficients of the graphs. The sparsity property has two advantages:

1. Improving the prediction accuracy by removing irrelevant and noisy graphs
2. Interpretable results, which means we could easily find informative graphs.

Their experimental results indicated that the advantage of their model is not only in increasing the prediction accuracy but also in finding informative graphs. Their results and their comparisons show that the unique property of their method is the sparse weights. Meaning that there are few non-zero graph weights and many of graph weights are exactly zero.

An et al. [19] extended the LGC [12] algorithm such that it learns from several graphs. They constructed one graph for each adopted feature descriptor and then combined them linearly with equal weights to form the fused graph. The resulted graph is then fed into the LGC method to estimate the class labels for the task of person identification.

Lin et al. proposed a Dynamic Graph Fusion Label Propagation (DGFLP) method [21] that merged the information of features and labels. For each constructed graph, the DGFLP method assigned unequal adaptive weights according to the information available in each graph, while for the label space graph, they adopted a fixed value.

The adoption of unequal weights in DGFLP and SMGI algorithms is motivated by the fact that the graphs that are constructed from various features and different graph construction algorithms do not have the same importance and information. Furthermore, in SMGI they show that the graphs constructed by varying the parameter of the graph construction technique can affect the information of the constructed graph.

One of the main drawbacks of the above-mentioned methods is that they are not sample dependent, meaning that they adopt a fixed fused graph for all samples. However, this may not be the best solution, since, for each sample, a different combination of the graphs might be a good option. For instance, in the task of person identification, we can have graphs constructed based on fingerprints [22] and face [23] features. While for some people their fingerprint could be more discriminative, for some people their face might be more discriminative for identification. Hence, a better solution is that for each identity, separately decide which graph or combination of the graphs could be more discriminative.

In this paper, we propose a sample dependent graph ranking method and a weighted decision fusion algorithm based on the information in each graph. The ranking algorithm sorts the graphs based on six weighted information extracted from the graphs. The ranking method sorts the graphs such that the graphs at the top of the list have the potential to provide accurate label in the label propagation task compared to the graphs at the bottom of the list. The graphs at the top are then employed for a combination phase, in which the results obtained from these graphs are then merged. In the decision fusion phase, we merge the decisions obtained from the most informative graphs and then decide about a sample label.

The rest of this article is organized as follows: In Section 2 we explain the proposed method. Section 3 is dedicated to the experimental results, and Section 4 concludes the article.

2. PROPOSED METHOD

2. 1. Preliminaries

Imagine we have N images in a classification task, where L out of N images are labeled and the remaining U=N-L images are unlabeled. For each image, we extract V features, hence, the feature vector will be $\left\{x_1^{(v)}, \dots, x_N^{(v)}\right\}_{v=1}^{V}$ where $x_1^{(v)}$ is the v^{th} feature vector with the dimensionality of D^v . Matrix $Y \in \mathbb{R}^{N \times C}$ contains the labels of the N samples for C classes. For each sample, we have a vector of $1 \times C$ which contains 1 for the corresponding class and zero elsewhere. Without loss of generality, we sort the data such that the labeled data comes first and then we put the data without the label. Hence, the data matrix will be like $\mathrm{X} = [\mathrm{x}_L\ ,\ \mathrm{x}_\mathrm{U}]$ where x_L and x_U represent the labeled and unlabeled data, respectively. Consequently, the adjacency matrix which shows the similarity between the data will have the following shape:

$$W = \begin{bmatrix} w_{ll} & w_{lu} \\ w_{ul} & w_{uu} \end{bmatrix}$$

where w_{ll} represents the similarity between the labeled samples, w_{lu} shows the similarity of the labeled sample with unlabeled samples, w_{ul} demonstrates the similarity of unlabeled samples respect to labeled ones and w_{uu} represents the similarity between unlabeled samples.

The graph can then be used for label propagation where the goal is to propagate the label of labeled nodes via an affinity graph such that the close nodes get similar labels. Mathematically, this can be considered as an optimization problem as

$$\min \sum_{i,j=1}^{n} W_{ij} \, \| f_i - f_j \|_2^2 \quad (1)$$

where W_{ij} is the similarity between $\mathrm{x_i}$ and $\mathrm{x_j}$ and f_i and f_j are the predicted labels of $\mathrm{x_i}$ and $\mathrm{x_j}$, respectively. The Gaussian Fields and Harmonic Functions (GFHF) [24] method adopts this function (i.e., Equation (1)) for label propagation. On the other hand, the LGC [25] method uses a second term with which it tries to minimize the distance between the predicted labels and real labels of labeled samples. The formulation of LGC is as:

$$\min \frac{1}{2}\left(\sum_{i,j=1}^{N} W_{ij} \, \| \frac{f_i}{\sqrt{D_{ii}}} - \frac{f_j}{\sqrt{D_{jj}}} \|_2^2 + \mu \sum_{i=1}^{L} \| f_i - y_i \|_2^2 \right) \quad (2)$$

where y_i is the real label and f_i is the predicted label of $\mathrm{x_i}$, $\mu > 0$ is the regularization parameter, and $\mathrm{D_{ii}}$ is the summation of the $\mathrm{i^{th}}$ row of the W matrix.

2. 2. Algorithm of the Proposed Method

In this section, at first, we explain the preprocessing applied on the graphs, then we describe the proposed scores that we adopt for graph ranking and finally, we explain the proposed fusion technique.

As we observed in Equations (1) and (2), the labels got spread according to the graph weights that connect the nodes. Consequently, the graph plays a critical role in the result of label propagation. Since in the label propagation algorithms the labels are spread through the edges, a wrong edge that connects two nodes that belong to two different classes can cause wrong labeling. Therefore, it is very important to eliminate these wrong edges in order to increase labeling performance. In other words, there must be no connection between the samples that belong to two different classes. In other words, there must be no edge between two nodes of two different classes. Since we do not know the label of unlabeled samples, we cannot eliminate the wrong edges for them. However, this process could be readily done for labeled data, in w_{ll} submatrix, where we have the true label of all nodes. Hence, we set to zero the weight of edges between the nodes that correspond to different classes.

The second step in the proposed method is to sort the graphs with the help of six parameters. In the following, we explain each of the parameters adopted to evaluate the graphs.

As we stated earlier in the article, the wrong edges in a graph can cause wrong labeling. On the other hand, correct edges (i.e., the edges that connect samples from the same class) can enhance the performance. The more correct edges in a graph, the better it performs in the label propagation. To determine how well the same class nodes are connected in each graph, we select w_{ll} subgraph, where we have the label of nodes and evaluate its edges that connect the same class nodes. We consider $\mathrm{w_{ll}}$ as a good representation of the whole graph. In other words, if the same class nodes are well connected (high similarity) in the w_{ll} submatrix, one can expect that it happens in the whole graph.

Therefore, we define the summation of intraclass weights (SWICW) for the class c as the first parameter and calculate it as

$$\mathrm{SWICW_c} = \sum_{i=1}^{N_c} \sum_{j=1}^{N_c} W_{ij}^{c} \quad (3)$$

where N_c is the number of samples in the $\mathrm{c^{th}}$ class and W_{ij}^{c} is the weight (similarity) between the $\mathrm{x_i}$ and $\mathrm{x_j}$ samples in class c. The idea of using this parameter is that a graph with large weights between the samples of the same class is less prone to wrong label propagation.

The other five parameters are based on the accuracy obtained on the labeled data. We define a backward label propagation scheme as follows. First, we propagate the label of labeled samples to the unlabeled samples via GFHF [24] method. Second, we consider the estimated label of unlabeled samples as their true labels and remove the label of labeled samples. It means that the labeled samples are considered as unlabeled and the predicted label of the unlabeled samples is considered as their true label. Third, we propagate the label of unlabeled samples to the labeled ones. Based on the results of backward label propagation on the labeled samples, we calculate the second, third, and forth scores by calculating the maximum, entropy, and standard deviation of membership probability of the class c according to Equations (4)- (6), respectively.

$$MPOB_c = \sum_{i=1}^{N_c} Max(f_i) \quad (4)$$

$$EPOB_c = \sum_{i=1}^{N_c} Entropy(f_i) \quad (5)$$

$$SPOB_c = \sum_{i=1}^{N_c} STD(f_i) \quad (6)$$

where f_i is a vector containing the membership probability of a sample belonging to each class.

The main idea is that the high certainty of detecting a class would promote higher accuracy. Our simulations indicate that the normalized average of these three parameters can represent how well the graphs perform in the label propagation task.

The fifth and sixth scores are based on the accuracy of backward label propagation. The first one (the fifth score) is the accuracy of the label propagation on all of the labeled samples. We assume that if the label propagation on the labeled samples has high accuracy, one can expect that the graph can well predict the label of unlabeled samples too.

The sixth score predicts how well a graph performs in predicting the samples of each class. Since we consider the overall accuracy as an estimation of the overall performance of a graph, we select the class accuracy of a graph as a variable that explains how well the graph predicts the label of samples in a specific class.

After calculating the six parameters explained above, for a labeled sample whose labels have been correctly estimated (in backward label propagation), we normalize the parameters such that their sum equals to one. The normalized parameters are then used as the optimal coefficients of each parameter.

Moreover, we adopt the average number of graphs that correctly propagate the labels as the number of graphs that should be used in the fusion step (denoted by S in Algorithm 1).

Finally, based on the obtained parameters, for each graph and each class, a rank is calculated which indicates the credibility of the opinion of each graph in predicting the samples in each class (Equation (7)).

$$\begin{aligned}
&Rank_c{}^i = \\
&Acc\_total^i \times\ Acc\_total\_Coefficient +\ldots \\
&Acc\_class_c^i \times\ Acc\_class\_Coefficient + \ldots \\
&SWICW_c^i \times SWICW\_Coefficient + \ldots \qquad (7) \\
&MPOB_c^i\ \times MPOB\_Coefficient - \ldots \\
&EPOB_c^i\ \times\ EPOB\_Coefficient + \ldots \\
&SPOB_c^i \times\ SPOB\_Coefficient
\end{aligned}$$

where i is the graph index and c is the class index.

To estimate the label of unlabeled samples, we perform as follows. For each unlabeled sample, we sort the graphs according to the calculated coefficients and select the graphs with high ranking values. Then, for each graph, we perform label propagation to obtain the predicted label vector of the unlabeled sample. Finally, we take the weighted average decision of each selected graph and report it as the label vector of the unlabeled sample. Algorithm 1 summarizes the proposed method, while Algorithm 2 describes the process of calculating the six graph parameters.

Algorithm 1. The proposed sample-dependent decision fusion method

```
Input: A set of graphs based on different features and parameters Wset, the label of labeled data L_L
Output: Label of the unlabeled samples
---------------------------------------------------------------
Score Calculation
1. Set to zero the weight of edges between the samples from different classes ($w_{ll}$).
2. Perform label propagation and backward label propagation for input graphs.
3. Calculate the six scores of the graphs (Algorithm 2).
4. For each sample whose label is correctly estimated in backward label propagation, calculate the coefficients of six parameters based on the normalized average of parameters.
---------------------------------------------------------------
Graph sorting and Decision Fusion
5. For each graph:
    5-1. For each class:
        5-1-1. calculate the rank of each class in each graph (Equation (7)).
6. For each unlabeled sample:
    6-1. Get the predicted class label and the class rank of each graph.
    6-2. Sort the graphs according to the class rank.
    6-3. Set the majority class label of the top S graphs as the label of the unlabeled sample.
```

Algorithm 2. Calculate the value of six scores of graph evaluation

```
Input: The set of constructed graphs based on different features and parameters Wset, the label of labeled data L_L, the results of backward label propagation $f_i$
Output: Six evaluation parameters namely, $Acc\_total^i$, $SWICW_c^i$, $MPOB_c^i$, $EPOB_c^i$, $SPOB_c^i$, $Acc\_class_c^i$.
---------------------------------------------------------------
1. For each graph $i$
2. Calculate the accuracy of backward label propagation and set it as $Acc\_total^i$
3. For each class c=1,…, C do the following:
    3-1. Set $SWICW_c^i$ using Equation (3).
    3-2. Set $MPOB_c^i$ using Equation (4).
    3-3. Set $EPOB_c^i$ using Equation (5).
    3-4. Set $SPOB_c^i$ using Equation (6).
    3-5. Set $Acc\_class_c^i$ as the accuracy of the $c^{th}$ class.
```

3. EXPERIMENTAL RESULTS

In this section, we evaluate and analyze the results of the simulations. In Section 3-1, we explain the employed datasets. In Section 3-2, we present the extracted features and the graph construction process. In Sections 3-3, 3-4, and 3-5, we evaluate the parameters and their correspondence with the accuracy of label propagation. Finally, in Section 3-6, we bring the result of the proposed ensemble method. All of the simulations have been carried out in the MATLAB environment.

3. 1. Face Datasets

We employed four gray image datasets which are described as below:

1. Extended Yale (Yale Ext): contains 16128 gray images from 28 persons in 9 different angles and 64 different lighting situations. We selected a subset of data with 1774 images.
2. PF01: contains the gray image of 103 persons, 53 men, and 50 women, in 17 different situations (1 normal face, 4 different lightings, 8 different angles, and 4 different face modes.
3. PIE: We use a reduced dataset containing 1926 face images of 68 individuals which contain pose, illumination, and facial expression variations.
4. Feret: we select a subset containing 1400 images of 200 persons (7 images for each individual) [26].

Figure 1 shows some samples of these four datasets.

3. 2. Feature Extraction and Graph Construction

For each image in the dataset, we use seven descriptors. The first one is the pure grayscale image. Moreover, we extract three variations of Local Binary Pattern (LBP) [27] descriptor all with the radius of one and neighborhood size of eight. The concatenated LBP image, with the dimensionality of 900, the normalized histogram of the uniform LBP with the dimensionality of 59 and a block-based LBP with the dimensionality of 944. For block-based LBP, the image is divided into 9 cells and from each cell, the histogram of uniform LBP is calculated. The nine histograms are then concatenated to form a 944-dimensional feature vector. Furthermore, we extract Gabor feature with 5 scales and 8 orientations that lead to a 2560 feature vector and finally two variations of a 9 channel Covariance feature [28] (whole image and block - based) with the dimensionality of 45 and 405 as image descriptors. For the block-based Covariance matrix, the image is divided into nine blocks and the extracted features from the blocks are then concatenated to form the 405-dimensional feature vector.

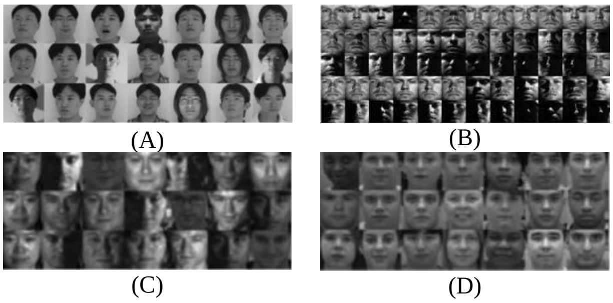

(A) (B)
(C) (D)

Figure 1. samples of image datasets. (A) PF01, (B) Yale Ext, (C) Pie, (D) Feret

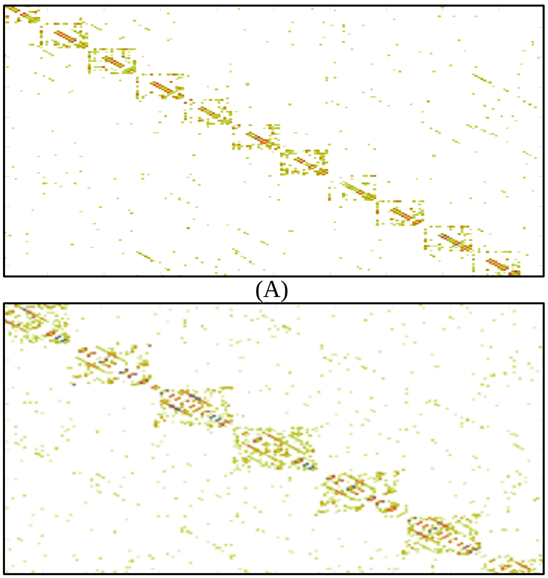
For the graph weight, we use L1_Roboust [29], which is a data self-representative method, meaning that one image is represented by the linear combination of the database and the contribution of each image is considered as its similarity and used as the edge weight. For each image descriptor, we used seven different threshold values (i.e., 0.05, 0.1, 0.5, 0.9, 1, 1.5, and 2) of the L1_Robust method to represent each sample where the coefficients obtained from each threshold is used to construct one graph. Hence, in total for each database, we constructed 49 graphs to use them in the selection and combination phases.

In the conducted experiments, to obtain results less dependent on the selected set of labeled\unlabeled samples, we repeat each experiment for 10 different combinations of labeled/unlabeled set and report the average of the accuracy.

3. 3. SWICW Parameter

In this section, we evaluate the results of graph ranking based on the proposed SWICW criteria. For a visual comparison, we put data from the same class alongside each other. For a good graph, we expect to observe large weights for the samples having the same class label and weak weights between the data of different classes. This causes a block shape graph wherein each block we have the samples correspond to the same class. Figure 2 shows a part of the graphs that have the highest SWICW, where Figure 3 shows the graphs with the lowest SWICW. As we observe in Figure 2, the graphs have a block shape which means that there exist edges between the samples from the same class, while in Figure 3 we observe that there is no clear edge between the nodes of the same class. This results visually confirms the goodness of these criteria.

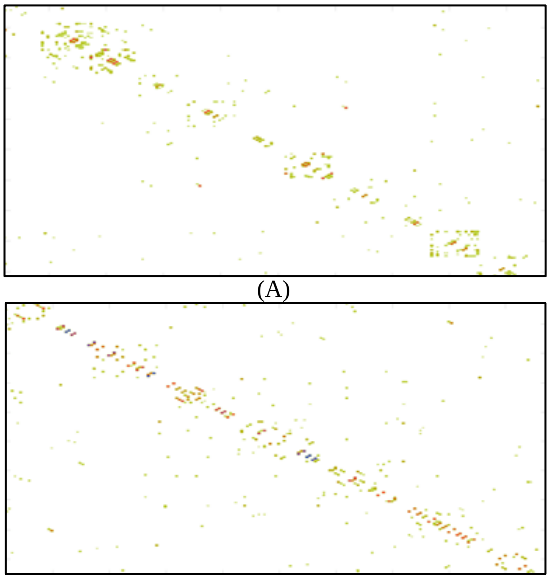
For a quantitative evaluation of the proposed criteria, for each of the 49 graphs built based on different features and parameters, the matching of SWICW values to the accuracy of label propagation methods LGC and GFHF are calculated and evaluated for one combination of label data (30%) and unlabeled data (70%) and also the average of 10 combinations. First, based on a specific combination of data, the SWICW values and the accuracy of label propagation methods LGC and GFHF have been compared (SWICW split). As we know, different combinations of labeled/unlabeled could result in different outcomes. Therefore, for evaluating how well the labeled data could be generalized for the entire data, the average accuracy of label propagation methods LGC and GFHF for 10 different combinations of labeled and unlabeled



(A)

(B)

Figure 2. Part of the graphs that have the highest SWICW values in (A) PF01, (B) Pie databases.

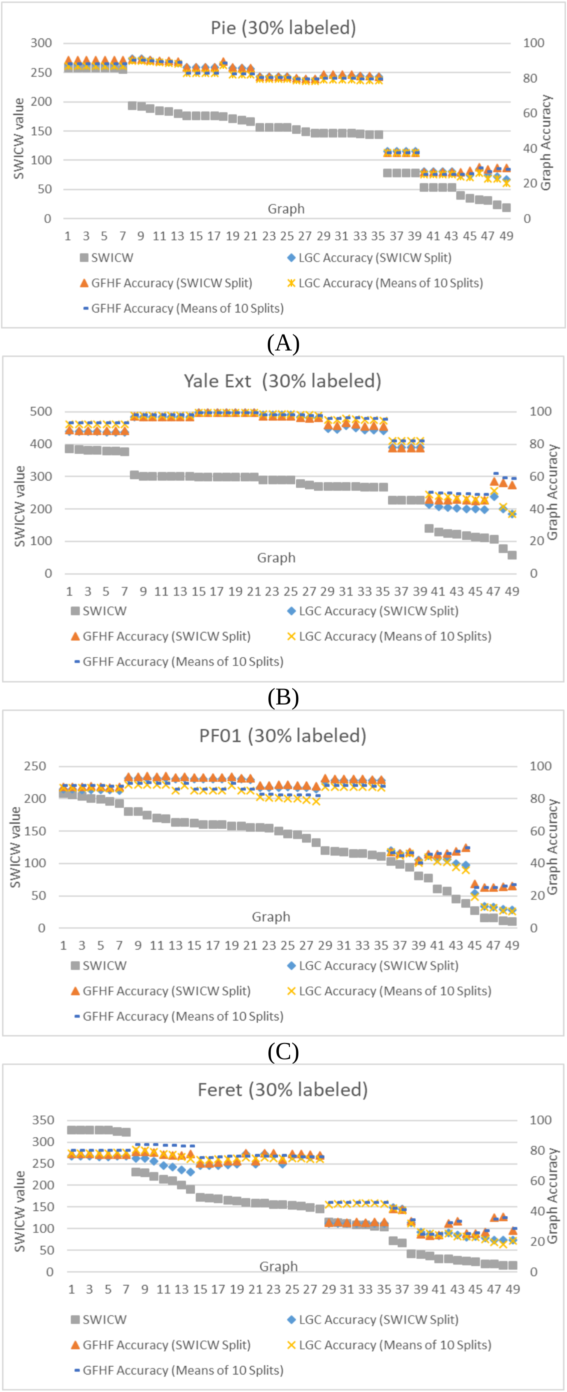


(A)

(B)

Figure 3. Part of the graphs which have the lowest SWICW value in (A) PF01 and (B) Pie datasets.

data for all 49 graphs were compared to the values of SWICW (means of 10 splits). Then, the results were sorted decadently based on the SWICW parameter and reported. We expect to observe low label propagation accuracy for the graphs with low values of SWICW and high accuracy for the graphs with high values of SWICW. We plot the SWICW parameter and the accuracy for PIE, Ext Yale, PF01 and FERET datasets in Figure 4. As the results show in Figure 4, although in some cases the proposed criteria are not completely matched to the label propagation accuracy, in most cases, the graphs with the high label propagation accuracies are located at the beginning of the list while the graphs at the end of the list have low accuracies.



(A)

(B)

(C)

(D)

Figure 4. Comparison of SWICW and label propagation accuracy of LGC and GFHF (for one and average of ten combinations) on (A) Pie, (B) Yale Ext, (C) PF01, and (D) Feret databases

3. 4. Comparison between the Accuracy of Label Propagation and Backward Label Propagation

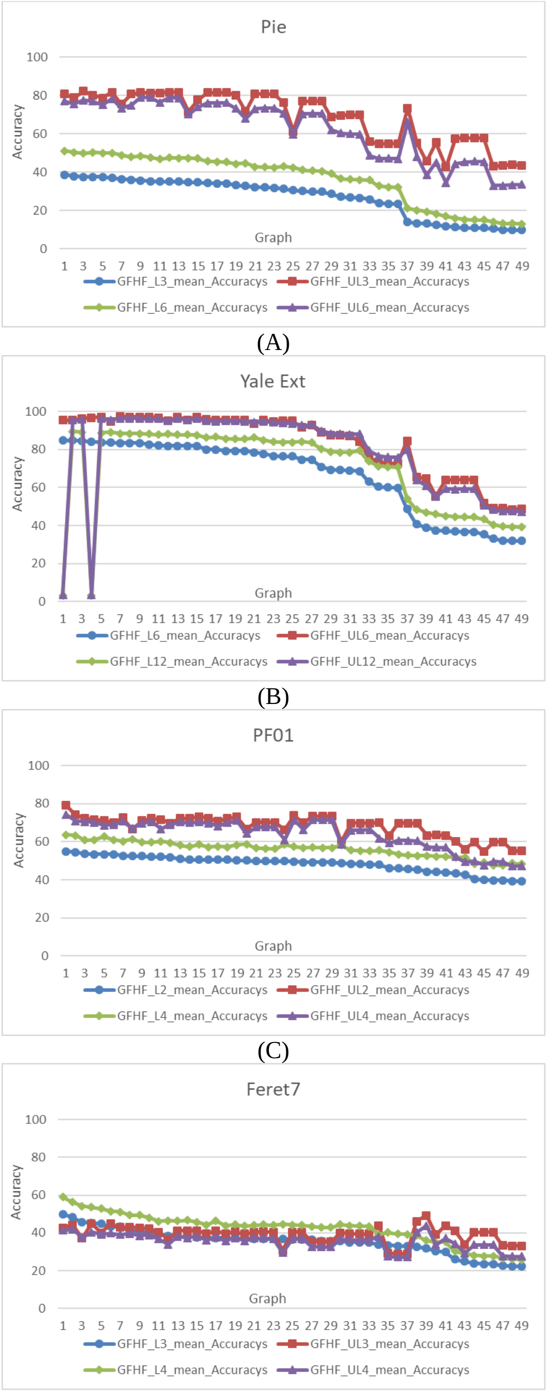
As we stated before, first, we perform the label propagation for all of the graphs and then the label is propagated backward. In this phase, the estimated labels of unlabeled data are propagated to the labeled data (which in here are considered as unlabeled data). Since we have the true labels, the accuracy of label propagation for each graph and each class is calculated. The goal of this experiment is to observe how well the backward label propagation accuracy can mimic the forward label propagation accuracy.

In Figure 5, the average accuracy of label propagation (on the unlabeled samples) and backward label propagation (on the labeled samples) for different combinations of labeled and unlabeled data are calculated for four datasets. "GFHF L3 mean accuracies" and "GFHF L4 mean accuracies" denote the accuracy of label propagation for 3 and 4 labeled data and "GFHF UL3 mean accuracies" and "GFHF UL4 mean accuracies" indicate the backward label propagation accuracies, respectively.

It should be noted that for instance, if we have 9 samples per class and 3 are used as labeled and 6 as unlabeled, in the backward label propagation we will have 6 labeled and 3 unlabeled samples. In Figure 5, the horizontal axis is label propagation accuracy and the vertical axis is the graph number (49 graphs). As we observe, the accuracy of backward label propagation shows similar behavior compared to the label propagation on the unlabeled samples. In other words, when the backward LP accuracy is high, the LP on unlabeled samples is high too and when backward LP accuracy is low, the LP on unlabeled samples is low too. Hence, by comparing the backward LP accuracies on the labeled samples, one can estimate which graph will have a higher accuracy on the unlabeled samples.

3. 5. Comparison of the Accuracy of the Label Propagation and the Features Extracted from the Label Vector

As we explained in Section 2, the mixture of maximum, entropy and standard deviation of the label vector can yield proper estimation for the performance of a graph. In Figure 6, we plot the label propagation accuracy and the average of the three above-mentioned parameters. For the four face databases as we can see, when the feature parameter from the label vector is high, the accuracy is high too while the low value of the proposed feature corresponds to the graphs with low accuracies. It indicates that this feature, although adopts only the label of the labeled samples, can well predict the accuracy of each graph on the unlabeled samples.



(A)

(B)

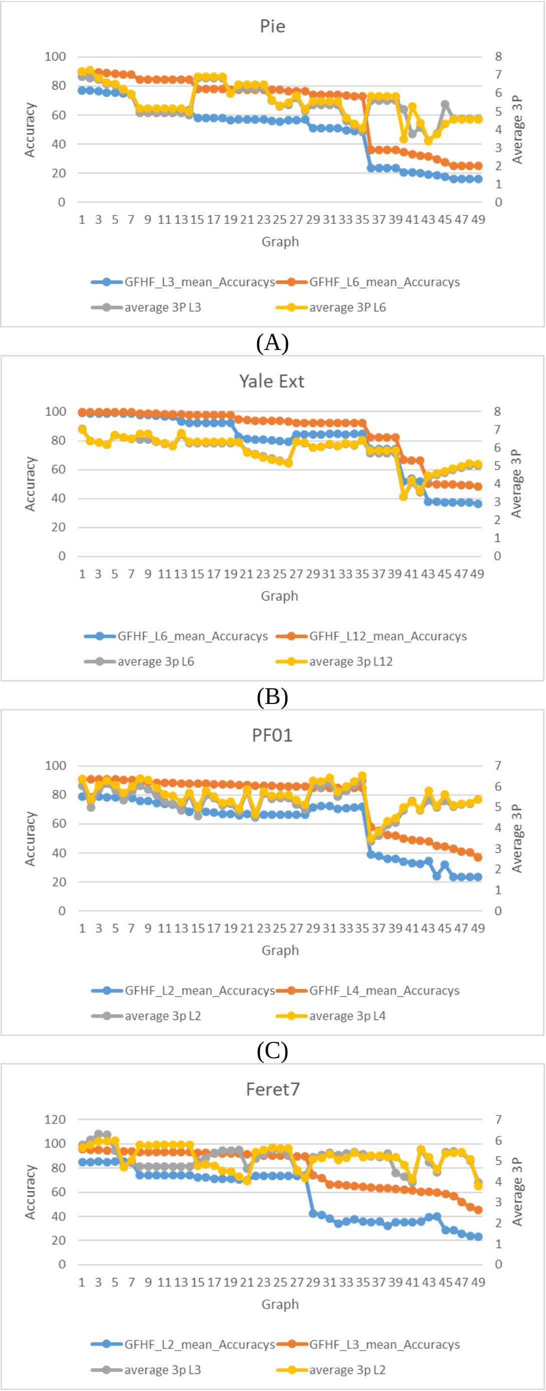
(C)

(D)

Figure 5. The accuracy of label propagation and backward label propagation for GFGF (average of ten different combinations). (A) Pie, (B) Yale Ext, (C) PF01, (D) Feret.

3. 6. Results of Decision Fusion

In this section, we numerically evaluate the proposed sample dependent



(A)

(B)

(C)

(D)

Figure 6. Comparison of the average of the three proposed parameters (maximum, entropy and standard deviation of class membership probability) with the average accuracy of the label propagation for (A) Pie, (B) Yale Ext, (C) PF01, (D) Feret

fusion method. Table 1 compares the results of the proposed method with SMGI, the graphs with the highest accuracy (denoted by "Best"), and the lowest accuracy (denoted by "Worst") for the four face datasets. For each database, we considered five different combinations of labeled and unlabeled data (10%, 20%, 30%, 50%, 60% as labeled and rest as unlabeled). For each number of labeled samples, we repeat the experiments for ten different combinations of labeled/unlabeled and calculate the accuracy. The average accuracy for Pie, Yale Ext, PF01, and Feret database are plotted in Figures 7-10, respectively where the horizontal axis denotes accuracy of label propagation method and the vertical axis (1 to 5) indicates 10%, 20%, 30%, 50%, 60% labeled samples.

Furthermore, we report the average and standard deviation of ten combinations of labeled/unlabeled samples in Table 1. As we observe, for Feret, PF01, and Pie database the proposed method outperforms the single graph with the highest accuracy. Furthermore, in the Yale Ext dataset, the proposed method outperforms the best graph when 50% and 60% of samples are used as labeled data. Moreover, for all databases and the number of training samples, the proposed method has higher accuracy compared to the SMGI method.

In Table 2, we report the average CPU time, over 10 different runs, for the proposed method and that of the SMGI algorithm on Extended Yale and Feret datasets. As we can see, despite the better accuracy of the proposed method, it has a higher CPU time compared to the SMGI. This is mainly because, in the proposed method, we have two processes of label propagations, forward and backward label propagation. An interesting observation is that in the proposed method, the CPU time decreases till 50% of labeled samples and after

TABLE 1. Comparison between the proposed method and SMGI and the Best and Worst graph

Dataset	Labeled percent	10%	20%	30%	50%	60%
Feret	# Labeled samples	1	2	3	4	5
	Best	60.11±10.6	74.9±10.5	84.31±8.3	90.17±13.8	92.6±12.5
	Worst	14.88±3.5	19.84±4	25.08±5.3	32.02±7.4	35.68±8.8
	Proposed method	**67.02±12.8**	**87.97±9.1**	**97.68±2.7**	**99.87±0.3**	**100±0**
	SMGI	42.15±15.7	44.1±16.7	66.38±22.3	61.02±15.6	79.03±19.1
pf01	# Labeled samples	2	4	6	9	11
	Best	75.74±5.9	85.41±3.1	89.76±4.3	93.60±2.7	95±2.3
	Worst	21.15±2.6	23.39±2.5	25.23±1.1	27.94±2.3	29±4.5
	Proposed method	**88.56±4.9**	**96.99±1.3**	**99.08±0.5**	**99.84±0.2**	**100±0**
	SMGI	46.87±18.1	65.2±17.6	80.97±6.3	87.24±6.7	88.89±4.3
pie	# Labeled samples	3	6	9	14	17
	Best	74.94±5.9	85.92±3.5	90.3±1.6	94.05±2.9	94.3±2.4
	Worst	15.31±2	21.33±2.6	25.05±3.2	30.62±1.8	31.99±3.3
	Proposed method	**75.66±4.6**	**90.74±2.4**	**96.22±0.6**	**98.88±1.3**	**99.66±1.4**
	SMGI	33.48±21.7	45.69±22.7	51.77±24.6	91.69±3.7	92.35±2.8
yaleExt	# Labeled samples	6	12	18	30	36
	Best	99.21±0.3	99.53±0.1	99.58±0.2	99.71±0.3	99.74±0.3
	Worst	35.12±1.9	43.31±3.5	49.12±2.8	54.72±2.0	56.88±3.2
	Proposed method	96.92±1.4	99.01±0.3	99.56±0.3	**99.86±0.1**	**99.88±0.1**
	SMGI	89.48±11.7	97.07±0.9	98.16±0.8	98.83±0.4	99±0.2

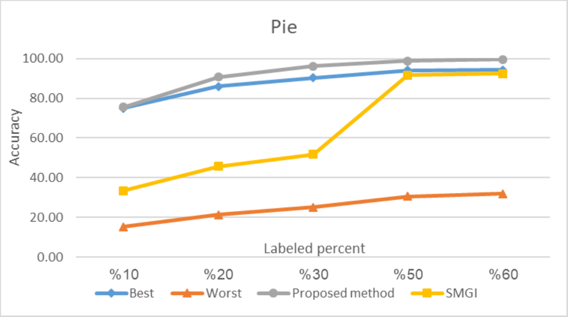


Figure 7. Comparison of the proposed method's accuracy respect to SMGI, the best and worst graph for the Pie dataset

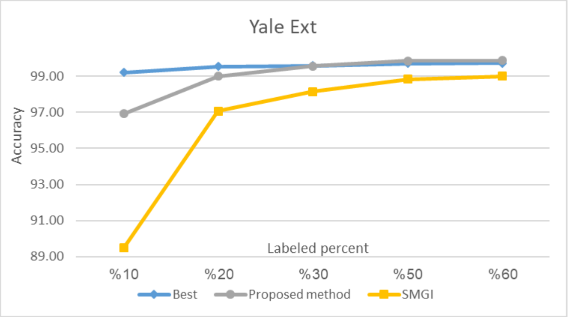


Figure 8. Comparison of the proposed method's accuracy respect to SMGI, the best and worst graph for Yale Ext dataset

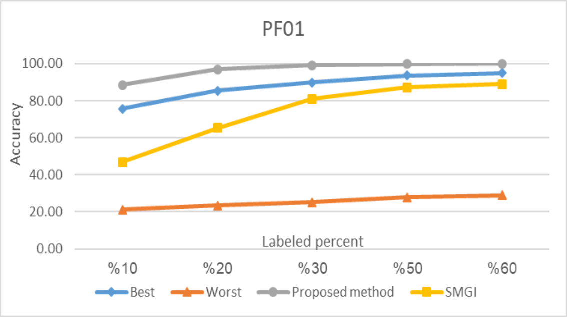


Figure 9. Comparison of the proposed method's accuracy respect to SMGI, the best, and worst graph for the PF01 dataset

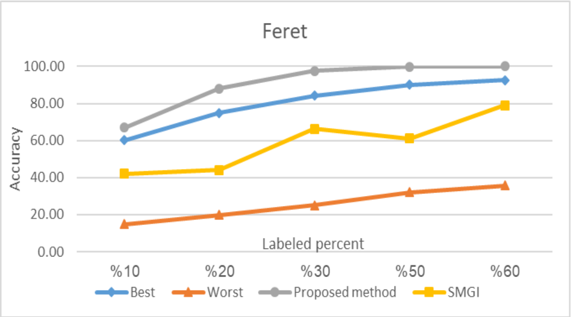


Figure 10. Comparison of the proposed method's accuracy respect to SMGI, the best, and worst graph for the Feret dataset

that, the CPU time of the proposed method increases again. This is because by increasing the number of labeled samples, even though the unlabeled samples decreases, in the backward label propagation, we propagate the labels to the labeled samples. Hence, after 50% of the labeled samples, we again observe an increase in the total CPU time of the proposed method.

TABLE 2. Elapsed CPU time of the proposed method versus SMGI

Dataset	Labeled percent	Run Time (seconds)	
		Proposed method	SMGI
Yale Ext	10%	9.27	3.31
	20%	7.91	2.55
	30%	7.13	2.44
	50%	6.17	2.43
	60%	6.27	2.39
Feret	10%	6.25	4.60
	20%	5.72	3.72
	30%	5.41	3.56
	50%	5.39	3.36
	60%	5.68	3.19

4. CONCLUSION

In this article, we proposed a ranking method that predicts how well a graph will perform in a classification task. By the fusion of six different indicators, we assign a score to each graph which enables us to sort and select the graphs with the best performance in the consecutive learning task. In addition to that, the proposed parameter indicates how well a graph can predict the samples in a specific class. Moreover, we proposed a sample dependent fusion technique that fuses the decisions obtained from each graph. Experimental results on four face databases with different combinations of the labeled\unlabeled samples show that the proposed ranking method can well sort the graphs according to their performance in the consecutive label propagation task and the proposed fusion method can enhance the accuracy compared to the single graph use and SMGI algorithm.

For future works, we intend to evaluate the effect of eliminating connections of samples in different classes on the final accuracy. Moreover, in the case that we have a few labeled data, the proposed method could not reach a suitable performance. Therefore, we intend to propose a fast graph selection method that is not dependent on the size of the labeled data.

5. REFERENCES

1. Wang, B., Mezlini, A.M., Demir, F., Fiume, M., Tu, Z., Brudno, M., Haibe-Kains, B. and Goldenberg, A., "Similarity network fusion for aggregating data types on a genomic scale", ***Nature Methods***, Vol. 11, No. 3, (2014), 333–337.
2. Zhu, X., "Semi-Supervised Learning: From Gaussian Fields to Gaussian Processes", (2003), 1-21. http://ra.adm.cs.cmu.edu/anon/2003/CMU-CS-03-175.pdf
3. Szummer, M. and Jaakkola, T., "Partially labeled classification with Markov random walks", In Advances in Neural Information Processing Systems 14: Proceedings of the 2001 Conference (Vol. 2), MIT Press, (2002), 945–952.
4. Joachims, T., "Transductive learning via spectral graph partitioning", In Proceedings of the 20th International Conference on Machine Learning (ICML-03), (2003), 290–297.
5. Sindhwani, V., Niyogi, P., and Belkin, M., "Beyond the point cloud: from transductive to semi-supervised learning", In Proceedings of the 22nd international conference on Machine learning, (2005), 824–831.
6. Belkin, M., Niyogi, P., and Sindhwani, V., "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples", ***Journal of machine learning research***, Vol. 7, No. Nov, (2006), 2399–2434.
7. Blum, A. and Chawla, S., "Learning from labeled and unlabeled data using graph mincuts" In Proceedings of the Eighteenth International Conference on Machine Learning, (2001), 19-26.
8. Bengio, Y., Delalleau, O., and Le Roux, N., "label propagation and quadratic criterion", MIT Press, (2006), 35-112.
9. Herbster, M., Pontil, M., and Wainer, L., "Online learning over graphs", In Proceedings of the 22nd international conference on Machine learning, (2005), 305–312.
10. Thrun, S., Saul, L.K., and Schölkopf, B., Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference, Vol. 16, MIT press (2004).
11. Wang, B., Tu, Z., and Tsotsos, J. K., "Dynamic label propagation for semi-supervised multi-class multi-label classification", In Proceedings of the IEEE international conference on computer vision, (2013), 425–432.
12. Gui, J., Hu, R., Zhao, Z. and Jia, W., "Semi-supervised learning with local and global consistency", ***International Journal of Computer Mathematics***, Vol. 91, No. 11, (2014), 2389–2402.
13. Lu, J., Hu, J., and Tan, Y.-P., "Discriminative deep metric learning for face and kinship verification", ***IEEE Transactions on Image Processing***, Vol. 26, No. 9, (2017), 4269–4282.
14. Mirmahboub, B., Mekhalfi, M. L., and Murino, V., "Person re-identification by order-induced metric fusion", ***Neurocomputing***, Vol. 275, (2018), 667–676.
15. Zhang, H., Huang, T.S., Nasrabadi, N.M. and Zhang, Y., "Heterogeneous multi-metric learning for multi-sensor fusion", In 14th International Conference on Information Fusion, (2011), 1–8.
16. Zhang, L. and Zhang, D., "Metricfusion: Generalized metric swarm learning for similarity measure", ***Information Fusion***, Vol. 30, (2016), 80–90.
17. Karasuyama, M. and Mamitsuka, H., "Multiple graph label propagation by sparse integration", ***IEEE Transactions on Neural Networks and Learning Systems***, Vol. 24, No. 12, (2013), 1999–2012.
18. Saeedeh, B. and Bosaghzadeh, A., "Deep graph fusion for graph based label propagation", In 2017 10th Iranian Conference on Machine Vision and Image Processing (MVIP), (2017), 149–153.
19. An, L., Chen, X., and Yang, S., "Multi-graph feature level fusion for person re-identification", ***Neurocomputing***, Vol. 259, (2017), 39–45.
20. Tong, T., Gray, K., Gao, Q., Chen, L., Rueckert, D. and Alzheimer's Disease Neuroimaging Initiative, "Multi-modal classification of Alzheimer's disease using nonlinear graph fusion", ***Pattern Recognition***, Vol. 63, (2017), 171–181.
21. Lin, G., Liao, K., Sun, B., Chen, Y. and Zhao, F., "Dynamic graph fusion label propagation for semi-supervised multi-modality classification", ***Pattern Recognition***, Vol. 68, (2017), 14–23.
22. Zhang, H., Li, S., Shi, Y. and Yang, J., "Graph Fusion for Finger Multimodal Biometrics", ***IEEE Access***, Vol. 7, (2019), 28607–28615.
23. Cheng, Y., Li, Y., Liu, Q., Yao, Y., Sai Vijay Kumar Pedapudi, V., Fan, X., Su, C. and Shen, S., "A graph based unsupervised feature aggregation for face recognition", In Proceedings of the IEEE International Conference on Computer Vision Workshops, (2019), 1-10.
24. Zhu, X., Ghahramani, Z., and Lafferty, J. D., "Semi-supervised learning using gaussian fields and harmonic functions", In Proceedings of the 20th International conference on Machine learning (ICML-03), (2003), 912–919.
25. Zhou, D., Bousquet, O., Lal, T.N., Weston, J. and Schölkopf, B., "Learning with local and global consistency", In Advances in Neural Information Processing Systems, (2004), 321–328.
26. Dornaika, F., Kejani, M. T., and Bosaghzadeh, A., "Graph construction using adaptive Local Hybrid Coding scheme", ***Neural Networks***, Vol. 95, (2017), 91–101.
27. Ojala, T., Pietikäinen, M., and Mäenpää, T., "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns", ***IEEE Transactions on Pattern Analysis & Machine Intelligence***, Vol. 24, No. 7, (2002), 971–987.
28. Tuzel, O., Porikli, F., and Meer, P., "Region covariance: A fast descriptor for detection and classification", In European conference on computer vision, Springer, Berlin, Heidelberg, (2006), 589–600.
29. Fujiwara, Y., Ida, Y., Arai, J., Nishimura, M. and Iwamura, S., "Fast algorithm for the lasso based l_1-graph construction", ***Proceedings of the VLDB Endowment***, Vol. 10, No. 3, (2016), 229–240.

Persian Abstract

چکیده

در بسیاری از موارد ارزیابی یک رویداد بر اساس یک ویژگی تک ممکن است به نتایج دقیق و مناسب منجر نشود. به علاوه، حتی اگر چند ویژگی داشته باشیم، نمی¬توان از پیش تعیین کرد که کدام ویژگی توصیف بهتری از رخداد مورد نظر را بیان می¬کند. گراف شباهت ابزاری است که می¬تواند ارتباط بین نمونه¬ها را نمایش دهد. در این مقاله، یک روش مرتب¬سازی برمبنای گراف و نمونه معرفی می¬کنیم که با استفاده از شش پارامتر پیشنهاد شده گراف¬ها را مرتب می¬کند. مرتب کردن گراف¬ها طوری انجام می¬پذیرد که گراف¬هایی که در بالای لیست قرار دارند عملکرد بهتری نسبت به گراف¬های انتهای لیست دارند. علاوه بر این، یک روش برای بهینه کردن پارامترها و ترکیب گراف¬ها در سطح تصمیم¬گیری ارایه شده است. نتایج پیاده¬سازی نشان می¬دهد که روش پیشنهادی علاوه بر اینکه گراف¬ها را به درستی بر مبنای دقت آنها مرتب می¬کند، در مرحله ترکیب، دقت را نسبت به استفاده از تک گراف افزایش می¬دهد.