Community Detection to Invariant Pattern Clustering in Images

Lusmar M. Freitas and Murillo G. Carneiro
Faculty of Computing
Federal University of Uberlândia
Uberlândia, Brazil
lusmarmf@gmail.com, mgcarneiro@ufu.br

***Abstract*—Community detection is a kind of clustering task which aims to find groups of vertices densely connected internally but sparsely connected to other groups. In comparison with conventional clustering methods, community detection methods are able to examine structural, functional and dynamical properties of the networked data, beyond its physical attributes. However, such techniques have been barely explored in the literature as most of the machine learning data sets are represented as non-graph data (e.g., feature vectors, images, texts, and so on). In this work, we propose a simple community detection framework based on the literature that covers since the graph construction process from feature vectors generated from non-graph data until the application and evaluation of community detection methods over such a graph. The framework is further evaluated on the problem of invariant pattern clustering of images, which consists of given a set of image objects taken from different angles, positions or rotations, clustering the images related to each object. Experiments were conducted considering three community detection methods (fast greedy, walk-trap and label propagation) and two relevant clustering methods (k-means and HDBSCAN). The results indicate FG as the better choice among those algorithms as it usually approximates efficiently the ground-truth groups while also keeping a reasonable number of communities. Moreover, our results suggest that community detection may be an efficient task not only to cluster graph data, but also domain applications represented by non-graph data.**

***Index Terms*—complex networks, community detection, clustering, invariant pattern, graph-based learning.**

I. Introduction

Clustering is a common machine learning task which aims to find groups of data items characterized by some (dis)similarity criterion. Such a task contribute to the process of knowledge discovery in data through of several exploratory analysis related to decision making, data mining, information retrieval and pattern analysis, just to name a few [1]. Formally, the clustering task can be defined as follows: given a data set $\mathbf{X} = \{x_1, \ldots, x_n\}$, with $x_i = \{f_1, \ldots, f_d\}$ denoting a d-dimensional vector, where each of the entries is called a feature, the objective is to find subgroups of data items, i.e., $\mathcal{C} = \{c_1, c_2, \ldots, c_k\}$, such that the members of the same subgroup are more similar than members of different groups.

Community detection is a particular case of clustering which is related to networked data. Communities represent the organization of vertices in clusters, which can be identified by the existence of many edges connecting intra-clusters vertices and few edges connecting inter-clusters vertices. They may represent (or uncover) vertices that probably play functions or have similar behaviors, as they examine not only the physical attributes of the data (e.g., distance or similarity), but also the topological ones. Formally, consider a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is the set of vertices and $\mathcal{E} = \{e_1, \ldots, e_L\} \subset \mathcal{V} \times \mathcal{V}$ denotes the set of edges, the objective is to identify communities ($\mathcal{C}$), i.e., subgroups of vertices densely intra-connected but sparsely connected to other groups.

Although community detection methods have presented successful results in a wide range of problems [2], the application of such methods is also very limited to graph data, which means that the topic has been barely explored for other very common kinds of data representation in machine learning, such as feature vectors, images and texts. To the best of our knowledge, the most related work to ours are presented in [3]–[5], in which community detection algorithms were investigated on data sets represented as feature vectors. Regarding other data representation, [6] presented a framework for image segmentation, which is based in super-pixels and employs the fast greedy algorithm [7] for community detection.

In this paper, we make a contribution in that sense, by modeling a simple framework to apply and evaluate community detection methods for non-graph data. To be specific, we analyze the proposed framework on the real-world problem of invariant pattern clustering from images. Such a problem consists of given a set of image objects captured from different angles, positions or rotations, have algorithms capable of clustering the images related to each object.

The remainder of this work is organized as follows: Sect. II describes the proposed model; Sect. III presents the experimental results, respectively; and Sect. IV concludes the paper.

II. Model Description

In the following, we describe in detail the framework proposed in this paper. In a few words, it consists of the following steps:

1) Obtain the data set $\mathbf{X}$ after preparing and pre-processing the raw data;
2) Build up an undirected graph $\mathcal{G}$ from the data set $\mathbf{X}$;
3) Apply a community detection method over $\mathcal{G}$ to obtain the set of communities $\mathcal{C}$;
4) Evaluation of $\mathcal{C}$ in terms of well-known metrics.

A. Data Preparation and Pre-Processing

As a data preparation we extract attributes from the non-graph data in order to build up the graph in the next step. After the raw data is converted to feature vectors, we apply some domain-driven pre-processing methods in order to obtain our data set $\mathbf{X}$. In case of the data is already available as feature vectors, we focus on the pre-processing (if necessary).

B. Graph Construction

In this step, a graph construction method $g(.)$ is applied over the feature vectors data set, $\mathbf{X}$, to obtain an undirected graph $\mathcal{G}$, i.e., $g(\mathbf{X}) \rightarrow \mathcal{G}$. The k-nearest neighbors graph (kNN-graph) [2], [8] has been defined as $g(.)$ in this paper. Let $\mathbf{D}$ be a distance (or similarity) matrix, δ be a vector-based distance function (e.g., Euclidean distance) such that $\mathbf{D}_{ij} = \delta(x_i, x_j)$, and Λ_i denote the set of k-nearest neighbors of x_i in terms of $\mathbf{D}$, the adjacency matrix $\mathbf{A}$ of a kNN-graph is given by:

$$\mathbf{A}_{ij} = \begin{cases} 1, & \text{if } x_j \in \Lambda_i \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

As $\mathbf{A}$ may no be symmetric, we make the graph undirected by applying the following post-processing: $\mathbf{A} = max(\mathbf{A}, \mathbf{A}^T)$.

C. Community Detection Algorithms

Here we present a brief description about the three community detection algorithms investigated in our framework.

1) Fast greedy: FG defines communities based on the greedy optimization of modularity [7], [9]. Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, the algorithm starts by associating each vertex $v_i \in \mathcal{V}$ to an isolated community $c_i \in \mathcal{C}$. At each step, FG merges the two communities, i.e., $c_u = \{c_i \cup c_j\}$, which the union maximize the modularity Q. The algorithm repeat this step until all vertices belongs to a unique community. A dendrogram is generated summarying all merging steps and a cut is made in the division which provides the highest value of Q. The time complexity of FG lies on $O(\mathcal{E} + \mathcal{V}\log^2 \mathcal{V})$.

2) Walk-trap: WT is based on random walk theory and its basic idea is that objects that meet more frequently (adjacent) in a short random walk probably belong to the same community [10]. At the beginning, each vertex belongs to an isolated community. In the following, the distances between all adjacent vertices are calculated by $r_{i,j}(t) = \sqrt{\sum_{q=1}^{n} \frac{(P_{iq}^t - P_{jq}^t)^2}{d(q)}}$, where P_{iq}^t is the probability of move from v_i to v_q through of a random walk of size t and $d(q)$ is the degree of the vertex q. WT then basically iterates the following steps: i) choose two communities c_i and c_j in P_q based on distance criterion between communities, i.e., $r_{c_i,c_j}(t) = \sqrt{\sum_{q=1}^{n} \frac{(P_{c_i q}^t - P_{c_j q}^t)^2}{d(q)}}$; ii) join these communities in a new community, i.e., $c_u = c_i \cup c_j$, and create the new partition $P_{q+1} = (P_q\{c_i, c_j\}) \cup \{c_u\}$; and iii) update the distance between communities until all vertices are in the same partition. The algorithm has time complexity of $O(\mathcal{V}^2 \log \mathcal{V})$.

3) Label propagation (LP): Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, LP starts by associating each vertex $v_i \in \mathcal{V}$ to an isolated community $c_i \in C$. At each step, the vertices are treated randomly and the labels are propagated on the graph according to the predominant label in their neighborhood [11]. Let $L = \{l_1, \ldots, l_p\}$ be the labels in the current step and $d_i^{l_j}$ the number of neighbors of v_i with label l_j, LP associates v_i to the label l_j if $d_i^{l_j} \geq d_i^{l_z}$, with z denoting every label associated to the set of L - l_j labels. Because tie cases are solved randomly, each execution of the algorithm can return a different result. The time complexity of LP lies on $O(\mathcal{V} + \mathcal{E})$.

D. Evaluation Metrics

In order to evaluate the generated communities, we consider both internal and external validation indexes. As we have the ground-truth groups of ETH-80 data, we have given more attention to the external ones in our analyses.

1) Internal index: Given a network partitioned in m communities, a matrix $E^{m \times m}$ which the elements, denoted by ϵ_{ij}, represent the fraction of the network edges that connect vertices between two communities i and j, the modularity is given by:

$$Q = \sum_i [\epsilon_{ii} - (\sum_j \epsilon_{ij})^2] = Tr(E) - \|E^2\| \ , \tag{2}$$

where $Tr(E)$ is the matrix trace and $\|E^2\|$ the sum of all elements.

2) External indexes: This category includes Normalized Mutual Information (NMI), Purity (Pu), Collocation (Co) and F_1. Let $\mathcal{C} = \{c_1, c_2, \ldots, c_{m1}\}$ denotes the communities (or clusters) obtained by a given community detection algorithm $f(.)$ and $Y = \{y_1, y_2, \ldots, y_{m2}\}$ the ground-truth groups, with $m1$ and $m2$ denoting respectively the number of communities obtained and the number of ground-truth groups. In the following, we formally define the indexes adopted here.

- Normalized Mutual Information (NMI) is a metric from information theory [12], which is defined by:

$$NMI(\mathcal{C}, Y) = \frac{I(\mathcal{C}, Y)}{[H(\mathcal{C}) + H(Y)]/2} \ , \tag{3}$$

  where I denotes the mutual information, which tells us how much we have learned about $\mathcal{C}$ if we know Y, and H the entropy.
- Purity (Pu) provides the percentage of the majority ground-truth group in each community. Given a ground-truth group $y_i \in Y$, a community $c_j \in \mathcal{C}$, and $m1$ (the number of generated communities), the purity can be defined as:

$$Pu(\mathcal{C}, Y) = \frac{1}{n} \sum_{j=1}^{m1} max_i(y_i \cap c_j) \ , \tag{4}$$

  where n denotes the number of instances.
- Collocation (Co), or Inverse Purity, calculates, for each ground-truth group, the percentage of the community with the largest number of instances for that group [13]. Given

a ground-truth group $y_i \in Y$, a community $c_j \in \mathcal{C}$, and $m2$ (the number of ground-truth groups), the collocation can be defined as:

$$Co(\mathcal{C}, Y) = \frac{1}{n} \sum_{i=1}^{m2} max_j (y_i \cap c_j) \ . \tag{5}$$

- F_1 calculates the harmonic mean between purity (Pu) and collocation (Co) and it is given by:

$$F_1 = \frac{2 \cdot Co \cdot Pu}{Co + Pu} \ . \tag{6}$$

Despite purity penalizes instances of different ground-truth groups in the same community, it does not reward objects of the same ground-truth group clustered in the same community, i.e., it can be maximum when each object belongs to an isolated community. By the contrary, collocation rewards objects of the same ground-truth group that are clustered together but not penalize when two or more ground-truth groups are mixed, i.e., it can be maximum when all objects are in the same cluster. Therefore, F_1 tries to maximize together both purity and collocation aiming at evaluate better those clusterings that provide pure communities but also with a reduced number of such communities, which certainly aid data analysis in the post-processing [14].

III. Computer Simulations

In this paper we consider the task of invariant pattern clustering and evaluate our study over the data sets of ETH-80 collection [15], in which each raw data item is represented by an image. To be specific, we aim at evaluating our community detection framework in function of three algorithms: fast greedy (FG), walk-trap (WT) and label propagation (LP), and considering a set of evaluation metrics presented before. We also compare the results of our framework against robust clustering methods, like HDBSCAN. For sake of clarity, we divide this section in five parts: data preparation, experimental setup, framework results, visualization analysis, and comparison against other clustering methods.

A. Data Preparation

The task of invariant pattern clustering in images consists of given a set of images objects taken at different angles, positions or rotations, have algorithms capable of clustering the images related to each object. To evaluate our community detection framework, we selected the ETH-80 data collection [15]. The raw data collection is represented by 3280 images divided into 8 categories: Apple, Pear, Tomato, Cow, Dog, Horse, Cup and Car. Each category contains 10 objects with great variation among them, even though they clearly belong to the same category. Each object has 41 images to represent it in different rotation, translation and scaling conditions. Figure 1 show examples of the 10 objects of each category.

As a data preparation step, we resize each image from 128x128 to 32x32 pixels (to improve computational performance) and then extract its attributes by calculating its RGB histograms over 8 bins each one. Thus every image is converted to a feature vector with 512 numeric attributes [16]. As all images lie on the same scale, we do not apply any pre-processing method here.

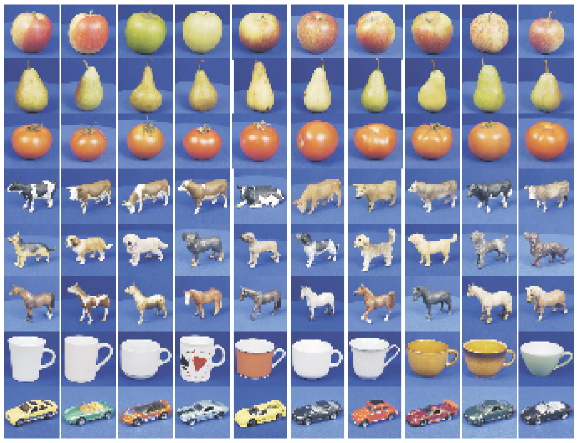


Fig. 1: Example of the categories of ETH-80 data base. Each category has 10 distinct objects.

B. Experimental Setup

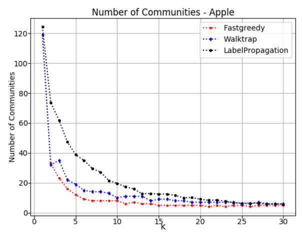
We evaluate our framework over eight data sets, each one denoting a ETH-80 category. After the transformation of the images to feature vectors, we generate the kNN-graph for such data sets. Thus, each image is mapped as a vertex into an underlying network. In such a step, we have fixed the Bhattacharyya distance as the vector-based distance metric and ranged the parameter k from $\{1, 2, \ldots, 30\}$ in order to provide distinct topological configurations of the networked data.

Regarding the community detection methods, they were executed with their default parameters. For LP, which is a stochastic method, we repeated the simulation 10 times and returned the averaged results. The analysis of the results follows the evaluation metrics defined before.

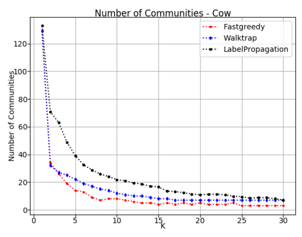
C. Community Detection Results

1) Number of Communities: Our first analysis refers to the number of communities obtained by each algorithm for the categories Apple and Cow, which is presented in Fig. 2. In the figure, the x axis denotes the variation of the parameter k in the graph construction step and the y axis the value obtained in terms of number of communities. As can be observed, the algorithms present relatively close values, especially when k increases. In general, the FG algorithm usually has fewer communities than the other algorithms, we believe that this is due to the fact that FG is based on a greedy approach to optimization of modularity, which joins the nodes until all are part of the same community. For sake of clarity, such an analysis repeats for the other six categories, but their graphics were omitted for reason of space.

2) Modularity: Here we analyze the community detection methods in terms of the modularity, as shown by Fig. 3. One can see that modularity decreases as k increases. In that case,



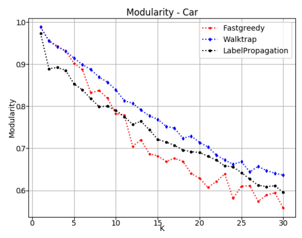
(a) Apple



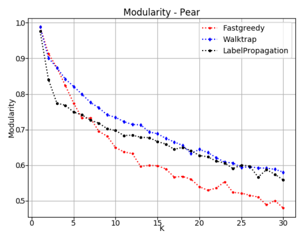
(b) Cow

Fig. 2: Number of communities (group) found by FG, WT and LP for some data sets: (a) Apple; and (b) Cow.

FG also tends to more distant from WT and LP. We also observe that sparse knn-graphs (smaller k values) present high modularity which is compliant with the modularity formulation in terms of random graphs. Overall, the literature says that modularity values greater than 0.5 indicates the existence of well-defined groups [7].



(a) Car



(b) Pear

Fig. 3: Modularity obtained by FG, WT and LP for some data sets: (a) Car; and (b) Pear.

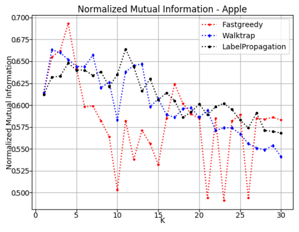
3) Normalized Mutual Information - NMI: Fig. 4 shows the NMI obtained by each community detection method. One can see that NMI is higher for smaller values of k. This is because NMI penalizes clusters that are not pure, once the number of clusters decreases as k increases and there are possibly more instances from distinct ground-truth groups in the same cluster. Such characteristic of NMI reflects its tendency in choosing clustering solutions with more communities [17], which is not usually desired. Indeed we are looking for clusterings that maximize purity and minimize the number of communities at the same time.

Fig. 4 also shows that FG can be sensitive to the variation of k in some data sets. As it usually provides less communities, k may have high influence in the purity of such clusters.

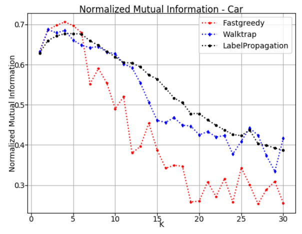
4) Purity Pu: The purity of the obtained communities is presented by Fig. 5. As expected, the purity values decrease as k increases, which means that purity is higher for isolated or small clusters. LP achieved the best performance here.

5) Collocation Co: Fig. 6 shows the collocation results obtained by FG, WT and LP. Unlike the purity metric, collocation values increase as k increases, which means that we are looking for a small number of clusters. FG achieved the best results here.

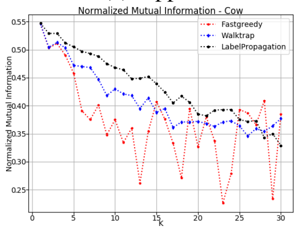
6) F_1: The results in terms of F_1 are presented in Fig. 7. One can see that FG and WT achieved the best results,



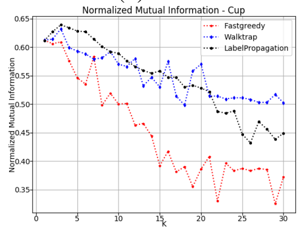
(a) Apple



(b) Car

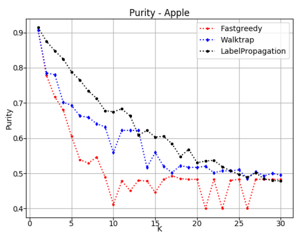


(c) Cow

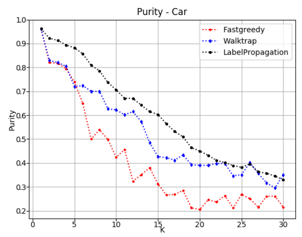


(d) Cup

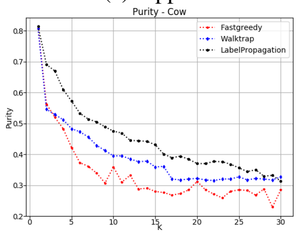
Fig. 4: Normalized Mutual Information (NMI) for the data bases (a) Apple, (b) Car, (c) Cow and (d) Cup.



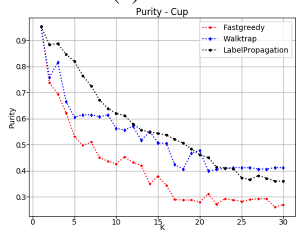
(a) Apple



(b) Cow



(c) Car



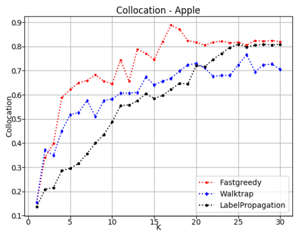
(d) Cup

Fig. 5: Purity for the data bases (a) Apple, (b) Car, (c) Cow and (d) Cup.

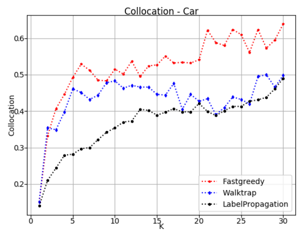
although LP seems to be more consistent regarding to the parameter variation. The harmonic mean between purity and collocation is coherent with we are looking for: high purity and small number of clusters. Therefore, it has been taken as the reference metric for the further analysis we present next.

D. Visualization Analysis

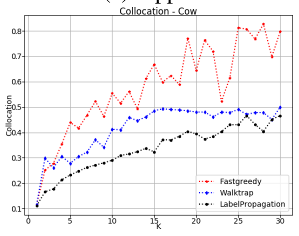
Now we move on to provide a qualitative analysis about the community detection algorithms investigated in our framework. Figure 8 presents the graph, communities and ground-truth groups for FG, WT and LP considering their best results on the Car data set. We have adopted the Kamada-Kawai layout [18] to generate the graphs. For LP, which is a stochastic method, we have considered the results obtained in its first execution. In the figure, communities are denoted by semi-



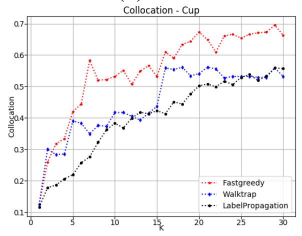
(a) Apple



(b) Car

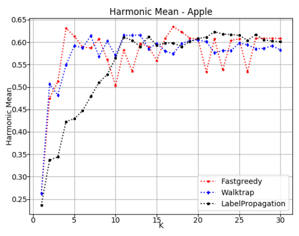


(c) Cow

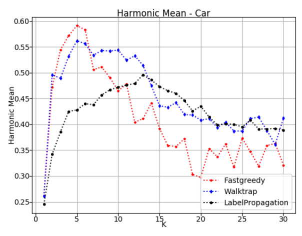


(d) Cup

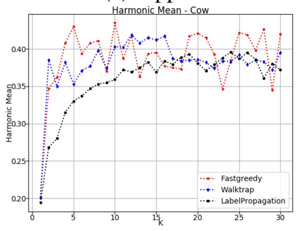
Fig. 6: Collocation for the data bases (a) Apple, (b) Car, (c) Cow and (d) Cup.



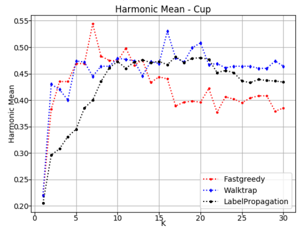
(a) Apple



(b) Car



(c) Cow



(d) Cup

Fig. 7: Harmonic Mean for the data bases (a) Apple, (b) Car, (c) Cow and (d) Cup.

transparent markers while ground-truth groups by individual colored vertices. Despite the number of communities is similar, the communities configurations are different. LP has the smallest number of communities, although they do not have the same purity as the communities found by FG and WT. By the contrary, the differences between FG and WT communities are little and lies on the number of communities. By analyzing the communities denoted by "A", "B" and "C" in the figure, one can see that FG is able to keep the purity similar to WT, but with less communities.

E. Comparison against conventional clustering methods

Now we move on to compare the community detection methods against other clustering algorithms. To be specific, we consider two traditional algorithms: the widely used k-means [19] and the robust HDBSCAN [20], [21]. In k-means, the parameter $\mathcal{K}$ denotes the number of clusters to be obtained by the algorithm; HDBSCAN has a parameter min_clust which defines the smallest size of a group in order to be considered as a cluster and the parameter $min\_samples$ which is responsible by taking a more or less conservative strategy in relation to noises and outliers. Regarding the parameter selection, k-means is optimized over the set $\mathcal{K} \in \{2, 4, \ldots, 100\}$; and HDBSCAN over the set $min\_clust \in \{2, 4, \ldots, 30\}$, with $min\_samples = 1$ defined empirically.

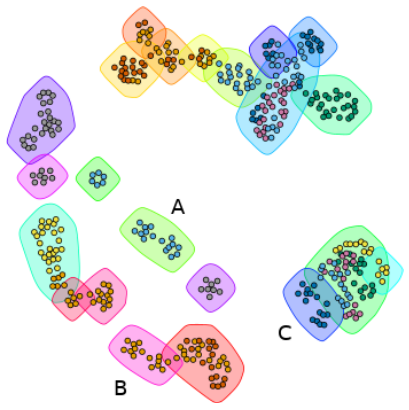
Table I shows the best F_1 value obtained by each one of the techniques under comparison. One can see that FG obtained the best results for seven data sets. The main advantage of FG seems to be able to found coherent groups while also keeping the number of clusters smaller than other techniques, which is a desired aspect to be considered in this study. Otherwise, k-means and HDBSCAN have troubles to cluster the invariant objects in their corresponding groups using a reasonable number of groups. We suspect that the community concept is less dependant of the physical features of the data (e.g., spatial distribution) as it also analyzes their topological features (e.g., structural distribution).

TABLE I: Comparison (in terms of F_1) among community detection and conventional methods over the clustering task of invariant object recognition in ETH-80 collection.

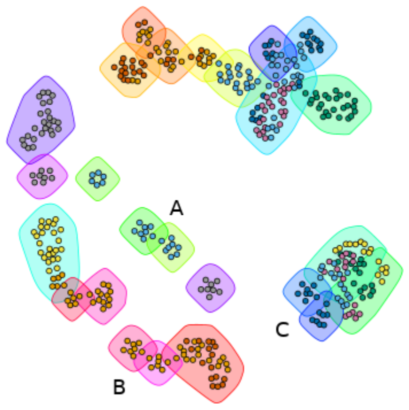
	Community Detection			Clustering Methods	
Category	FG	WT	LP	k-means	HDBSCAN
Apple	**63,4**	61,6	62,3	50,4	58,8
Car	**59,1**	56,2	49,6	38,0	36,3
Cow	**43,5**	41,9	39,6	29,0	30,7
Cup	**54,4**	53,0	48,2	47,5	49,1
Dog	46,5	**47,2**	42,8	40,3	39,2
Horse	**43,1**	42,9	40,9	38,3	39,2
Pear	**42,7**	41,0	38,5	32,8	32,5
Tomato	**43,5**	41,7	39,2	41,9	43,4

IV. Conclusions

In this work we model a simple framework to extend the application of community detection techniques to non-graph data. Although most of machine learning data is available as feature vectors, the usage of graphs can provide a different kind of data analysis which not only considers its physical attributes but also is able to take into account the topological ones. The salient features of such a representation have been enhanced by the development of complex networks tools able to uncover data insights through of the analysis of structural, functional and dynamics properties of the networked data. The framework consists of four steps: i) feature extraction from the raw data, ii) graph construction from the feature vectors previously generated, iii) detection of communities (or groups) in such a graph, and iv) the calculation of evaluation metrics based on literature. To be specific, three community detection algorithms have been adopted in the framework: fast greedy, walk-trap and label propagation .



(a) FG



(b) WT

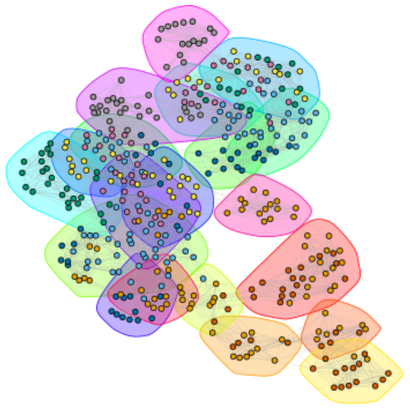

(c) LP

Fig. 8: Communities and ground-truth groups found by FG, WT and LP for the Car data set.

In order to analyze the performance of such an approach, we consider the task of invariant pattern clustering over eight data sets here. A wide range of computer simulations have been conducted in terms of internal and external indexes, such as number of communities, modularity, NMI, purity, collocation and F_1. Besides the comparisons among FG, WT and LP, we also compared such methods against conventional clustering methods, like k-means and HDBSCAN. Interestingly, our results indicates FG as the better choice among all evaluated algorithms. FG usually approximates efficiently the ground-truth groups while also keeping a reasonable number of communities. We believe that such a result can be explained due to the graph structure that provides a robust representation for the community concept, which is much less dependant of the physical features of the data (e.g., spatial distribution). Such a salient feature makes community detection methods promising methods not only to cluster data already represented as graph, but also data represented by feature vectors.

Future works should focus on extending the framework analysis by considering other graph construction and community detection methods as well as new application domains.

Acknowledgment

Authors thank the financial support given by the Brazilian National Council for Scientific and Technological Development - CNPq (grant number 439556/2018-0).

References

[1] P. Berkhin, “A survey of clustering data mining techniques,” pp. 25–71, 2006.

[2] S. Fortunato, “Community detection in graphs,” *Physics reports*, vol. 486, no. 3, pp. 75–174, 2010.

[3] T. B. S. Oliveira, L. Zhao, K. Faceli, and A. C. P. L. F. Carvalho, “Data clustering based on complex network community detection,” in *IEEE Congress on Evolutionary Computation*. IEEE, 2008, pp. 2121–2126.

[4] G. F. de Arruda, L. da Fontoura Costa, and F. A. Rodrigues, “A complex networks approach for data clustering,” *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 23, pp. 6174–6183, 2012.

[5] R. Motta, A. de Andrade Lopes, B. M. Nogueira, S. O. Rezende, A. M. Jorge, and M. C. F. de Oliveira, “Comparing relational and non-relational algorithms for clustering propositional data,” in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. ACM, 2013, pp. 150–155.

[6] O. A. Linares, G. M. Botelho, F. A. Rodrigues, and J. B. Neto, “Segmentation of large images based on super-pixels and community detection in graphs,” *IET Image Processing*, vol. 11, no. 12, pp. 1219–1228, 2017.

[7] M. E. Newman, “Fast algorithm for detecting community structure in networks,” *Physical review E*, vol. 69, no. 6, p. 066133, 2004.

[8] M. G. Carneiro, L. Zhao, and J. L. G. Rosa, “Graph-based semi-supervised learning for semantic role diffusion,” in *Symposium on Knowledge Discovery, Mining and Learning*, 2016, pp. 108–115.

[9] A. Clauset, M. E. Newman, and C. Moore, “Finding community structure in very large networks,” *Physical review E*, vol. 70, no. 6, p. 066111, 2004.

[10] P. Pons and M. Latapy, “Computing communities in large networks using random walks,” in *International symposium on computer and information sciences*. Springer, 2005, pp. 284–293.

[11] U. N. Raghavan, R. Albert, and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” *Physical review E*, vol. 76, no. 3, p. 036106, 2007.

[12] A. Lancichinetti and S. Fortunato, “Community detection algorithms: a comparative analysis,” *Physical review E*, vol. 80, no. 5, p. 056117, 2009.

[13] J. Lang and M. Lapata, “Unsupervised semantic role induction via split-merge clustering,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 2011, pp. 1117–1126.

[14] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, “A comparison of extrinsic clustering evaluation metrics based on formal constraints,” *Information retrieval*, vol. 12, no. 4, pp. 461–486, 2009.

[15] B. Leibe and B. Schiele, “Analyzing appearance and contour based methods for object categorization,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2003, pp. II–409.

[16] M. G. Carneiro, T. H. Cupertino, R. Cheng, Y. Jin, and L. Zhao, “Nature-inspired graph optimization for dimensionality reduction,” in *IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2017, pp. 1113–1119.

[17] A. Amelio and C. Pizzuti, “Is normalized mutual information a fair measure for comparing community detection methods?” in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ACM, 2015, pp. 1584–1585.

[18] T. Kamada and S. Kawai, “An algorithm for drawing general undirected graphs,” *Information processing letters*, vol. 31, no. 1, pp. 7–15, 1989.

[19] A. K. Jain, “Data clustering: 50 years beyond k-means,” *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[20] R. J. Campello, D. Moulavi, and J. Sander, “Density-based clustering based on hierarchical density estimates,” in *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 2013, pp. 160–172.

[21] L. McInnes, J. Healy, and S. Astels, “HDBSCAN: Hierarchical density based clustering,” *The Journal of Open Source Software*, vol. 2, no. 11, p. 205, 2017.