Journal of Computer & Robotics 15 (2), Summer and Autumn 2022, 27-35

Computer & Robotics

A Technical Review on Unsupervised Learning of Graph and Hypergraph Pattern Analysis

Aref Safari

Department of Computer Engineering, Shahr-e-Qods Branch, Islamic Azad University, Tehran, Iran

Received 05 August 2022; Accepted 28 October 2022

Abstract

Graph and hypergraph matching are fundamental problems in pattern analysis problems. They are applied to various tasks requiring 2D and 3D feature matching, such as image alignment, 3D reconstruction, and object or action recognition. Graph pattern analysis considers pairwise constraints that usually encode geometric and appearance associations between local features. On the other hand, hypergraph matching incorporates higher-order relations computed over sets of features, which could capture both geometric and appearance information. Therefore, using higher-order constraints enables matching that is more robust (or even invariant) to changes in scale, non-rigid deformations, and outliers. Many objects or other entities such as gesture recognition and human activities in the spatiotemporal domain can be signified by graphs with local information on nodes and more global information on edges or hyperedges. In this research, and essential review have been done on the unsupervised methods to explore and communicate meta-analytic data and results with a large number of novel graphs proposed quite recently.

Keywords: Pattern Analysis, Unsupervised Learning, Graph Theory, Unsupervised Learning

1.Introduction

The problem of finding correspondences between two graph models arises in many machine vision tasks. The types of features could vary significantly from one application to the next. We could use graph matching for registering shapes, recognizing object categories or go to more complex problems such as recognizing activities in video. Graph matching is ultimately about finding agreements between features extracted at the local level and between information computed at the higher order of edges or hyperedges (in the case of hypergraph matching). During the matching process, which is usually iterative, messages about possible node correspondences are passed through edges or hyperedges. Information is eventually spread to the whole set of potential node assignments between the two graphs, and node-to-node matching convergence is eventually reached. Note that by considering information at orders beyond the pairwise edges, hypergraph matching could be more robust to changes in scale, rotations, or other transformations in geometry and appearance at the level of groups of nodes or cliques. The interest in unsupervised learning is steadily increasing in machine learning, computer vision, and robotics research. Classical works are based on the observation that real-world data naturally groups into certain classes based on certain core, innate properties related to color, texture, form, or shape. Thus, elements that are similar based on such properties should belong to the same group or cluster, while those that are dissimilar should be put in different clusters. While these

*Corresponding Author. Email: safari.aref@gmail.com

unsupervised learning principles are not described in precise mathematical terms, they are sufficiently clear and intuitive.

Therefore, the very vast research field of clustering in machine learning was born, with a plethora of algorithms being proposed during the last fifty years in [1], which could be grouped into several main classes: (1) methods related to K-means algorithm in [2] and Expectation-Maximization (EM) in [3], which have an explicit probabilistic formulation and attempt to maximize the data likelihood conditioned on the class assignments; (2) methods that directly optimize the density of clusters, such as the Mean Shift algorithm [4],[5] and Density-Based Spatial Clustering (DBSCAN) in. [6]; (3) hierarchical approaches that form clusters from smaller sub-clusters in a greedy agglomerative fashion Day and Edelsbrunner [7-10] or divisive clustering methods (DIANA) Kaufman and Rousseeuw [11], which start from a large cluster and iteratively divide the larger clusters into smaller ones; (4) spectral clustering algorithms, which are based on the eigenvectors and eigenvalues of the adjacency matrix or the Laplacian of the graph associated with the data points Cheeger [12-16]. The clustering algorithms discussed in the present research and applied to different pattern analysis problems are mostly related to the class of spectral clustering methods. Until not so long ago, most unsupervised learning research focused on proposing and studying various kinds of clustering algorithms Safari et al. [17-19] and [20-23], and for a good reason. Most unsupervised learning tasks require, implicitly or explicitly, some clustering. We all researchers in machine learning would hope, even without saying it, that the complete structure of the world, with its entities moving, relating, and acting in different ways and being grouped into specific classes, should emerge naturally in a pure unsupervised learning setup. The discovery of such a structure with well-formed entities and relations immediately implies some data clustering.

2.Research Background

Research focusing on graph matching has been increasing in recent years [1–15], with most new approaches using or being derived from the initial IQP formulation [2, 16]. Not many papers investigate the task of learning the parameters of the graph matching model. However, interest in the topic is increasing [17–21]. Graph matching with higher-order constraints, known as hypergraph matching, is also receiving increasing attention in machine vision [22–29] mainly due to a more powerful geometric modeling, often capable of similarity or even affine-invariant matching and the increased capacity to capture-invariant information at the higher level of cliques. Another relevant research direction in graph matching is solving the task efficiently directly in the original discrete domain, in which the final solution should be [6, 27, 30, 31]. In some papers, such as [6-9], they showed in extensive experiments that optimizing sub-optimally directly in the correct discrete domain could often be largely superior to global optimization in the relaxed, continuous domain followed by a very simplistic binarization step. Another interesting direction in graph matching is to find node correspondences among several graphs, which would enable graph matching across multiple images or object models. The multi-graph matching task has found some notable approaches in the literature [32, 33], including the initial tensor formulation [34].

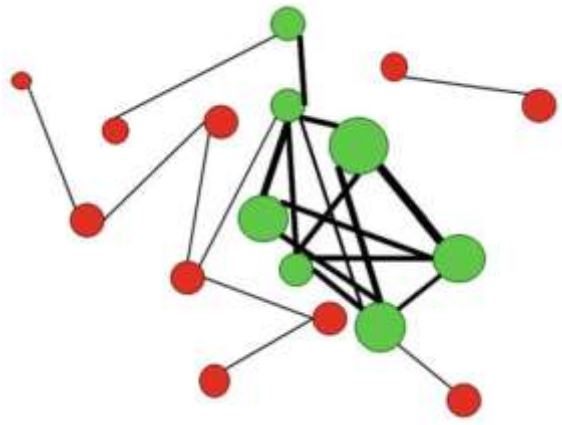

Fig. 1. Correct assignments (shown in green) are likely to form a strong cluster by establishing stronger pairwise second-order agreements (many thicker edges) and preserving better the local appearance of features (larger nodes) [11].

On the other hand, the alignments at the level of nodes and edges establish strong links in a particular

graph of candidate assignments between the nodes of the two graphs. The candidate assignments that are most likely correct through such strong agreement links will naturally form a strong cluster. Therefore, finding the correct assignments reduces to a clustering problem that often should obey certain matching constraints, such that one node from one graph could match at most one node in the other. Therefore, the connection between graph matching and clustering comes naturally, which is why algorithms that were designed for one problem (e.g., spectral clustering or integer projected fixed point for graph matching), could be easily adapted for the other problem with minimal modification (usually at the level of constraints on the solution vector). Next, we present a reliable mathematical formulation for graph matching pattern analysis. Then, based on the intuitions introduced above, we introduce spectral graph matching, and integer projected fixed point algorithms. They will be further extended to the higher order matching case (hypergraph matching) and, adapted to the task of clustering with higher order constraints. It is important to remember that all methods presented in this section, for inference or learning, are fundamentally linked to the general concept of clustering, which is at the core of all unsupervised learning.

3.Method and Materials

The material presented in this section is based on previous state-of-the-art works [19, 35]. It consists of a set of methods for graph and hypergraph matching and learning, generally considered state-of-the-art in terms of accuracy, computational speed, and overall efficiency. The applied approaches to unsupervised and semi-supervised learning for graph and hypergraph matching are the first such algorithms proposed in the machine vision pieces of literature.

3.1. Principles of Unsupervised Learning

Accidental alignments are rare. They usually indicate correct alignments between a model and an image when they happen. Alignments, which could be geometric or appearance based, rare as they are, when they take place, form a strong cluster of agreements that re-enforce each other in multiple ways. We cannot stress enough how important the above principle is for inference and learning in the case of graph matching. In the context of this task, alignments often refer to agreements in geometry or appearance between nodes and edges (or hyperedges) of the two graphs. They may also refer to similarities at higher levels of semantic abstraction. Such agreements, however, do not regularly happen accidentally in nature. Accidental alignments are indeed rare and very sparse without forming strong clusters of agreements in space and time. For example, some parts of an object may look similar to another unrelated object by accident. It is also possible that by chance, a random pattern in the continuously moving world may resemble some well-known object (e.g., the configuration of clouds that may resemble a puppy). However, it is improbable that two structures, stable in space and time, look-alike at the level of whole objects without any meaningful connection. When such agreements, at the level of shape, appearance, and full structure, are overwhelming, they usually indicate a meaningful connection between the two entities. Such a relationship could be, for example, at the level of identity (e.g., it is the same object seen from different viewpoints or seen in two different moments in time) or at the level of a semantic category (e.g., the objects are of the same kind, have a similar purpose or meaning).

3.2. Graph Matching

The graph matching problem with pairwise constraints consists of solving for the indicator vector x∗ that maximizes a quadratic score function with certain mapping constraints:

$$\mathbf{x}^* = \arg max(\mathbf{x}^T\mathbf{M}\mathbf{x}) \text{ s.t. } \mathbf{A}\mathbf{x} = \mathbf{1}, \mathbf{x} \in \{0,1\}^n \qquad (1)$$

Where **x** is an indicator vector such that $x_{ia} = 1$ if feature i from one image (or graph) is matched to feature a from the other image (or graph), and zero otherwise; $\mathbf{Ax} = \mathbf{1}$, $\mathbf{x} \in \{0, 1\}n$ enforces one-to-one constraints on **x** such that one feature from one image can be matched to at most one other feature from the other. In this work, **M** is a matrix with

positive elements containing the pairwise score functions, such that $M_{i,a;jb}$ measures how well the pair of features (i,j) from one image agrees in terms of geometry and appearance (e.g., difference in local appearance descriptors, pairwise distances, angles, etc.) with a pair of candidate matches (a,b) from the other. While the Spectral Matching (SM) algorithm requires **M** to have non-negative elements, the Integer Projected Fixed Point (IPFP) method does not impose this requirement. Also note that the local, unary terms of candidate assignments can be stored on the diagonal of **M**; in practice, we noticed that considering such local, unary information in the pairwise scores $M_{i,a;jb}$ and leaving zeros on the diagonal produce superior results; a slightly different form of the matching objective that combines both linear and quadratic terms is also possible: $\mathbf{dx} + \mathbf{x}^T\mathbf{Mx}$. In Figure 2 we show that the geometric information could be represented by rotation-invariant angles such as βi, rotation-dependent angles pairwise distances, which could be scale-invariant if divided by the perimeter. Appearance could be represented by a feature vector encoding gradient, color, or texture information. The relative importance of each cue is application-dependent and should be automatically learned during training.

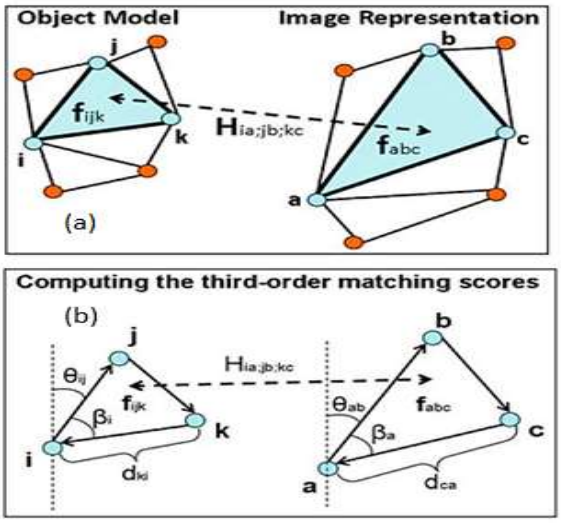

Fig.2. (a): matching the object model to the image by using third-order representations (matching scores computed over triangles, considering both geometry and appearance). (b): the third-order scores can be functions of both geometry and appearance [6].

Many algorithms, including Integer Projected Fixed Point (IPFP), can handle both problems with slight modifications. In this section, we present similar learning algorithms applicable to both problems, graph matching, and MAP Inference, to emphasize that the two tasks are closely related and give a general, encompassing view of the optimization and learning approaches. Coming back to the main Quadratic Programming (QAP) for graph matching formulation, we mention that $M_{i,a;jb}$ is essentially a function that is defined by a certain parameter vector w. The type of pairwise scores $M_{i,a;jb}$ that we use, also closely related to the type of relationships we apply to hypergraph matching, is:

$$M_{ia;jb} = \exp\left(-\mathbf{w}^T \mathbf{g}_{ia;jb}\right) \quad (2)$$

where **w** is a vector of learned parameter weights, and **g***ia*; *jb* is a vector that usually contains non-negative errors and deformations to describe the changes in geometry and appearance when we match the pair of features (i,j) to the pair (a,b).

3.3. Hypergraph Matching

Hypergraph matching follows naturally from graph matching by extending (1) to a tensor formulation that can include higher order constraints. The first hypergraph matching approaches in machine vision are the probabilistic method [24] and the tensor-based higher-order power method [22]. The formulation we present applies to hypergraph models of any order. However, in practice, third-order constraints offer a good compromise between efficiency and the ability to capture the appearance and geometry of objects. The classical graph matching approach deals with pairwise relationships between features that are not invariant under similarity, affine, or projective transformations. Invariance to such transformations is possible if third-order constraints are used [22]. Third-order relationships could still be handled efficiently while being able to model both higher-order geometry and appearance. Objects could be described by triangulated meshes, with features located at corners and appearance computed over the triangles'

interiors. In the rest of the paper, we discuss only third-order hypergraph matching, while keeping in mind that the same formulation can be extended, in principle, to matching using relations of any order. In Figure 3, we show the likely structure of the matrix **M**. The correct assignments will form a strong block in this matrix with large pairwise elements, while the pairwise scores between incorrect assignments will be mostly zero. This will be reflected in the leading eigenvector of **M**.

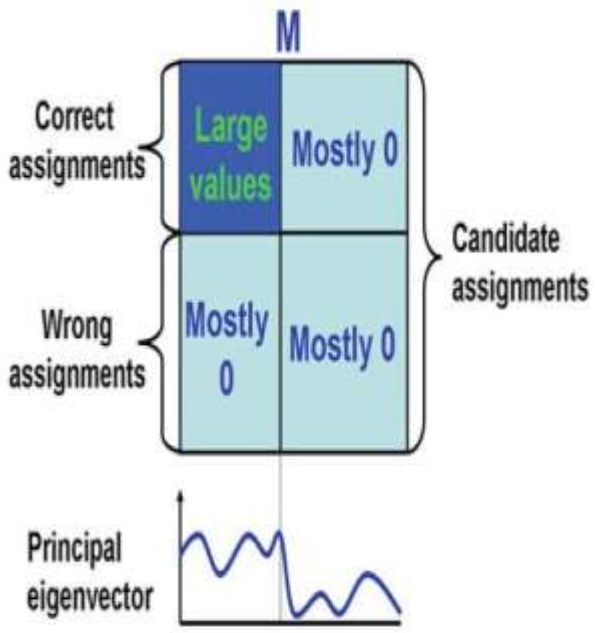

Fig.3. The structure of the matrix M: correct assignments will form a strong block in M with large pairwise elements, while the pairwise scores between incorrect assignments will be mostly zero. This statistical property of M will be reflected in its principal eigenvector [13].

Given two sets of features, one from a model image I_m and the other from a test image I_t , third-order (hypergraph) matching consists of finding correspondences between the two sets, such that a matching score, as a sum over triplets of candidate matches, is maximized. The score could consider geometric and appearance information over the triplets of potential assignments, extending the pairwise scores used for graph matching. The mapping constraints are written in matrix form $\mathbf{A}\mathrm{x} \leq 1, \mathrm{x} \in \{0,1\}\mathrm{n}$, with a binary matrix. The third-order matching score is:

$$S_{(x)} = \sum_{ia,jb,kc} \boldsymbol{H}_{ia;jb;kc,} x_{ia}, x_{jb}, x_{kc} \qquad (3)$$

Here, $\boldsymbol{H}$ a super-symmetric tensor with non-negative elements. They are increasing with the quality of the match between tuples of features $H_{ia;jb;kc,}$ indicates how well features (i,j,k) from the model image match, in terms of appearance and geometry, features (a,b,c) from the test image. Solving hypergraph matching means finding the solution that maximizes $S_{(x)}$, under one-to-one mapping constraints:

$$\mathbf{x}^* = \arg max \sum_{ia,jb,kc} \boldsymbol{H}_{ia;jb;kc,} x_{ia}, x_{jb}, x_{kc} \qquad (4)$$

4.Experiments

In this section we describe the experimental results on methods for representing web documents using graphs instead of the traditional vector representations. All representations are based on the adjacency of terms in a web document. These representations are named: standard, simple, n-distance, n-simple distance, raw frequency and normalized frequency. Under the standard method each unique term (word) appearing in the document, except for stop words such as “the,” “of,” and “and” which convey little information, becomes a node in the graph representing that document. Each node is labeled with the term it represents. Note that we create only a single node for each word even if a word appears more than once in the text. Second, if word a immediately precedes word b somewhere in a “section” s of the document, then there is a directed edge from the node corresponding to term a to the node corresponding to term b with an edge label ***s***. We take into account certain punctuation (such as periods) and do not create an edge when these are present between two words. Sections we have defined for web documents are: title, which

contains the text related to the document's title and any provided keywords (meta-data); link, which is text that appears in hyperlinks on the document; and text, which comprises any of the readable text in the document (this includes link text but not title and keyword text). Next we remove the most infrequently occurring words on each document, leaving at most m nodes per graph (m being a user provided parameter).

4.1. Dataset of the Study

In order to evaluate the performance of the graph based k-means algorithm as compared with the traditional vector methods, we performed experiments on three different collections of web documents, called the F-series, the J-series, and the K-series. These data sets were selected because of two major reasons. First, all of the original HTML documents are available, which is necessary if we are to represent the documents as graphs; many other document collections only provide a preprocessed vector representation, which is unsuitable for use with our method. Second, ground truth assignments are provided for each data set, and there are multiple classes epresenting easily understandabl groupings that relate to the content of the documents. Some web document collections are not labeled or are presented with some other task in mind than contentrelated clustering. The F-series originally contained 98 documents belonging to one or more of 20 subcategories of four major category areas: manufacturing, labor, business and finance, and electronic communication and networking. Because there are multiple subcategory classifications for many of these documents, we have reduced the categories to just the four major categories mentioned above in order to simplify the problem. There were five documents that had conflicting classifications were classified to belong to two or more of the four major categories) which we removed, leaving 93 total documents.

4.2. Performance Analysis

In the last step of this section, an ROC curve analysis was conducted to have a reliable estimate of the performance of the proposed model, and the results were statistically verified as follows [32-36]:

$$\text{Precision} = \frac{TP}{(TP + FP)} \times 100\% \tag{5}$$

$$\text{Recall} = \frac{TP}{(TP + TN)} \times 100\% \tag{6}$$

$$\text{F-Measure} = \frac{2 \text{ "Precision } ^{*} \text{ Recall}}{\text{Precision } + \text{ Recall}} \tag{7}$$

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{8}$$

$$\mu_i = \frac{1}{10} \sum_{k=1}^{10} AUCj \tag{9}$$

where μ_i is the means of the accuracy of the ROC curve for the 10-fold cross-validation. A 10-fold cross-validation technique was applied, randomly partitions the original sample into k equal sized subsamples. A single subsample is retained as the validation data for testing the model from the ten sub-samples. The remaining nine are used as training data.

Table-1
Obtained results of unsupervised methods.

Type	AUC%	Recall	Precision	F-1
Graph (House)	79.23	78.52%	79.67%	80.33%
Graph (Hotel)	82.07	81.23%	82.41%	82.77%
Hyper Graph (House)	90.92	91.79%	92.24%	90.33%
Hyper Graph (Hotel)	92.05	91.76%	92.27%	91.17%
Web Docs (Yahoo)	91.03	90.89	91.13	90.79
Web Docs (reuters)	92.07	91.98	92.11	91.87

Also, a two-sample t-test (left tailed) has been conducted. The null hypothesis was defined as H_0=

$\mu_i > \mu_j$ and, H_1: $\mu_i < \mu_j$, where μ_i and μ_j are the means of the area under the ROC curve (AUC) of assessment [20-21]. The results that validate the theoretical claims are shown in Figure 4 on the On the other hand, the details of these experiments are

5.Discussion

On all four experiments, the correlation between **v** and the ground truth **t** increases at every gradient step even though the ground truth is unknown to the learning algorithm. The matching rate improves at the same time and at a similar rate with the correlation, showing that maximizing this correlation also maximizes the final performance. We display a representative example of the eigenvector for one pair of faces as it becomes more and more binary during training. Suppose the eigenvector is almost flat at the last iteration after the first iteration. In that case, it is very close to the binary ground truth, with all the correct assignments having larger confidences than any of the wrong ones. Also, on all individual experiments, both approximations from Equations 2 and 3 become increasingly accurate with each gradient step, we started from a set of parameters w

Graph matching and Hyper-Graph matching for ten different iterations of 10-fold cross-validation given below. There are a few relevant results to consider. House and Hotel dataset experiments, respectively.

that does not favor any assignment (w = 0, which means that before the very first iteration, all non-zero scores in **M** are equal to 1).

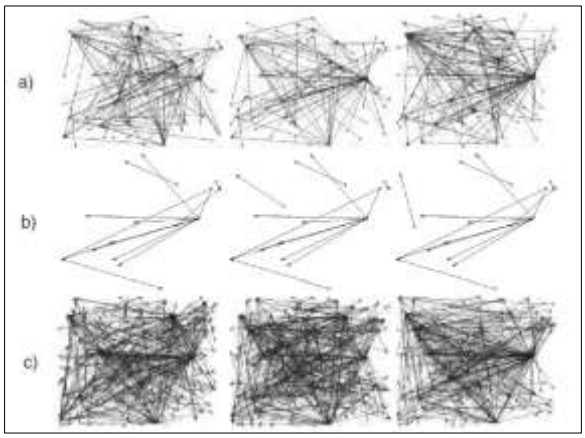

Fig.4. Graphs from the time series obtained from the a real network: (a) three graphs from one cluster; (b) and (c) graphs from different clusters.

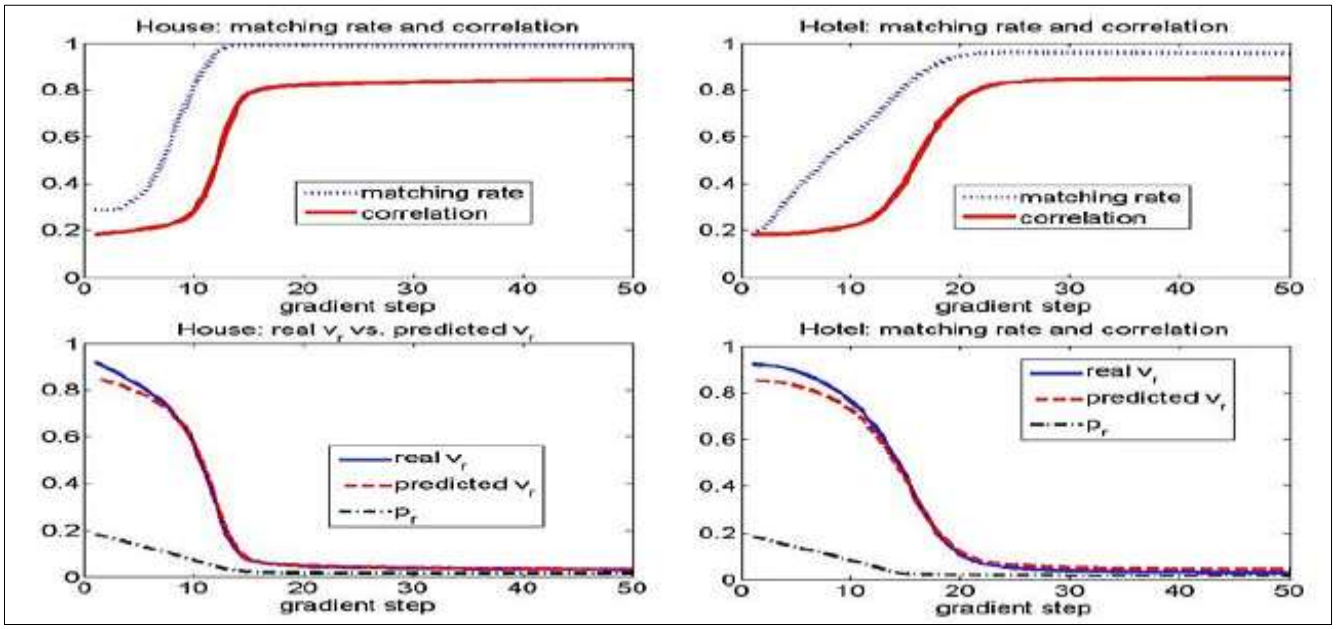

Fig.5. Unsupervised learning stage. First row: matching rate and correlation of eigenvector with the ground truth during training per gradient step.

6. Conclusion

In this work, some efficient learning and optimization methods have been described, which are sufficient for graph/hypergraph matching and MAP inference problems based on Quadratic Integer Program formulations. This work demonstrated through several extensive experiments that important algorithms have the potential to improve matching and inference accuracy significantly. Also, this research brought valuable insights and methods to graph and hypergraph matching to hopefully open new roads for further exploration, especially inefficient matching and unsupervised learning. This research handled two main aspects of matching: (1) presenting the optimal solutions with a brief review for pattern analysis in unsupervised graph and hypergraph matching problems in the continuous domain and (2) obtaining a high-quality discrete solution with important theoretical properties, unlike the traditional approaches to obtaining the final hard assignment based on greedy algorithms or the classical Hungarian method. There are many avenues for future work in matching with second- or higher-order constraints. They bring a significant boost in matching performance, as they are more robust to changes in scale, rotations, perspective or affine transformations, various appearance changes, and non-rigid deformations.

References

[1]. Gold S, Rangarajan A (1996) A graduated assignment algorithm for graph matching. PAMI 18(4):377–388

[2]. Berg A, Berg T, Malik J (2005) Shape matching and object recognition using low distortion correspondences. In: CVPR

[3]. Cour T, Srinivasan P, Shi J (2006) Balanced graph matching. In: NIPS

[4]. Schellewald C, Schnorr C (2005) Probabilistic subgraph matching based on convex relaxation. In: International conference on energy minimization methods in computer vision and pattern recognition

[5]. Torr P (2003) SolvingMarkov random fields using semidefinite programming. In: International conference on artificial intelligence and statistics.

[6]. Leordeanu M, Hebert M, Sukthankar R (2009) An integer projected fixed point method for graph matching and map inference. In: NIPS

[7]. Cho M, Lee J, Lee KM (2010) Reweighted random walks for graph matching. In: ECCV

[8]. Egozi A, Keller Y, Guterman H (2013) A probabilistic approach to spectral graph matching. Pattern Anal Mach Intell 35(1)

[9]. Zhou F, la Torre FD (2012) Factorized graph matching. In: Computer vision and pattern recognition

[10]. Cho M, Lee K (2012) Progressive graph matching: making a move of graphs via probabilistic voting. In: Computer vision and pattern recognition

[11]. Tolias G, Avrithis Y (2011) Speeded-up, relaxed spatial matching. In: ICCV

[12]. Escolano F, Hancock E, Lozano M (2011) Graph matching through entropic manifold alignment. In: Computer vision and pattern recognition

[13]. Torresani L, Kolmogorov V, Rother C (2013) A dual decomposition approach to feature correspondence, Pattern Anal Mach Intell 35(2)

[14]. Liu Z, Qiao H, Yang X, Hoi S (2014) Graph matching by simplified convex-concave relaxation procedure. Int J Comput Vis 1–18

[15]. Zhou F, la Torre FD (2013) Deformable graph matching. In: Computer vision and pattern recognition, pp 2922–2929.

[16]. A.Safari, R. Hosseini and M. Mazinani, "A Type-2 Fuzzy Time Series Model for Pattern Similarity Analysis: A Case Study on Air Quality Forecasting," in IEEE Intelligent Systems, (2021).

[17]. A.Safari, R. Hosseini, M. Mazinani, “A novel deep interval type-2 fuzzy LSTM (DIT2FLSTM) model applied to COVID-19 pandemic time-series prediction”, Journal of Biomedical Informatics, Vol 123, (2021).

[18]. Safari, A., Hosseini, R., Mazinani, M. (2022). An Adaptive Intelligent Type-2 Fuzzy Logic Model to Manage Uncertainty of Short and Long Time-Series in Covid-19 Patterns Prediction: A Case Study on Iran. Computational Intelligence in Electrical Engineering.

[19]. Cho M, Alahari K, Ponce J (2013) Learning graphs to match. In: International conference on computer vision, pp 25–32 102 2 Unsupervised Learning of Graph and Hypergraph Matching

[20]. Zanfir A, Sminchisescu C (2018) Deep learning of graph matching. In: Proceedings of the IEEE

conference on computer vision and pattern recognition, pp 2684–2693

[21]. Duchenne O, Bach F, Kweon I, Ponce J (2009) A tensor-based algorithm for high-order graph matching. In: CVPR

[22]. Chertok M, Keller Y (2010) Efficient high order matching. IEEE Trans Pattern Anal Mach Intell 32:2205–2215

[23]. Zass R, Shashua A (2008) Probabilistic graph and hypergraph matching. In: CVPR

[24]. Çeliktutan O, Wolf C, Sankur B, Lombardi E (2014) Fast exact hyper-graph matching with dynamic programming for spatio-temporal data. J Math Imaging Vis 1–21

[25]. Wang J, Chen S, Fuh C (2014) Attributed hypergraph matching on a Riemannian manifold. Mach Vis Appl 25(4)

[26]. Yan J, Zhang C, Zha H, Liu W, Yang X, Chu SM (2015) Discrete hyper-graph matching. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1520–1528

[27]. NieWZ, Liu AA, Gao Z, Su YT (2015) Clique-graph matching by preserving global and local structure. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4503–4510

[28]. Liu AA, Nie WZ, Gao Y, Su YT (2016) Multi-modal clique-graph matching for view-based 3d model retrieval. IEEE Trans Image Process 25(5):2103–2116

[29]. Adamczewski K, Suh Y, Mu Lee K (2015) Discrete Tabu search for graph matching. In: Proceedings of the IEEE international conference on computer vision, pp 109–117

[30]. Jiang B, Tang J, Ding C, Luo B (2017) Binary constraint preserving graph matching. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4402–4409

[31]. Shi X, LingH,HuW, Xing J, Zhang Y(2016) Tensor power iteration formulti-graphmatching. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5062–5070.

[32]. Safari, A. (2021). Deep-Quantitative Medical Image Analysis Methods Applied on Brain Tumor Diagnosis. *Signal Processing and Renewable Energy*, 5(2), 1-13.

[33]. Safari, A. (2019). MCSM-DEEP: A Multi-Class Soft-Max Deep Learning Classifier for Image Recognition. *Journal of Advances in Computer Research*, 10(4), 75-85.

[34]. Safari, A., Mazinani, M., Hosseini, R. (2017). A Novel Type-2 Adaptive Neuro-Fuzzy Inference System Classifier for Modelling Uncertainty in Prediction of Air Pollution Disaster (RESEARCH NOTE). International Journal of Engineering, 30(11), 1746-1751.

[35]. Aref Safari, Danial Barazandeh, Seyed Ali Khalegh Pour. A Novel Fuzzy-C Means Image Segmentation Mode for MRI Brain Tumor Diagnosis. J. ADV COMP ENG TECHNOL, 6(1) Winter 2020: 19-2

[36]. Safari A, Hosseini R, Mazinani M. Dynamic Type-2 Fuzzy Time Warping (DT2FTW): A Hybrid Model for Uncertain Time-Series Prediction. IJFIS 2021; 21:338 348.