Graph Embedding through Probabilistic Graphical Model applied to Symbolic Graphs

Hana Jarraya, Oriol Ramos Terrades, and Josep Lladòs

Universitat Autònoma de Barcelona, Computer Vision Center CVC Campus UAB,
Edifici O, s/n, 08193 Cerdanyola del Vallès, Barcelona, Spain
{hanajarraya, oriolrt, josep}@cvc.uab.es

Abstract. We propose a new Graph Embedding (GEM) method that takes advantage of rich structural pattern representation. We learn the Probabilistic Graphical Model (PGM) parameters by using Structured Support Vector Machines (SSVM) and then we use it in the Attributed Graph (AG) signature in a lower dimensional vectorial space. In this paper, we report our encouraging first results on the GREC dataset. Although they are not better that the current state-of-the-art, they are good enough to continue our research on it.

Keywords: Attributed Graph, Probabilistic Graphical Model, Graph Embedding, Structured Support Vector Machines

1 Introduction

Nowadays there are many tools that use Pattern Recognition (PR) techniques to help us in our daily tasks. Such tools automatically extract features from input signals and, using machine learning techniques are able to classify, suggest, identify a wide range of objects. However, the complexity of some of these tasks make that their overall performance are still far of being satisfactory enough. Structural PR is a field that encodes structural information of input signals to enrich the representation space and it improves the overall performance of PR systems. They extract line segments and organize them into data structures such as graphs. These data structures model two important aspects: the hierarchical composition of sub-patterns from complex pattern; and the relations between these sub-patterns. This problem is presented by the capacity of finding sub-structures, sub-graph matching, and the computational complexity of matching algorithm. Moreover, the difficulty is to define a general representation. Each problem has their specific graph representation with their own set of attributes [1–3]. An other confrontation is to reduce the sensitivity to noise, error tolerance.

In the last three decades, explicit Graph EMbedding (GEM) presents actually a transformation in representation spaces. It allows to define approximate polynomial solution for hard combinatorial problems. Among the advantages of this

tool, we cite the dimensionality reduction, the sensitivity to noise and structural information preservation. [4] presents recent works that are using the explicit GEM applied on some PR applications. These works try to preserve the structure of data and avoid the noise as much as possible.
Here, we think that PGM can be useful since it is strong framework to define probabilities over complex structure for these reasons: The probabilities allow us to consider options that are unlikely, exclusive and exhaustive distortion possibilities that can happen. Therefore, they are able to settle the noise sensitivity in graphs since the errors are kind of extreme potentials. And The PGM presents graph structure. So, we can say that PGM could be a solution for structure reservation and noise reduce. The learning over PGM is a delightful traditional problem [5, 6], called Structured Learning. One of the recent successful techniques for prediction in Structured models is Structured Support Vector Machines (SSVM). It is based on the unsupervised learning of a parameter vector that characterizes the PGM taking into account of its different interactions and attributes. Due to this vector, the SSVM can predict within an inference task the probabilities to decide the class of one graph. The difficulties at this point are two: first define the different interactions and attributes; and second, parameter learning. The first belongs to a modeling problem. And the experience tells us which kind of attributes we should use to model our task. The second is related to the parameter learning problem and herein, SSVM is a technique that we can use to solve this problem. We can find more on parameter learning in [6].
In this paper we propose the use of PGM for GEM in order to estimate a parameter vector built from the graph attributes and structure. We have validate our proposed model on a symbol dataset, GREC graphs [7]. The first obtained results are good enough to encourage our research in that direction. The literature demonstrates other important GEM methods applied on symbol recognition [2, 8–12], we will only consider the latest to compare our idea with. We have structured the rest of the paper as follows. In section 2, we illustrate the related works on Explicit GEM methods applied on Symbol Recognition. We introduce the definitions and notations of the main concepts used, in section 3. In Section 4, we detail the proposed model. In section 5, we discuss the tests applied on the learning parameters and interpret the false classified graphs. Section 6 presents the conclusion and future work.

2 Related works

GEM approaches map either explicitly or implicitly graphs into high dimensional spaces as we can perform the basic mathematical computations required by various statistical PR techniques. The implicit GEM methods are based on graph kernel which is a function defined as a dot product evaluated in graph space. A strict limitation of implicit GEM is that it does not permit all operations that could be defined on vector spaces [4]. In an other way, explicit GEM methods explicitly embed an input graph into a feature vector and thus enable the use of all the methodologies and techniques devised for vector spaces. This feature vector is of fixed size no matter is the size and order of graphs. It requires to contain relevant representative features at the same time fairly is well generic to describe any input graph. The most important challenges of explicit GEM are the loss of structural information, attributes encoding and noisy impact [4]. We can cite some explicit GEM approaches from the state-of-the-art that tackle with these problems in [4].

Topological Embedding method has important properties that are the topology preservation and label encoding [13]. It uses a generic lexicon of topological structures as a non isomorphic graphs network composed of n edges up to N.

The Attributes Statistic based Embedding method maps an AG into a naive feature vector [14]. It computes the frequencies of simple sub-graph. It is limited to node coordinates space attributes and no consideration for edge attributes.

The Fuzzy Multilevel Graph Embedding (FMGE) [2] method defines a vector based on learning of histogram fuzzy intervals. It uses fuzzy logic to reduce noise, substructures homogeneity and topology information to reserve the structure and encoding labels edges and nodes to be generic.

So, for our work, we use the PGM to be generic, the structured learning to reduce loss of information and the substructures homogeneity and topology information to save the topology. Moreover, we cite [6] for structured training algorithms of PGM such as SSVM (N-slack formulation), gradient descend, stochastic gradient descend and sub-gradient methods. We chose to train the parameter vector of PGM with SSVM algorithm.

3 Definitions and Notations

We define the concepts: explicit GEM, AG and Morgan Index (MI) [2, 3, 15].

Definition 1. *Attributed graph (AG): Let A_V and A_E denote the domains of possible values for attributed vertices and edges respectively. These domains are assumed to include a special value that represents a null value of a vertex or an edge. Here, the term attributed graph is used to refer to an undirected attributed graph. An AG over (A_V, A_E) is defined to be a four-tuple:*

$$AG = (V, E, \mu^V, \mu^E)$$

where V is a set of vertices, $E \subseteq V \times V$ is a set of edges, $\mu^V : V \mapsto A_V^k$ V is function assigning k attributes to vertices and, $\mu^E : E \mapsto A_E^l$ is a function assigning l attributes to edges. The order of an AG is given by $|V|$ i.e. the number of vertices in AG. The size of AG is given by $|E|$ i.e. the number of edges in AG. The degree of a vertex v_i in AG refers to the number of edges connected to v_i.

Definition 2. *Explicit Graph Embedding: Given an AG, explicit GEM is a function ϕ, which maps graph AG from graph space G to a point $(f_1, f_2, \ldots, f_n)$ in n dimensional vector space $\mathbb{R}^n$. It is given as:*

$$\begin{aligned} \phi : G &\mapsto R^n \\ AG &\mapsto \phi(AG) = (f_1, f_2, \ldots, f_n) \end{aligned}$$

Definition 3. *Morgan Index (MI): Given an AG and a node $v \in AG$, [15] defines the Morgan Index of v as:*

$$MI_i(v) = \begin{cases} MI_0(v) = node\_degree(v), & \text{if } i = 0 \\ \sum_u MI_{i-1}(u), & \text{Otherwise} \end{cases}$$

where u is a node adjacent to v, the MI of level 0 is the node degree, i is the level of Morgan Index and $MI_{i-1}(u)$ is the summation of the adjacent nodes degree of v in previous iteration $i-1$ of the propagation MI technique.

Definition 4. *A Probabilistic Graphical Model is defined by observed and hidden variables X and Y as nodes of the graphical model and unary and pairwise potentials Φ and Ψ as edges. The joint feature function after the parameter learning defined by the conditional probability distribution:*

$$P(Y|X,\omega) = \frac{1}{Z(X,\omega)} \exp\left(\sum_{i \in nodes} \omega_i \Phi(x_i, y_i) \sum_{(i,j) \in edges} \omega_{i,j} \Psi(x_{i,j}, y_i, y_j) \right)$$

where the partition function is:

$$Z(X,\omega) = \sum_{x \in nodes} \exp\left(\sum_{i \in edges} \omega_i \Phi(x_i, y_i) \sum_{(i,j) \in E} \omega_{i,j} \Psi(x_{i,j}, y_i, y_j) \right)$$

where $\omega = \{\omega_i, \omega_{i,j}\}$ is the set of unary and pairwise PGM learned parameters.

4 Graph EMbedding through Probabilistic Graphical Model

To propose a new GEM idea, we thought about complex structure presented in PGM. The training over a dataset defined by SSVM leads to the parameterized conditional distribution of labels in PGM. It aims to find the best parameter vector ω^* that makes the probability distribution $p(y|x,\omega)$ close to $p(y|x)$. The structured learning problem formulation is explained in two steps: first, unsupervised learning of PGM parameters; second, this parameter vector should maximize to approximate the computation of predicted probabilities over the random variables in PGM. Here, we choose to use the Structural SVM to train the PGM that represent our AG. We cite the N-Slack formulation of SSVM for the training in our work in [5]. Our challenges are to answer these questions about, learning (a) Are we using the best learning algorithm for our problem? (b) Do we have enough data for training?; and regarding the PGM used to represent AG: Are we using the best representation? Can be other feature function that better model the graph attribute?

4.1 Graph attributes and structure

As application framework in this paper, we have defined AGs or graphical symbols from the GREC database. The graph nodes represent x, y and *type* and graph edges $frequency$, *type* and *angle*. Here, we explain the different types of encoded attributes associated to the graph and the different types of used graph structures. First, we consider the following attributes: (a) Geometrical details: geometrical attributes associated to the nodes and edges of the AG, extracted from symbol drawing such as length and orientation of line and coordinates x and y of intersection points; (b) Structural details: defined by node degree, subgraph homogeneity defined in [2]; (c) Topological details: MI, with fixed level. Three different graph structures have been studied:(i) Full k-connected graph: the graph consists on connected attributed nodes with attributed edges, or hole AG; (ii) Void k-connected graph: the graph is composed of connected attributed nodes extracted from the AG; (iii) Unconnected graph: we consider only unconnected attributed nodes derived from AG. We embed each numeric/symbolic attribute in each graph. Also, structural and topological attributes also are added to nodes and edges. We used a Condition Random Field (CRF) model for the cases of graph structures are unconnected graph and void k-connected graph. And in the case of full connected graph, we used the Edge feature CRF model.

4.2 Model the AG as PGM

We built the PGM based on an AG. We are given an $AG = (V, E, \mu^V, \mu^E)$ and a PGM defined by X are observed variables and Y are hidden variables and dependencies between these variables. x_i defines the label for the node v_i based on its attributes μ^{v_i} and y_i is hidden variable associated to it. The variables are connected according to AG structure of the input data, defined as the conditional independence structure of PGM. Thus, $x_{i,j}$ defines the attributes of the edge (i,j). The distribution probability defined over the PGM previously is based on the definition 4 and the notation defined here based on the AG as input data. The output of the proposed explicit GEM, is the n dimensional learned parameters vector defined by $\omega = (\omega_i, \omega_{i,j})$. And n is the sum of Unary matrix dimension U and Pairwise matrix dimension P where U number of classes $\times$ number of node attributes, and P number of $classes^2 \times$ number of edge attributes. $\mathcal{Y}^n$ is the n

dimensional space that represents all the possible label configurations. We deal with classification problem. The label for the entire AG can be one between 22 classes of GREC dataset. We use a majority vote strategy to assign a label to AG. On other words, we count the occurrences of the labels over all the nodes of one AG, the result label is the most occurring, chosen as AG label. The node labeling problem $y\prime$ is defined as maximization problem : $y\prime = \text{argmax}_{y \in \mathcal{Y}^n}\, y$. We see in Fig. 1 an example of modeling a symbol of shape square as undirected graphical model, PGM where the label of the graph is Y_G.

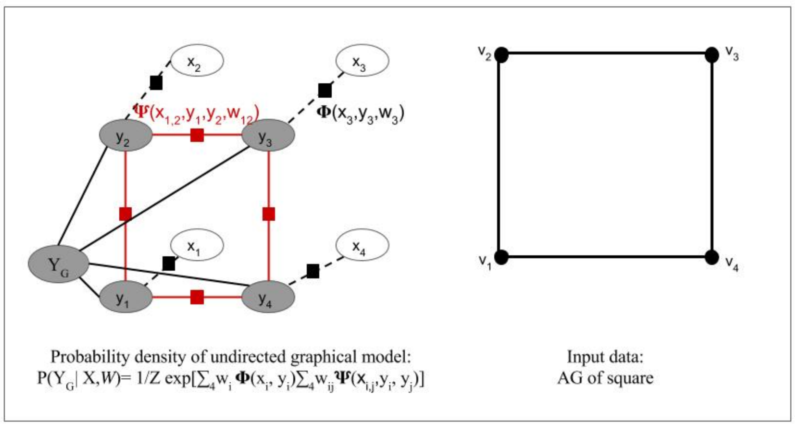

Fig. 1: an AG example of square symbol with four nodes connected with for edges where x_i is the one node attributes and $x_{i,j}$ is the one edge attributes represented by Probabilistic Graphical Model.

5 Experimentation

We test our method based on GREC dataset because its graphs have numeric and symbolic attributes. GREC Graphs are constructed as follows. The images occur at five different distortion levels. For each distortion level one example of a drawing is given. Depending on the distortion level, either erosion, dilation, or other morphological operations are applied. Finally, graphs are extracted from the resulting denoised images by tracing the lines from end to end and detecting intersections as well as corners. Ending points, corners, intersections and circles are represented by nodes and labeled with a 2-dimensional attribute giving their position. The nodes are connected by undirected edges which are labeled as line or arc. An additional attribute specifies the angle with respect to the horizontal direction or the diameter in case of arcs. For an adequately sized set, all graphs are distorted 9 times to obtain a data set containing 1,100 graphs uniformly distributed over the 22 classes. The maximum number of nodes and edges is respectively 25 and 30. They are balanced between training (286) and validation (286) and test (528). We computed some attributes as explained in 4.1. The level is equal to 2 for MI. We used Structured training library pystruct [16], particularly SSVM on its N-Slack formulation. Next, we performed several experiments to study the performance of the proposed GEM. First, we study the stability of the N-Slack algorithm depending on 3 parameters: the number of iterations, the regularization parameter C and the inference algorithm. Second, we evaluate the impact of the different attributes on the Accuracy Rate (AR). Then, we analyze the classification errors.

5.1 Training algorithm parameters impact

We study the strength of the training algorithm SSVM. We consider the three parameters of SSVM. The first parameter is the iteration that is the maximum number of steps over dataset to find the constraints of the SSVM algorithm. Secondly, the parameter C is the penalization parameter of the SSVM as defined in [5].The third parameter is the inference algorithm used on training, by the N-Slack algorithm and also on when building the GEM. First, while varying the iteration parameter, we get the same results of accuracy rate for the three inference algorithms. So, this parameter has no influence on SSVM.
Then, we differentiate the inference method depending on the parameter C. Then, we execute with three approaches: Quadratic Pseudo-Boolean Optimization (QPBO), approximate maximum a posteriori (ad3) and belief propagation as an iterative, local, message-passing algorithm for finding the maximum a posteriori (max-product). We vary C from 4×10^5 to 8×10^5 randomly. We notice that AR varies from 9% to 77%. The AR is increasing respectively with the increase of C, from appreciatively 9% to 73% for QPBO, from 61% to 73% for ad3 and from 63% to 67%. We conclude that QPBO is the best compared to the other methods over the interval of C values in a way it always offers higher AR, as shown in red plot in Fig. 2.

5.2 Impact of Structural and Topological Attributes

We want to evaluate the impact of attributes and structure on the performance of our GEM approach. The graph structure is varying within the different structures defined in 4.1: unconnected graph, Void k-connected graph and Full k-connected graph. The used attributes can be: geometrical attributes, structural attributes and topological attributes. And, for the model can be CRF or EdgeFeatureCRF in pystruct. And we use the inference unary for the case of unconnected graph and QPBO for the rest cases.

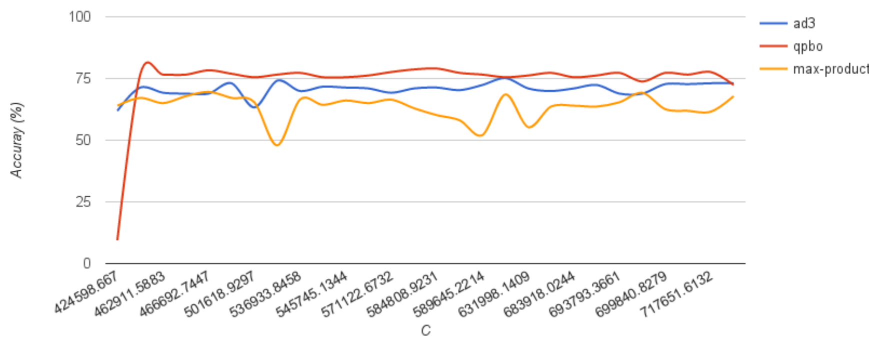

Fig. 2: Impact of Inference methods.

Input	Accuracy Rate(AR %)		
Graph str./enriched	geometrical att.	structural att.	topological att.
Full k-connected	59.1	71.4	76.22
Void k-connected	25.95	72.35	79.55
unconnected	16.66	15.34	32

Table 1: SVM Classification GREC DataSet.

Table 1 presents Accuracy Rates (AR) while learning more information, for GREC dataset with best configuration of its parameters for each case of input data. In fact, while varying the combination of parameters values, we get close deviated ARs. That's why, we addressed the problem to choose the best configuration of parameters by quadratic polynomial regression function applied on the ARs corresponding to a list of a fixed random combinations values. The ARs have been obtained by employing Structural SVM classifier. Table 1 shows that the structural attributes added to the graph structure offer better ARs than geometrical attributes. And, the topological attributes provide better ARs than the structural attributes. This is first because of the fact that the number of attributes for each graph is increasing while adding the structural and topological attributes. In addition to that, this is thanks to the fact that they provide severely discerning information about the graph. The embedded graph development clearly shows that our Embedding process gets a discriminatory power from structural and topological attributes.
We compare our model with the FMGE [2] based on classification task. FMGE solves the problem of noise sensitivity and topology reservation by Fuzzy Logic and homogeneity information but our model solves it by learning structured dependencies in a probabilistic model. Our method provides 79% as AR. And FMGE presents 99.4%. Our current results are not sufficiently competitive with

the results of FMGE. Future work will focus on improving them by providing more training graphs.

5.3 Analysis of classification errors

We answer the question: Why do we have such classification errors? We will analyze the false classified graphs for the best configuration of $C \simeq 536$ and the input data are Full k-connected graphs with topological attributes. Table 2 gives some statistic about classification errors. It summarizes the percentage of bad classified graphs. Each ground truth class is confused with a wrong predicted class with a percentage. This percentage is the number of assigned graphs to this wrong predicted class. For the first example, the class 10 is confused with class 1(Fig. 3). We notice that the graph that includes inside it unconnected small parts are negligible. Other examples of misclassified graphs, have the same problem. The GEM is neglecting the small parts outside connected to big parts. For the second example, the edge in class 5 is of type arc. So here, our GEM considers it as similar to an edge of type line. The second and third example are showing the lost of structure. For other examples, the model shows invariance to rotation. We categorize, somehow, the kind of errors as the following (see Fig. 3):

- (i) invariance to rotation;
- (ii) negligence of : (a)the small parts that are: inside connected, outside connected, inside unconnected;(b) small length edges. The edge related to too close 2 nodes is negligible comparing to the longest edge;
- (iii) confuse between geometrical attributes: for edges arc and line, for nodes the coordinates (consider the two nodes as one) and sees two parallel edges as one.

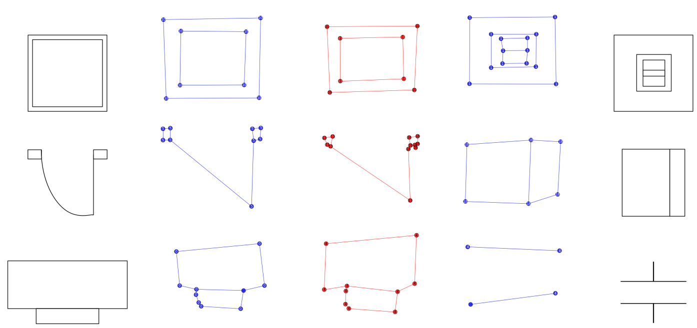

Fig. 3: Three miss-classified graphs examples. 1^{st} row: (left)ground truth of class 10, (middle)graph from class 10 labeled as class 1, (right)ground truth of class 1. 2^{cd} row: (left)ground truth of class 5, (middle)graph from class 5 labeled as class 18 ,(right)ground truth of class 18. 3^{rd} row: (left)ground truth of class 5, (middle)graph of class 0 labeled as class 4, (right)ground truth of class 4.

GT	10	5	0	1	4	2	10	16	7	15	18	1	3	17	18	1	4	8	11	15	21
Pred	1	18	4	15	0	8	6	0,21	1	0	2	6	7	2	8	7	20	6,9,16	2	6	0,6
AR(%)	41	33	29	25		20.8			16.6			12.5				8.3					

Table 2: SVM Classification GREC DataSet with $C \simeq 536,934$.

6 Conclusion

We've proposed a general scheme for building explicit GEM. This scheme is demonstrated by learning the parameters of PGM based on AG structure and its attributes. This vector is the proposed signature for the graph in a vectorial space. Our GEM is taking advantage of the learning and computational strengths of state-of-the-art of PGM learning and rich graph representation to be a new tool for classification. Current results seem to be encouraging. To improve, we need to provide more data for the training of our approach which can be a problem in some cases. For future perspective we can focus on performing more tests on comparing between structured learning methods.

References

1. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. International journal of pattern recognition and artificial intelligence **18**(03) (2004) 265–298
2. Luqman, M.M., Ramel, J.Y., Lladós, J., Brouard, T.: Fuzzy multilevel graph embedding. Pattern Recognition **46**(2) (2013) 551–565
3. Fankhauser, S., Riesen, K., Bunke, H., Dickinson, P.: Suboptimal graph isomorphism using bipartite matching. International Journal of Pattern Recognition and Artificial Intelligence **26**(06) (2012) 1250013
4. Conte, D., Ramel, J.Y., Sidère, N., Luqman, M.M., Gaüzère, B., Gibert, J., Brun, L., Vento, M.: A comparison of explicit and implicit graph embedding methods for pattern recognition. (2013) 81–90
5. Joachims, T., Finley, T., Yu, C.N.J.: Cutting-plane training of structural svms. Machine Learning **77**(1) (2009) 27–59
6. Nowozin, S., Lampert, C.H.: Structured learning and prediction in computer vision. Foundations and Trends in Computer Graphics and Vision **6**(3–4) (2011) 185–365
7. Riesen, K., Bunke, H.: Iam graph database repository for graph based pattern recognition and machine learning. (2008) 287–297
8. Tombre, K., Tabbone, S., Dosch, P.: Musings on symbol recognition. (2005) 23–34
9. Dutta, A., Lladós, J., Bunke, H., Pal, U.: A product graph based method for dual subgraph matching applied to symbol spotting. (2013) 11–24
10. Ho, H.N., Rigaud, C., Burie, J.C., Ogier, J.M.: Redundant structure detection in attributed adjacency graphs for character detection in comics books. (2013)
11. Rusinol, M., Karatzas, D., Lladós, J.: Spotting graphical symbols in camera-acquired documents in real time. (2013) 3–10

12. Santosh, K., Wendling, L.: Graphical symbol recognition. Wiley Encyclopedia of Electrical and Electronics Engineering (2015)
13. Sidere, N., Héroux, P., Ramel, J.Y.: Vector representation of graphs: Application to the classification of symbols and letters. In: 2009 10th International Conference on Document Analysis and Recognition, IEEE (2009) 681–685
14. Gibert, J., Valveny, E., Bunke, H.: Graph embedding in vector spaces by node attribute statistics. Pattern Recognition **45**(9) (2012) 3072–3083
15. Morgan, H.: The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. Journal of Chemical Documentation **5**(2) (1965) 107–113
16. Müller, A.C., Behnke, S.: Pystruct: learning structured prediction in python. Journal of Machine Learning Research **15**(1) (2014) 2055–2060