A Novel Stochastic Attributed Relational Graph Matching Based on Relation Vector Space Analysis

Bo Gun Park, Kyoung Mu Lee, and Sang Uk Lee

School of Electrical Eng., ASRI, Seoul National University,
151-600, Seoul, Korea
gun@diehard.snu.ac.kr, kyoungmu@snu.ac.kr, sanguk@sting.snu.ac.kr

Abstract. In this paper, we propose a novel stochastic attributed relational graph (SARG) matching algorithm in order to cope with possible distortions due to noise and occlusion. The support flow and the correspondence measure between nodes are defined and estimated by analyzing the distribution of the attribute vectors in the relation vector space. And then the candidate subgraphs are extracted and ordered according to the correspondence measure. Missing nodes for each candidates are identified by the iterative voting scheme through an error analysis, and then the final subgraph matching is carried out effectively by excluding them. Experimental results on the synthetic ARGs demonstrate that the proposed SARG matching algorithm is quite robust and efficient even in the noisy environment. Comparative evaluation results also show that it gives superior performance compared to other conventional graph matching approaches.

1 Introduction

Object recognition has been one of the most challenging problems in computer vision for several decades. Since an object can be distinguished from other objects by its own structure, the description of a structured object in terms of its primitives pats and mutual relations between them, has been an important issue in artificial intelligence. Due to its representational power, the graph models has been widely used for formulating such structured abstract pattern. Among various kinds of graphs, attributed relational graph (ARG) has been considered as the most effective data structures, which consists of nodes and attribute vectors for encoding unary properties and mutual relations [6] [9].

So far, a great deal of works have been reported in the literature for developing efficient and robust graph matching techniques. According to [8], the graph matching algorithms can be categorized into two classes: Search-based approach and optimization-based approach. Search-based approaches [6][7][9][10] construct the states-spaces representing graph matching status, which are searched with techniques similar to the tree search [6] in order to find the optimal solution satisfying some criteria. In general, most of them have the

J. Blanc-Talon et al. (Eds.): ACIVS 2006, LNCS 4179, pp. 978–989, 2006.

exponential complexity in the worst case, although a few methods [9] showed a high-order polynomial complexity with the help of heuristics. On the other hand, optimization-based approaches including relaxation labeling [4] [5], simulated annealing [11], genetic algorithms [12], and interpolator-based algorithms [15] [16] consider the graph matching as one of the energy minimization problems. Most of them try to find the sub-optimal solution in a continuous state space based on heuristic methods, that usually take polynomial time [4] [5]. In real applications, however, the graph representation that exhibits the morphology of the graph and the attributes is often corrupted by noise or imprecise abstraction. In order to cope with these uncertainties in the graph structure and attributes, probabilistic graph models have been introduced [1] [4] [5]. Wong *et al.* [1] interpreted an ensemble of ARGs as the outcomes of a random graph, in which nodes, edges, and all attributes were random variables with the probability density function trained from sample graphs by using supervised learning. Sanfeliu *et al.* [2] [3] presented the function-described graph (FDG) as the extension of the random graph, which is the compact representation of a set of ARGs. In order to alleviate the statistical independence of nodes and edges, some qualitative knowledge of the second-order probabilities of the elements was incorporated into FDGs. Recently, a new partial ARG matching approach was proposed that introduced the relation vector space concept to cope with large variation of attributes and partial matching [18]. However, since it can not incorporate the unary information into the model, its performance and the applications are very limited.

In this paper, we propose a new robust stochastic partial ARG matching technique which utilizes both the binary relation and unary attribute information in the relational vector space. The proposed graph matching algorithm consists of two phases: In the first stage, the candidate subgraphs are extracted and sorted according to the correspondence measure, which are based on the stochastic analysis in the relation vector space. This process significantly reduces the number of possible matches, and in the result the proposed algorithm has the polynomial computational complexity. Then, missing nodes for each candidates are identified by the iterative voting scheme through the error analysis until no more node is found to be missed, and the final subgraph matching is carried out effectively by excluding them.

2 Attributed Relational Graph

2.1 Definition of ARG

Let us define an ARG with N nodes as

$$\begin{aligned} &\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{U}, \mathcal{B}, \mathcal{F}), \qquad (1)\\ &\mathcal{V} = \{v_1, \ldots, v_N\}, \;\; \mathcal{E} = \{e_{ij} | i = 1, \ldots, N, j = 1, \ldots, N, i \neq j\},\\ &\mathcal{U} = \{\mathbf{a}_i | i = 1, \ldots, N\}, \;\; \mathcal{B} = \{\mathbf{r}_{ij} | i = 1, \ldots, N, j = 1, \ldots, N, i \neq j\},\\ &\mathcal{F} = \{\mathcal{R}_i | \mathcal{R}_i = \{\mathbf{r}_{ij} | v_i, v_j \in \mathcal{V}, i \neq j\}, i = 1, \ldots, N\}, \end{aligned}$$

where $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes and edges in the graph, respectively. If the edge between node v_i and v_j exists, e_{ij} is equal to 1, otherwise, it is 0. And $\mathbf{a}_i$ is an N_U-dimensional unary attribute vector of the node v_i, and $\mathbf{r}_{ij}$ denotes the N_B-dimensional binary attribute vector of the edge connecting node v_i and v_j. $\mathcal{F}$ is the set of relation vector spaces that encode the structural information centered at each node [18].

2.2 Preliminaries of ARG Matching

Definition 1 (Attribute matrices). *The unary attribute matrices* $\mathbf{U}_i \in R^{N\times 1}$ *and the binary attribute matrices* $\mathbf{B}_j \in R^{N\times N}$ *of a graph* $\mathcal{G}$ *with* N *nodes are defined by*

$$\mathbf{U}_i = [\mathbf{a}_1(i)\cdots\mathbf{a}_N(i)]^T, i = 1,\ldots,N_U, \tag{2}$$

$$\mathbf{B}_i = \begin{bmatrix} \mathbf{r}_{11}(i) & \cdots & \mathbf{r}_{1N}(i) \\ \vdots & \mathbf{r}_{lm}(i) & \vdots \\ \mathbf{r}_{N1}(i) & \cdots & \mathbf{r}_{NN}(i) \end{bmatrix}, i = 1,\ldots,N_B, \tag{3}$$

where $\mathbf{a}_i(k)$ *and* $\mathbf{r}_{ij}(k)$ *represent the* k*-th elements of* $\mathbf{a}_i$ *and* $\mathbf{r}_{ij}$*, respectively.*

Assume that two ARGs, $\mathcal{G}$ and $\mathcal{G}'$, are given. Messmer *et al.* [7] rigorously defined the graph isomorphism and the subgraph isomorphism by establishing the linear relation between the attribute matrices of two graphs through the permutation matrix. However, they only considered the ideal case, that is, without noise. Thus, in this section, we generalize the graph isomorphism and the subgraph isomorphism to cope with the corruption due to noise.

Definition 2 (Graph isomorphism). *Two graphs* $\mathcal{G}'$ *and* $\mathcal{G}$ *are called isomorphic if there exists an* $N' \times N$ *permutation matrix* $\mathbf{P}$ *such that*

$$\mathbf{U}'_i = \mathbf{P}\mathbf{U}_i + \epsilon\mathbf{C}_i^U\mathbf{N}_i^U, \tag{4}$$

$$\mathbf{B}'_i = \mathbf{P}\mathbf{B}_i\mathbf{P}^T + \epsilon \begin{bmatrix} \ddots & & \\ & Sum(\mathbf{C}_{lm}^B * \mathbf{N}_i^B) & \\ & & \ddots \end{bmatrix},$$

where $\mathbf{N}_i^U \in R^{N'\times 1}$ *and* $\mathbf{N}_j^B \in R^{N'\times N'}$ *are the noise matrices of which components are statistically independent.* $\mathbf{C}_i^U \in R^{N'\times N'}$ *and* $\mathbf{C}_{lm}^B \in R^{N'\times N'}$ *are the noise correlation matrices.*

In (4), operator $'*'$ represents the component-wise multiplication operation, and the function $'Sum(\cdot)'$ is the sum of all elements of the matrix $\cdot$.

Definition 3 (Subgraph isomorphism). *Two graphs* $\mathcal{G}'$ *and* $\mathcal{G}$ *are called subgraph isomorphic if there exist two sub-graphs,* $\hat{\mathcal{G}} = (\hat{\mathcal{V}},\hat{\mathcal{E}},\hat{\mathcal{U}},\hat{\mathcal{B}},\hat{\mathcal{F}}) \subset \mathcal{G}$ *and* $\hat{\mathcal{G}}' = (\hat{\mathcal{V}}',\hat{\mathcal{E}}',\hat{\mathcal{U}}',\hat{\mathcal{B}}',\hat{\mathcal{F}}') \subset \mathcal{G}'$ *that are isomorphic.*

From the above definitions, the ARG matching problem can be thought as the ARG matching by the subgraph isomorphism, that is the extraction of subgraphs and the inference of $\mathbf{P}$ satisfying (4). In other words, it can be termed as "*correspondence problem*" between two ARGs [10] [16].

3 Proposed SARG Matching Algorithm

3.1 Correspondence Measure

As stated in Section 2, the graph matching problem can be transformed into the correspondence problem. So, let us define some basic concepts related to the correspondence. Assume that a reference graph $\mathcal{G}_M = (\mathcal{V}^{\mathcal{G}_M}, \mathcal{E}^{\mathcal{G}_M}, \mathcal{U}^{\mathcal{G}_M}, \mathcal{B}^{\mathcal{G}_M}, \mathcal{F}^{\mathcal{G}_M})$ and an input graph $\mathcal{G}_I = (\mathcal{V}^{\mathcal{G}_I}, \mathcal{E}^{\mathcal{G}_I}, \mathcal{U}^{\mathcal{G}_I}, \mathcal{B}^{\mathcal{G}_I}, \mathcal{F}^{\mathcal{G}_I})$ are given.

Definition 4 (Correspondence). *If i-th node of $\mathcal{G}_M$ and the l-th node of $\mathcal{G}_I$ match by each other, then there exist a correspondence between them, and it is denoted by*

$$v_i^{\mathcal{G}_M} \leftrightarrow v_l^{\mathcal{G}_I} \text{ or } v_l^{\mathcal{G}_I} = Cor(v_i^{\mathcal{G}_M}) \text{ or } l = Cor(v_i^{\mathcal{G}_M}). \tag{5}$$

Definition 5 (Stochastic neighborhood). *The stochastic neighborhood of binary attribute vector $\mathbf{r}_{ij}^{\mathcal{G}_M}$ in the relation vector space $\mathcal{R}_l^{\mathcal{G}_I}$ under the assumption of $v_i^{\mathcal{G}_M} \leftrightarrow v_l^{\mathcal{G}_I}$ is defined by*

$$\begin{aligned} \mathcal{N}_{\mathcal{R}_l^{\mathcal{G}_I}}(\mathbf{r}_{ij}^{\mathcal{G}_M}) &= \{v_m^{\mathcal{G}_I} \,|\, ||\mathbf{r}_{lm}^{\mathcal{G}_I} - \mathbf{r}_{ij}^{\mathcal{G}_M}||_{prob} < \Delta\}, \\ &= \{v_m^{\mathcal{G}_I} \,|\, p(\mathbf{r}_{lm}^{\mathcal{G}_I} - \mathbf{r}_{ij}^{\mathcal{G}_M}) \times p(\mathbf{a}_m^{\mathcal{G}_I} - \mathbf{a}_j^{\mathcal{G}_M}) < \Delta\}. \end{aligned} \tag{6}$$

Definition 6 (Support flow). *The support flow from $v_j^{\mathcal{G}_M}$ to $v_i^{\mathcal{G}_M}$ under the assumption of $v_i^{\mathcal{G}_M} \leftrightarrow v_l^{\mathcal{G}_I}$ is defined by*

$$\mathcal{F}_{sup}(v_j^{\mathcal{G}_M} | v_i^{\mathcal{G}_M}, v_l^{\mathcal{G}_I}) = \begin{cases} \sum\limits_{\substack{v_m^{\mathcal{G}_I} \in \\ \mathcal{N}_{\mathcal{R}_l^{\mathcal{G}_I}}(\mathbf{r}_{ij}^{\mathcal{G}_M})}} \frac{p(\mathbf{r}_{lm}^{\mathcal{G}_I} - \mathbf{r}_{ij}^{\mathcal{G}_M}) \cdot \mathcal{M}_{cor}(v_j^{\mathcal{G}_M}, v_m^{\mathcal{G}_I})}{|\mathcal{N}_{\mathcal{R}_l^{\mathcal{G}_I}}(\mathbf{r}_{ij}^{\mathcal{G}_M})|} & \text{if } \mathcal{N}_{\mathcal{R}_l^{\mathcal{G}_I}}(\mathbf{r}_{ij}^{\mathcal{G}_M}) \neq \emptyset, \\ Max \;\; p(\mathbf{r}_{lm}^{\mathcal{G}_I} - \mathbf{r}_{ij}^{\mathcal{G}_M}) \cdot \mathcal{M}_{cor}(v_j^{\mathcal{G}_M}, v_m^{\mathcal{G}_I}) & \text{if } \mathcal{N}_{\mathcal{R}_l^{\mathcal{G}_I}}(\mathbf{r}_{ij}^{\mathcal{G}_M}) = \emptyset, \end{cases} \tag{7}$$

where $\mathcal{M}_{cor}(v_j^{\mathcal{G}_M}, v_m^{\mathcal{G}_I})$ is a correspondence measure between $v_j^{\mathcal{G}_M}$ and $v_m^{\mathcal{G}_I}$

The support flow in (7) represents how much a neighboring node supports the given node correspondence. Fig. 1 shows an example of the stochastic neighborhoods and support flows. Provided that i-th node of $\mathcal{G}_M$ corresponds to the l-th node of $\mathcal{G}_I$, $\mathcal{R}_l^{\mathcal{G}_I}$ must be similar to $\mathcal{R}_i^{\mathcal{G}_M}$. From the definition in (7), it is noted that as the similarity between two relation vector spaces of the corresponding nodes increases, the sum of the support flows from other nodes also increases. As a result, the sum of the support flows can be used as an indication for the correspondence between two nodes. Based on this observation, the *correspondence measure* is defined as the average of the support flows from other nodes.

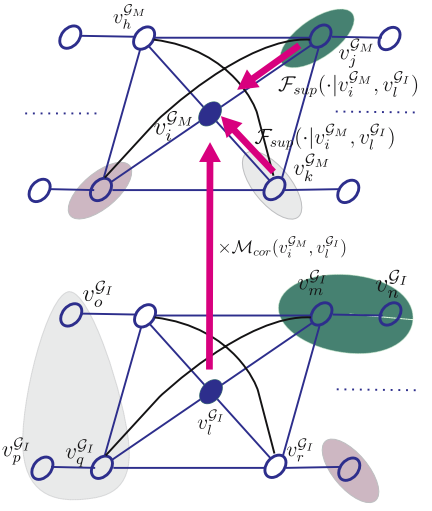

Fig. 1. Graphical representation of stochastic neighborhoods and support flows. Corresponding neighborhoods are represented by the colored regions, and the arrows show the support flows from neighbor nodes to the specific node in a reference graph.

Definition 7 (Correspondence measure). *The correspondence measure between* $v_i^{\mathcal{G}_M}$ *and* $v_l^{\mathcal{G}_I}$ *is defined by*

$$\mathcal{M}_{cor}(v_i^{\mathcal{G}_M}, v_l^{\mathcal{G}_I}) = \sum_{j=1, j\neq i}^{N^{\mathcal{G}_M}} \frac{\mathcal{F}_{sup}(v_j^{\mathcal{G}_M}|v_i^{\mathcal{G}_M}, v_l^{\mathcal{G}_I}) \cdot \mathcal{M}_{cor}(v_i^{\mathcal{G}_M}, v_l^{\mathcal{G}_I})}{(N^{\mathcal{G}_M} - 1)}. \tag{8}$$

Note that in order to embed the structural consistency, the actual correspondence measure is calculated iteratively through an updating process using (7) and (8) as follows.

1) *Initialize:* $k = 0$

$$\mathcal{M}_{cor}^{(0)}(v_i^{\mathcal{G}_M}, v_l^{\mathcal{G}_I}) = p(\mathbf{a}_l^{\mathcal{G}_I} - \mathbf{a}_i^{\mathcal{G}_M}). \tag{9}$$

2) *Updating process:* $k \geq 1$

$$\mathcal{F}_{sup}^{(k)}(v_j^{\mathcal{G}_M}|v_i^{\mathcal{G}_M}, v_l^{\mathcal{G}_I}) = \begin{cases} \sum\limits_{\substack{v_m^{\mathcal{G}_I} \in \\ \mathcal{N}_{\mathcal{R}_l^{\mathcal{G}_I}}(\mathbf{r}_{ij}^{\mathcal{G}_M})}} \frac{p(\mathbf{r}_{lm}^{\mathcal{G}_I} - \mathbf{r}_{ij}^{\mathcal{G}_M}) \cdot \mathcal{M}_{cor}^{(k-1)}(v_j^{\mathcal{G}_M}, v_m^{\mathcal{G}_I})}{|\mathcal{N}_{\mathcal{R}_l^{\mathcal{G}_I}}(\mathbf{r}_{ij}^{\mathcal{G}_M})|} & \text{if } \mathcal{N}_{\mathcal{R}_l^{\mathcal{G}_I}}(\mathbf{r}_{ij}^{\mathcal{G}_M}) \neq \emptyset, \\ Max \quad p(\mathbf{r}_{lm}^{\mathcal{G}_I} - \mathbf{r}_{ij}^{\mathcal{G}_M}) \cdot \mathcal{M}_{cor}^{(k-1)}(v_j^{\mathcal{G}_M}, v_m^{\mathcal{G}_I}) & \text{if } \mathcal{N}_{\mathcal{R}_l^{\mathcal{G}_I}}(\mathbf{r}_{ij}^{\mathcal{G}_M}) = \emptyset, \end{cases} \tag{10}$$

$$\mathcal{M}_{cor}^{(k)}(v_i^{\mathcal{G}_M}, v_l^{\mathcal{G}_I}) = \sum_{j=1, j\neq i}^{N^{\mathcal{G}_M}} \frac{\mathcal{F}_{sup}^{(k)}(v_j^{\mathcal{G}_M}|v_i^{\mathcal{G}_M}, v_l^{\mathcal{G}_I}) \cdot \mathcal{M}_{cor}^{(k-1)}(v_i^{\mathcal{G}_M}, v_l^{\mathcal{G}_I})}{(N^{\mathcal{G}_M} - 1)}. \tag{11}$$

3.2 Selection of Candidate Sub-graphs

In the combinatorial graph matching methods, constructing meaningful candidate subgraphs and ordering them are important issues. The proposed SARG matching algorithm extracts and sorts candidate subgraphs by measuring correspondence stochastically in the relation vector space.

Provided that i-th node of $\mathcal{G}_M$ corresponds to the l-th node of $\mathcal{G}_I$, the subspace of $\mathcal{R}_l^{\mathcal{G}_I}$ that is similar to $\mathcal{R}_i^{\mathcal{G}_M}$ can be constructed as

$$\hat{\mathcal{R}}_l{}^{\mathcal{G}_I} = \{\mathbf{r}_{lm}^{\mathcal{G}_I} | v_m^{\mathcal{G}_I} \in \mathcal{N}_{\mathcal{R}_l^{\mathcal{G}_I}}(\mathbf{r}_{ij}^{\mathcal{G}_M}), j = 1, \ldots, N^{\mathcal{G}_M}, j \neq i\}. \tag{12}$$

Then, a set of initial candidate subgraphs subject to $v_i^{\mathcal{G}_M} \leftrightarrow v_l^{\mathcal{G}_I}$ can be constructed by selecting one node in each neighbor $\mathcal{N}_{\mathcal{R}_l^{\mathcal{G}_I}}(\mathbf{r}_{ij}^{\mathcal{G}_M})$ as

$$\begin{aligned}\mathcal{A}(v_i^{\mathcal{G}_M}, v_l^{\mathcal{G}_I}) = \{(v_{k_1}^{\mathcal{G}_I}, \ldots, v_{k_{i-1}}^{\mathcal{G}_I}, v_l^{\mathcal{G}_I}, v_{k_{i+1}}^{\mathcal{G}_I} \ldots, v_{k_{N^{\mathcal{G}_M}}}^{\mathcal{G}_I})| \\ v_{k_j}^{\mathcal{G}_I} \in \mathcal{N}_{\mathcal{R}_l^{\mathcal{G}_I}}(\mathbf{r}_{ij}^{\mathcal{G}_M}),\ j = 1, 2, \ldots, N^{\mathcal{G}_M}, j \neq i\}.\end{aligned} \tag{13}$$

Once all $\mathcal{A}(v_i^{\mathcal{G}_M}, v_l^{\mathcal{G}_I})$, for $i = 1, \ldots, N^{\mathcal{G}_M}$, and $l = 1, \ldots, N^{\mathcal{G}_I}$, are obtained, the total initial candidate subgraphs are given by the union of all sets $\mathcal{A}(v_i^{\mathcal{G}_M}, v_l^{\mathcal{G}_I})$;

$$\mathcal{A}_T = \bigcup_{i,l} \mathcal{A}(v_i^{\mathcal{G}_M}, v_l^{\mathcal{G}_I}). \tag{14}$$

Now, meaningless initial subgraphs can be excluded out from (14) by ordering them according to the correspondence measure defined in (8), and the final candidate subgraphs with high priority are given by

$$\hat{\mathcal{A}}_T = \{\mathcal{A}(v_i^{\mathcal{G}_M}, v_l^{\mathcal{G}_I}) | \mathcal{M}_{cor}(v_i^{\mathcal{G}_M}, v_l^{\mathcal{G}_I}) \geq \alpha\}. \tag{15}$$

3.3 Missing Node Detection and Correction

Assume that a candidate subgraph $\mathcal{G}_C \in \hat{\mathcal{A}}_T$ and a reference graph $\mathcal{G}_M$ are given. Each node in $\mathcal{G}_C$ has one-to-one correspondence to a node with same index in $\mathcal{G}_M$, and this relation is denoted by $v_i^{\mathcal{G}_C} \leftrightarrow v_i^{\mathcal{G}_M}$ or $v_i^{\mathcal{G}_C} = Cor(v_i^{\mathcal{G}_M})$.

Then the missing nodes can be detected by constructing and analyzing the node loss vector given by,

$$\mathcal{L} = [l(1) \cdots l(N^{\mathcal{G}_M})]^T, \tag{16}$$

where $l(i)$ is the number of nodes that satisfies the inequality $p(\mathbf{r}_{ij}^{\mathcal{G}_C} - \mathbf{r}_{ij}^{\mathcal{G}_M}) < p_{thres}$. Each detected missing node has the correspondence to NULL node ($v_i^{\mathcal{G}_M} \leftrightarrow v_0$).

For example, after detecting missing nodes from one of the candidate subgraphs in Fig. 2 (a), $\mathcal{V}^{\mathcal{G}_M}$ is partitioned as two sets as $\mathcal{V}_C^{\mathcal{G}_M}$ and $\mathcal{V}_0^{\mathcal{G}_M} = \{v_1^{\mathcal{G}_M}, v_2^{\mathcal{G}_M}\}$ as shown in Fig. 2 (b), where dark circles connected by the arrow represent the corresponding node pair and the dotted circles mean the detected missing nodes. However, it is certain that $v_1^{\mathcal{G}_M}$ has the correspondence to one node in $\mathcal{G}_I$ instead

of the NULL node. Thus, we propose the additional procedure to correct wrong correspondence to the NULL node as $v_1^{\mathcal{G}_M}$.

Now, the node set of $\mathcal{G}_M$, $\mathcal{V}^{\mathcal{G}_M}$ can be partitioned into two sets as

$$\mathcal{V}^{\mathcal{G}_M} = \mathcal{V}_C^{\mathcal{G}_M} + \mathcal{V}_0^{\mathcal{G}_M}, \tag{17}$$

where all nodes in $\mathcal{V}_0^{\mathcal{G}_M}$ have the correspondence to the NULL node, v_0. In order to reduce the false detection and make as many correspondences as possible, we recompute the correspondence measures for the nodes in $\mathcal{V}_0^{\mathcal{G}_M}$ again by using

$$\mathcal{M}_{cor}(v_i^{\mathcal{G}_M}, v_l^{\mathcal{G}_I}) = \sum_{v_j^{\mathcal{G}_M} \in \mathcal{V}_C^{\mathcal{G}_M}, j \neq i} p(\mathbf{r}_{ij}^{\mathcal{G}_M} - \mathbf{r}_{lCor(v_j^{\mathcal{G}_M})}^{\mathcal{G}_M}) \tag{18}$$

$$\cdot p(\mathbf{a}_i^{\mathcal{G}_M} - \mathbf{a}_l^{\mathcal{G}_I}) \cdot p(\mathbf{a}_j^{\mathcal{G}_M} - \mathbf{a}_{Cor(v_j^{\mathcal{G}_M})}^{\mathcal{G}_I}), \text{ for } v_i^{\mathcal{G}_M} \in \mathcal{V}_0^{\mathcal{G}_M}.$$

Then, we determine the final correspondence for each node in $\mathcal{V}_0^{\mathcal{G}_M}$ by

$$v_i^{\mathcal{G}_M} \leftrightarrow \tag{19}$$

$$\begin{cases} v_l^{\mathcal{G}_I} & \text{if } v_l^{\mathcal{G}_I} = \arg\max_{v_k^{\mathcal{G}_I}} \mathcal{M}_{cor}(v_i^{\mathcal{G}_M}, v_k^{\mathcal{G}_I}) \text{ and } \mathcal{M}_{cor}(v_i^{\mathcal{G}_M}, v_l^{\mathcal{G}_I}) \geq \beta\ , \\ v_0 & \text{otherwise.} \end{cases}$$

Fig. 2 shows an example. By recomputing the correspondence measure based on the strong correspondences as described above, missing nodes are detected and the final correct correspondences can be obtained as in Fig. 2 (c).

3.4 Matching

Once all the candidate subgraphs are corrected, we can find the subgraph that best matches the model using the similarity measure given by

$$\mathcal{S}(\mathcal{G}_M, \mathcal{G}_C) = \sum_{i=1}^{N^{\mathcal{G}_M}} \mathcal{D}(\mathcal{R}_i) \cdot p(\mathbf{a}_i^{\mathcal{G}_1} - \mathbf{a}_i^{\mathcal{G}_2}) \cdot \omega_i \tag{20}$$

$$= \sum_{i=1}^{N^{\mathcal{G}_M}} \omega_i \cdot p(\mathbf{a}_i^{\mathcal{G}_1} - \mathbf{a}_i^{\mathcal{G}_2}) \cdot [\prod_{j=1, j \neq i}^{N^{\mathcal{G}_M}} p(\mathbf{r}_{ij}^{\mathcal{G}_C} - \mathbf{r}_{ij}^{\mathcal{G}_M}) \cdot \gamma_{ij}],$$

where $\mathcal{D}(\mathcal{R}_i)$ is a function to measure the difference between the relation vector space $\mathcal{R}_i^{\mathcal{G}_1}$ of $\mathcal{G}_1$ and $\mathcal{R}_i^{\mathcal{G}_2}$ of $\mathcal{G}_2$, and ω_i and γ_{ij} are the weighting for the i-th node and the binary relation between node v_i and v_j, respectively.

Then, the best matched subgraph is selected as follows.

$$\text{Matched Graph} = \arg \max_{\mathcal{G}_C \in \hat{\mathcal{A}}_T} \mathcal{S}(\mathcal{G}_M, \mathcal{G}_C). \tag{21}$$

4 Computational Complexity

In this section, the computational complexity of the proposed SARG matching algorithm is analyzed. The computation of the proposed SARG matching

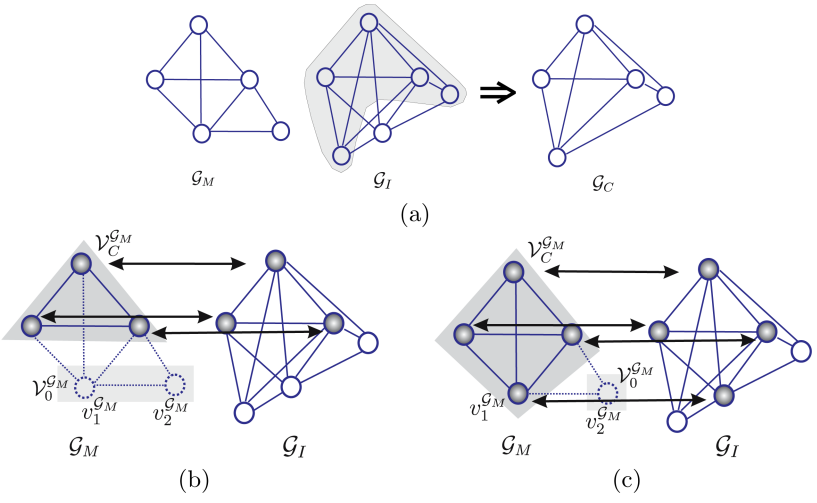

Fig. 2. Missing node detection and correcting process. (a) One of the candidate subgraphs are selected and (b) missing nodes are detected by analyzing the node loss vector, where the corresponding node pair is connected by the arrow and the dotted circles mean missing nodes. By recomputing the correspondence measure based on the strong correspondence, (c) the final correspondences are obtained.

algorithm consists of two parts: (1) Selecting the candidate subgraphs and (2) identifying missing node and matching. In the first stage, the stochastic neighborhoods should be constructed before computing the correspondence measures, and it needs iterative calculation. Then the candidate subgraphs are extracted and sorted according to the correspondence measure. Assume that a reference graph $\mathcal{G}_M$ with $N^{\mathcal{G}_M}$ nodes and an input graph $\mathcal{G}_I$ with $N^{\mathcal{G}_I}$ nodes are given, and K iterations are performed to update the correspondence measure. Construction of the stochastic neighborhoods for all node pairs is proportional to the square of the total possible number of correspondences, $N^{\mathcal{G}_M}N^{\mathcal{G}_I}$, that is approximately equal to $O({N^{\mathcal{G}_M}}^2{N^{\mathcal{G}_I}}^2)$. And since, at each iteration step, $N^{\mathcal{G}_M}$ support flows in (7) should be evaluated per each correspondence pair, the cost of computing the correspondence measures becomes $O(K \cdot {N^{\mathcal{G}_M}}^2 N^{\mathcal{G}_I})$. And, the total cost for extracting and sorting the candidate subgraphs is proportional to the total number of correspondences,i.e., $O(N^{\mathcal{G}_M}N^{\mathcal{G}_I})$. In the result, the computational complexity of the first part is

$$\begin{aligned} T_1 &= T_{neighbor} + T_{correspondence} + T_{candidate} \\ &= O({N^{\mathcal{G}_M}}^2{N^{\mathcal{G}_I}}^2) + O(K \cdot {N^{\mathcal{G}_M}}^2 N^{\mathcal{G}_I}) + O(N^{\mathcal{G}_M}N^{\mathcal{G}_I}) = O({N^{\mathcal{G}_M}}^2{N^{\mathcal{G}_I}}^2). \end{aligned}$$

In the second stage, for each candidate subgraph, the missing node detection and correction processes are carried out first, and then matching is done by evaluating the similarity between two ARGs. Therefore the computational complexity of the second part is proportional to the product of the number of candidate subgraphs and the computational cost required for one candidate subgraph. Note that for a given candidate subgraph, the costs for detecting missing nodes

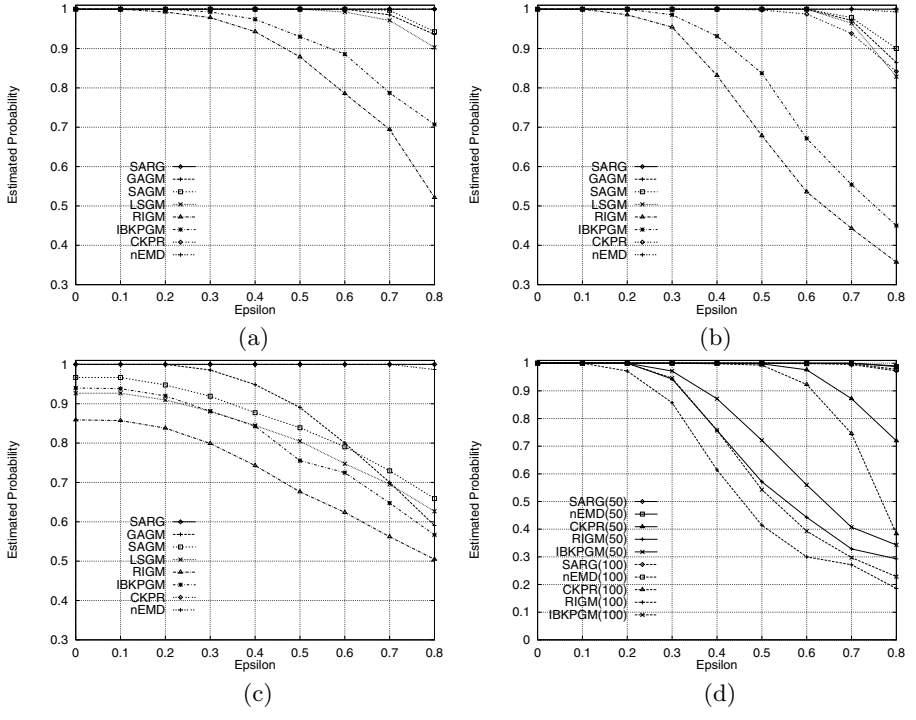

Fig. 3. Graph matching results under the independent noise condition with $(N^{\mathcal{G}_M}, N^{\mathcal{G}_I}, N_U, N_B)$ = (a) $(10, 10, 3, 3)$, (b) $(30, 30, 3, 3)$, (c) $(15, 5, 5, 5)$, and (d) $(N^{\mathcal{G}_M}, N^{\mathcal{G}_I}, 3, 3)$, where $N^{\mathcal{G}_M} = N^{\mathcal{G}_I} = 50, 100$

and matching are all $O({N^{\mathcal{G}_M}}^2)$, while that of correcting process is $O(N^{\mathcal{G}_M} N^{\mathcal{G}_I})$, since it only requires the recomputation of $N^{\mathcal{G}_M} N^{\mathcal{G}_I}$ correspondence measures and the selection of the node having the maximum correspondence measure. And the number of candidate subgraphs denoted by $n(\hat{\mathcal{A}}_T)$ is less than the product of the numbers of nodes in two graphs, that is, $n(\hat{\mathcal{A}}_T) \leq \zeta \cdot N^{\mathcal{G}_M} N^{\mathcal{G}_I}$, where ζ is a constant that typically varies between 0.01 and 0.2. Thus the computational cost for the latter part is

$$\begin{aligned} T_2 &= T_{detection} + T_{correction} + T_{matchnig} \\ &= O(\zeta \cdot {N^{\mathcal{G}_M}}^3 N^{\mathcal{G}_I}) + O(\zeta \cdot {N^{\mathcal{G}_M}}^2 {N^{\mathcal{G}_I}}^2) + O(\zeta \cdot {N^{\mathcal{G}_M}}^3 N^{\mathcal{G}_I}) = O({N^{\mathcal{G}_M}}^3 N^{\mathcal{G}_I}). \end{aligned}$$

In summary, the total computational cost required for the proposed SARG matching algorithm is

$$T_{total} = T_1 + T_2 = \max[O({N^{\mathcal{G}_M}}^2 {N^{\mathcal{G}_I}}^2), O({N^{\mathcal{G}_M}}^3 N^{\mathcal{G}_I})]. \tag{22}$$

5 Experimental Results

To evaluate the matching performance of the proposed algorithm, we have tested it on synthetic ARGs as in [16]. Synthetic ARGs were generated by the following

Table 1. The computational complexity of each graph matching algorithm in terms of the processing time for one pair graph matching

$(N^{\mathcal{G}_M}, N^{\mathcal{G}_M}, N_U, N_B)$	$(15,5,5,5)$	$(10,10,3,3)$	$(30,30,3,3)$	$(50,50,3,3)$	$(100,100,3,3)$
nEMD	0.007s	0.009s	0.990s	10.24s	262.1s
CKPR	0.004s	0.005s	0.249s	4.34s	63.3s
SARG	0.001s	0.002s	0.145s	1.21s	14.9s

procedures: First, given fixed $(N^{\mathcal{G}_M}, N^{\mathcal{G}_I}, N_U, N_B)$, a reference graph $\mathcal{G}_M$ was randomly generated, in which all attributes had a random number between 0 and 1. Then, an input graph was constructed using randomly generated permutation matrix $\mathbf{P}$. Next, independent noise matrices $\mathbf{N}_i^U$'s and $\mathbf{N}_i^B$'s were obtained by multiplying a uniformly distributed random variable on the interval $[-\frac{1}{2}, \frac{1}{2}]$ by the noise power $\epsilon \in [0, 1.0]$.

5.1 Independent Noise

In order to generate the independent noise, we fixed all noise correlation matrices such that $\forall \mathbf{C}_i^U = \mathbf{I}$ and $\forall \mathbf{C}_{lm}^B = \mathbf{\Delta}_{lm}$. For the benchmarks, we have selected GAGM [8], SAGM [13], LSGM [14], RIGM [15] and IBKPGM [16] algorithms among various ARG matching algorithms in [16], since they showed better matching performance than others. Moreover, the performance was compared to CKPR [5] and nEMD [19]. The estimated probability of correct node-to-node matching was evaluated as a function of the noise magnitude ϵ. To reflect the graph matching performance in term of probability, for a given value of ϵ, we have done 300 trials for each graph matching algorithm.

Fig. 3 (a) and (b) summarize the matching results for the full graph matching when $(N^{\mathcal{G}_M}, N^{\mathcal{G}_I}, N_U, N_B) = (10,10,3,3)$ and $(30,30,3,3)$, respectively. It is noted that SARG outperforms other graph matching algorithms especially for large values of ϵ. And, generally, since nEMD, CKPR and SARG consider the NULL node explicitly, they showed superior subgraph matching performances than the others. However, as the noise power increased, the matching rate of nEMD decreased, while those of CKPR and SARG remained almost constant. In Fig. 3 (d), the performances of some algorithms are shown for $N^{\mathcal{G}_M} = N^{\mathcal{G}_I} = 50$ and 100. Due to the limitations on the computational cost and the memory, some algorithms were excluded for comparison. Actually, as the number of nodes increased, the graph matching performance became severely degraded. However, nEMD and SARG were very robust to the increase of the number of nodes, and SARG performed best.

5.2 Complexity Analysis

We have analyzed and compared the computational complexity of the proposed algorithm with those of nEMD and CKPR in terms of the processing time for one pair graph matching. We measured the processing time for one pair graph

matching by varying the number of nodes and the number of attributes of graphs, and calculated the average processing time after 500 trials per each condition. The results are presented in Table 5.2, where the processing time was evaluated in seconds. It is noted that the proposed algorithm is much faster than the others, especially for the graphs with a large number of nodes.

6 Conclusion

In order to match ARGs by subgraph isomorphism efficiently, in this paper, we proposed a novel stochastic attributed relational graph (SARG) matching technique using the stochastic analysis in the relation vector space, which embeds the global structure as well as the local structure centered at a specific node. The new concepts related to the correspondence, such as the stochastic neighborhood, the support flow, the correspondence measure and the similarity were defined in terms of the probability and the geometrical distribution of the attribute vectors in the relation vector space. The proposed SARG matching algorithm consists of 2 step procedures. In the first stage, a finite number of subgraphs were extracted from the test graph and ordered according to the correspondence measure to the model graph. Then, missing nodes for each candidate subgraphs were detected and the correspondences are reestablished by eliminating the effects of them. Finally, the refined subgraphs are matched to the model graph by measuring the similarity between them.

Experimental results on the synthetic ARGs demonstrated the robustness and efficiency of the proposed SARG matching algorithm. And it was also verified empirically that the proposed SARG matching algorithm was much faster than conventional graph-based algorithms.

Acknowledgement

This work has been supported in part by the ITRC (Information Technology Research Center) support program of Korean government and IIRC (Image Information Research Center) by Agency of Defense Development, Korea.

References

1. A. K. C. Wong and M. You, "Entropy and distance of random graphs with apllication to structural pattern recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 7, no. 5, September 1985.
2. A. Sanfeliu, R. Alquézar, J. Andrade, J. Climent, F. Serratosa and J. Vergés, "Graph-based representations and techniques for image processing and image analysis," *Pattern Recognition,* vol. 35, pp. 639-650, 2002.
3. F. Serratosa, R. Alquézar and A. Sanfeliu, "Function-described graphs for modelling objects represented by sets of attributes graphs," *Pattern Recognition,* vol. 36, pp. 781-798, 2003.

4. S. Z. Li, "Matching : invariant to translations, rotations and scale changes," *Pattern Recognition,* vol. 25, pp. 583-594, 1992.
5. W. J. Christmas, J. Kittler and M. Petrou, "Structural matching in computer vision using probabilistic relaxation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 8, pp. 749-764, 1995.
6. B. T. Messmer and H. Bunke, "A new algorithm for error-tolerant subgraph isomorphism detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 5, pp. 493-503, 1998.
7. B. T. Messmer and H. Bunke, "A decision tree approach to graph and subgraph isomorphism detection," *Pattern Recognition,* vol. 32, pp. 1979-1998, 1999.
8. S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 4, pp. 377-388, April 1996.
9. W. H. Tsai and K. S. Fu, "Subgraph error-correcting isomorphisms for syntactic pattern recognition," *IEEE Trans. Systems Man and Cybernatics,* vol. 13, no. 1, pp. 48-62, January/February 1983.
10. Y. El-Sonbaty and M. A. Ismail, "A new algorithm for subgraph optimal isomorphism," *Pattern Recognition,* vol. 31, no. 2, pp. 205-218, 1998.
11. L. Herault, R. Horaud, F. Veillon and J. J. Niez, "Symbolic image matching by simulated annealing," *Proc. British Machine Vision Conference,* Oxford, pp. 319-324, 1990.
12. M. Krcmar and A. Dhawan, "Application of genetic algorithms in graph matching," *Proc. Int'l Conf. Neural Networks,* vol. 6, pp. 3872-3876, 1994.
13. B. J. van Wyk and M. A. van Wyk, "The spherical approximation graph matching algorithm," *Proc. Int'l Workshop on Multidiscilinary Design Optimization,* pp. 280-288, August 2000.
14. B. J. van Wyk and J. Clark, "An algorithm for approximate least-squares attributed graph matching," *Problems in Applied Mathematics and Computational Intelligence,* pp. 67-72, 2000.
15. M. A. van Wyk, T. S. Durrani and B. J. van Wyk, "A RKHS interpolator-based graph matching algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 7, pp. 988-995, 2002.
16. B. J. van Wyk and M. A. van Wyk, "Kronecker product graph matching," *Pattern Recognition,* vol. 36, pp. 2019-2030, 2003.
17. R. Nevatia and K. R. Babu, "Line extraction and description," *Computer Graphics and Image Processing,* vol.13, no.1, pp.250-269, July 1980.
18. B. G. Park, K. M. Lee, S. U. Lee, and J.H. Lee, "Recognition of partially occluded objects using probabilistic ARG-based matching," *Computer Vision and Image Understanding,* vol. 90, no. 3, pp. 217-241, June 2003.
19. D. H. Kim, I. D. Yun, and S. U. Lee, "A new attributed relational graph matching algorithm using the nested structure of earth mover's distance," *Proceedings of IEEE International Conference on Pattern Recognition,* Cambridge, UK, pp. 48-51, August 2004.