

Estimating the Joint Probability Distribution of Random Vertices and Arcs by Means of Second-Order Random Graphs

Francesc Serratosa¹, René Alquézar², and Alberto Sanfeliu³

¹ Universitat Rovira i Virgili
Dept. d'Enginyeria Informàtica i Matemàtiques, Spain
Francesc.Serratosa@etse.urv.es
<http://www.etse.urv.es/~fserrato>

² Universitat Politècnica de Catalunya
Dept. de Llenguatges i Sistemes Informàtics, Spain
alquezar@lsi.upc.es

³ Universitat Politècnica de Catalunya
Institut de Robòtica i Informàtica Industrial, Spain
sanfeliu@iri.upc.es

Abstract. We review the approaches that model a set of Attributed Graphs (AGs) by extending the definition of AGs to include probabilistic information. As a main result, we present a quite general formulation for estimating the joint probability distribution of the random elements of a set of AGs, in which some degree of probabilistic independence between random elements is assumed, by considering only 2nd-order joint probabilities and marginal ones. We show that the two previously proposed approaches based on the random-graph representation (First-Order Random Graphs (FORGs) and Function-Described Graphs (FDGs)) can be seen as two different approximations of the general formulation presented. From this new representation, it is easy to derive that whereas FORGs contain some more semantic (partial) 2nd-order information, FDGs contain more structural 2nd-order information of the whole set of AGs. Most importantly, the presented formulation opens the door to the development of new and more powerful probabilistic representations of sets of AGs based on the 2nd-order random graph concept.

1 Introduction

There are two major problems that practical applications using structural pattern recognition are confronted with. The first problem is the computational complexity of comparing two AGs. The time required by any of the optimal algorithms may in the worst case become exponential in the size of the graphs. The approximate algorithms, on the other hand, have only polynomial time complexity, but do not guarantee to find

the optimal solution. For some applications, this may not be acceptable. The second problem is the fact that there is more than one model AG that must be matched with an input AG, what means that the conventional error-tolerant graph matching algorithms must be applied to each model-input pair sequentially. As a consequence, the total computational cost is linearly dependent on the size of the database of model graphs. For applications dealing with large databases, this may be prohibitive.

To alleviate these problems, some attempts have been made to try to reduce the computational time of matching the unknown input patterns to the whole set of models from the database. Assuming that the AGs that represent a cluster or class are not completely dissimilar in the database, only one structural model is defined from the AGs that represent the cluster, and thus, only one comparison is needed for each cluster [3,6,7,8].

In this paper, we review the approaches that model a set of AGs by extending the definition of graphs to include probabilistic information [3,4,6,8]. The resulting model, called *random graph* (RG) representation, is described in the most general case through a joint probability space of random variables ranging over pattern primitives (graph vertices) and relations (graph arcs). It is the union of the AGs in the cluster, according to some synthesis process, together with its associated probability distribution. In this manner, a structural pattern can be explicitly represented in the form of an AG and an ensemble of such representations can be considered as a set of outcomes of the RG.

In the following section, we introduce the formal definitions used throughout the paper. In section 3, we recall First-Order Random Graphs (FORGs) [6,8] and Function-Described Graphs (FDGs) [1,3,5,9], which are the two main approximations of the general RG concept proposed in the literature. The approach presented in the paper by Sengupta *et al.* [4] can be regarded as similar to the FORG approach. In section 4, we give a quite general formulation for estimating the joint probability of the random elements in a RG synthesised from a set of AGs. In sections 5 and 6, we show respectively that the FORG and FDG approaches can be seen as different simplifications of the general formulation given in section 4. Finally, in the last section we provide some discussion about our contribution and its future implications.

2 Formal Definitions of Random-Graph Representation

Definition 1. Let Δ_v and Δ_e denote the domains of possible values for attributed vertices and arcs, respectively. These domains are assumed to include a special value Φ that represents a *null value* of a vertex or arc. An AG G over (Δ_v, Δ_e) is defined to be a four-tuple $G = (\Sigma_v, \Sigma_e, \gamma_v, \gamma_e)$, where $\Sigma_v = \{v_k \mid k = 1, \dots, n\}$ is a set of vertices (or nodes), $\Sigma_e = \{e_{ij} \mid i, j \in \{1, \dots, n\}, i \neq j\}$ is a set of arcs (or edges), and the mappings $\gamma_v : \Sigma_v \rightarrow \Delta_v$ and $\gamma_e : \Sigma_e \rightarrow \Delta_e$ assign attribute values to vertices and arcs, respectively.

Definition 2. A *complete AG* is an AG with a complete graph structure (Σ_v, Σ_e) , but possibly including null elements. An AG G of order n can be *extended* to form a

complete AG G' of order k , $k \geq n$, by adding vertices and arcs with null attribute values Φ . We call G' the k -extension of G .

Definition 3. Let Ω_v and Ω_e be two sets of random variables with values in Δ_v (random vertices) and in Δ_e (random arcs), respectively. A *random-graph structure* R over (Δ_v, Δ_e) is defined to be a tuple $(\Sigma_v, \Sigma_e, \gamma_v, \gamma_e, P)$, where $\Sigma_v = \{\omega_k \mid k = 1, \dots, n\}$ is a set of vertices, $\Sigma_e = \{\varepsilon_{ij} \mid i, j \in \{1, \dots, n\}, i \neq j\}$ is a set of arcs, the mapping $\gamma_v : \Sigma_v \rightarrow \Omega_v$ associates each vertex $\omega_k \in \Sigma_v$ with a random variable $\alpha_k = \gamma_v(\omega_k)$ with values in Δ_v , and $\gamma_e : \Sigma_e \rightarrow \Omega_e$ associates each arc $\varepsilon_{ij} \in \Sigma_e$ with a random variable $\beta_k = \gamma_e(\varepsilon_{ij})$ with values in Δ_e . And, finally, P is a joint probability distribution $P(\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_m)$ of all the random vertices $\{\alpha_i \mid \alpha_i = \gamma_v(\omega_i), 1 \leq i \leq n\}$ and random arcs $\{\beta_j \mid \beta_j = \gamma_e(\varepsilon_{kl}), 1 \leq j \leq m\}$.

Definition 4. A *complete RG* is a RG with a complete graph structure (Σ_v, Σ_e) , but possibly including null random elements (its probability of instantiation to the null value is one, i.e. $\Pr(\alpha = \Phi) = 1$ or $\Pr(\beta = \Phi) = 1$). A RG R of order n can be *extended* to form a complete RG R' of order k , $k \geq n$, by adding null random vertices and null random arcs. We call R' the k -extension of R . Note that both R' and R represent the same model.

Definition 5. Any AG obtained by instantiating all random vertices and random arcs of a RG in a way that satisfies all the structural relations is called an *outcome graph* of the RG. Hence, a RG represents the set of all possible AGs that can be outcome graphs of it, according to an associated probability distribution.

Definition 6. For each outcome graph G of a RG R , the joint probability of random vertices and arcs is defined over an instantiation that produces G , and such instantiation is associated with a structural isomorphism $\mu : G' \rightarrow R$, where G' is the extension of G to the order of R . Let G be oriented with respect to R by the structurally coherent isomorphism μ ; for each vertex ω_i in R , let $\mathbf{a}_i = \gamma_v(\mu^{-1}(\omega_i))$ be the corresponding attribute value in G' , and similarly, for each arc ε_{kl} in R (associated with random variable β_j) let $\mathbf{b}_j = \gamma_e(\mu^{-1}(\varepsilon_{kl}))$ be the corresponding attribute value in G' . Then the *probability of G according to* (or given by) *the orientation* μ , denoted by $P_R(G|\mu)$, is defined as

$$P_R(G|\mu) = \Pr\left(\bigwedge_{i=1}^n (\alpha_i = \mathbf{a}_i) \wedge \bigwedge_{j=1}^m (\beta_j = \mathbf{b}_j)\right) = p(\mathbf{a}_1, \dots, \mathbf{a}_n, \mathbf{b}_1, \dots, \mathbf{b}_m) \quad (1)$$

3 Approximating Probability Distributions in the Literature

When estimating the probability distribution of the structural patterns from an ensemble, it is impractical to consider the high order probability distribution $P(\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_m)$ where all components and their relations in the structural patterns are taken jointly (eq. 1). For this reason, some other more practical approaches have been presented that propose different approximations [1,3,4,5,6]. All of them take into account in some manner the incidence relations between attributed vertices and arcs, i.e. assume some sort of dependence of an arc on its connecting vertices. Also, a common ordering (or labeling) scheme is needed that relates vertices and arcs of all the involved AGs, which is obtained through an optimal graph mapping process called synthesis of the random graph representation. In the following sections, we comment the two main such approaches, FORGs and FDGs.

3.1 First-Order Random Graphs (FORGs)

Wong and You [6] proposed the First-Order Random Graphs (FORGs), in which strong simplifications are made so that RGs can be used in practice. They introduced three suppositions about the probabilistic independence between vertices and arcs:

- 1) The random vertices are mutually independent;
- 2) The random arcs are independent given values for the random vertices;
- 3) The arcs are independent of the vertices except for the vertices that they connect.

Definition 7 A FORG R is a RG that satisfies the assumptions 1, 2, 3 shown above.

Based on these assumptions, for a FORG R , the probability $P_R(G|\mu)$ becomes

$$P_R(G|\mu) = \prod_{i=1}^n p_i(\mathbf{a}_i) \prod_{j=1}^m q_j(\mathbf{b}_j \mid \mathbf{a}_{j1}, \mathbf{a}_{j2}) \tag{2}$$

where $p_i(\mathbf{a}) \triangleq \Pr(\alpha_i = \mathbf{a})$, $1 \leq i \leq n$, are the marginal probability density functions for vertices and $q_j(\mathbf{b} \mid \mathbf{a}_{j1}, \mathbf{a}_{j2}) \triangleq \Pr(\beta_j = \mathbf{b} \mid \alpha_{j1} = \mathbf{a}_{j1}, \alpha_{j2} = \mathbf{a}_{j2})$, $1 \leq j \leq m$, are the conditional probability functions for the arcs, where α_{j1}, α_{j2} refer to the random vertices for the endpoints of the random arc β_j .

The storage space of FORGs is $O(nN + mMN^2)$ where N and M are the number of elements of the domains Δ_v and Δ_e .

3.2 Function-Described Graphs (FDGs)

The FORG approach, although simplifies the representation considerably, continues to be difficult to apply in real problems where there is a large number of vertices in the AGs and their attributes have an extensive domain. The main cause of this problem is the dependence of the arc attributes with respect to the attributes of the

vertices that the arc connects (assumption 3). Although this supposition is useful to constrain the generalisation of the given set of AGs, it needs a huge amount of data to estimate the probability density functions and bears a high computational cost. On the other hand, an important drawback of FORGs, which is due to the probability independence assumptions 1 and 2, is that the structural information in a sample of AGs is not well preserved in the FORG synthesised from them. This is, an FORG represents an over-generalised prototype that may cover graph structures quite different from those in the sample.

With the aim of offering a more practical approach, Serratosa *et al.* [1,3,5,9] proposed the Function-Described Graphs (FDGs), which lead to another approximation of the joint probability P of the random elements. On one hand, some independence assumptions (a) are considered, but on the other hand, some useful 2nd-order functions (b) are included to constrain the generalisation of the structure.

(a) Independence assumptions in the FDGs

- 1) The attributes in the vertices are independent of the other vertices and of the arcs.
- 2) The attributes in the arcs are independent of the other arcs and also of the vertices. However, it is mandatory that all non-null arcs be linked to a non-null vertex at each extreme in every AG covered by an FDG. In other words, any outcome AG of the FDG has to be structurally consistent.

(b) 2nd-order functions in the FDGs

In order to tackle the problem of the over-generalisation of the sample, the *antagonism*, *occurrence* and *existence* relations are introduced in FDGs, which apply to pairs of vertices or arcs. In this way, random vertices and arcs are not assumed to be mutually independent, at least with regards to the structural information, since the above relations represent a *qualitative information* of the 2nd-order joint probability functions of a pair of vertices or arcs.

To understand these 2nd-order relations it is convenient to split the domain of the joint probabilities in four regions (see figure 1.a). The first one is composed by the points that belong to the Cartesian product of the sets of actual attributes of the two elements, corresponding to the cases where both elements are defined in the initial non-extended AG and therefore their value is not null. The second and third regions are both straight lines in which only one of the elements has the null value. This covers the cases when one of the two elements does not belong to the initial AG and has been added in the extending process. Finally, the fourth region is the single point where both elements are null, which includes the cases when none of them appear in the initial AG.

The 2nd-order relations on the vertices are defined as follows (the 2nd-order relations on the arcs are defined in a similar way [3,5]):

Antagonism relations: Two vertices of the FDG are *antagonistic* if the probabilities in the first region are all zero,

$$A_{\omega}(\omega_i, \omega_j) = \begin{cases} 1 & \text{if } \Pr(\alpha_i \neq \Phi \wedge \alpha_j \neq \Phi) = 0 \\ 0 & \text{if } \Pr(\alpha_i \neq \Phi \wedge \alpha_j \neq \Phi) > 0 \end{cases} \quad (3)$$

which means that, although these vertices are included in the prototype as different elementary parts of the covered patterns, they have never taken place together in any AG of the reference set used to synthesise the FDG. Figure 1.b shows the joint probabilities of the vertices ω_i and ω_j defined as antagonistic.

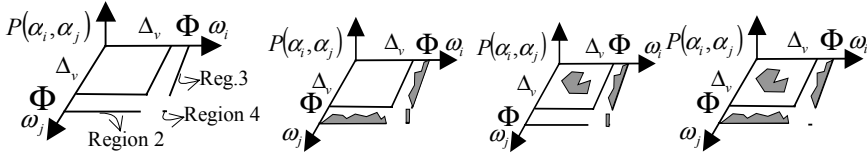


Fig. 1. (a) Split of the joint domain of two random vertices in four regions. 2nd-order density function of (b) two antagonistic, (c) occurrent and (d) existent vertices

Occurrence relations: There is an *occurrence relation* from ω_i to ω_j if the joint probability function equals zero in the second region,

$$O_{\omega}(\omega_i, \omega_j) = \begin{cases} 1 & \text{if } \Pr(\alpha_i \neq \Phi \wedge \alpha_j = \Phi) = 0 \\ 0 & \text{if } \Pr(\alpha_i \neq \Phi \wedge \alpha_j = \Phi) > 0 \end{cases} \quad (4)$$

That is, it is possible to assure that if ω_i does appear in any AG of the reference set then ω_j must appear too. The case of the third region is analogous to the second one with the only difference of swapping the elements. Figure 1.c shows the joint probabilities of vertices ω_i and ω_j , with an occurrence from ω_i to ω_j .

Existence relations: Finally, there is an *existence relation* between two vertices if the joint probability function equals zero in the fourth region,

$$E_{\omega}(\omega_i, \omega_j) = \begin{cases} 1 & \text{if } \Pr(\alpha_i = \Phi \wedge \alpha_j = \Phi) = 0 \\ 0 & \text{if } \Pr(\alpha_i = \Phi \wedge \alpha_j = \Phi) > 0 \end{cases} \quad (5)$$

that is, all the objects in the class described by the FDG have at least one of the two elements. Figure 1.d shows the joint probabilities of two vertices ω_i and ω_j satisfying an existence relation.

Definition 8 A *Function-Described Graph F* is a RG that satisfies the assumptions 1 and 2 shown above and contains the information of the 2nd-order relations of *antagonism*, *occurrence* and *existence* between pairs of vertices or arcs.

Based on these assumptions, for an FDG F , the probability $P_r(G|\mu)$ becomes

$$P_F(F|\mu) = \prod_{i=1}^n p_i(\mathbf{a}_i) \prod_{j=1}^m q_j(\mathbf{b}_j) \quad (6)$$

where $p_i(\mathbf{a})$ is defined as in FORGs and $q_j(\mathbf{b}) \triangleq \Pr(\beta_j = \mathbf{b} \mid \alpha_{j1} \neq \Phi, \alpha_{j2} \neq \Phi)$. Note that, due to the structural consistency requirements, there is no need to store the conditional probabilities $\Pr(\beta_j = \mathbf{b} \mid \alpha_{j1} = \Phi \vee \alpha_{j2} = \Phi)$ in the FDGs since, by definition $\Pr(\beta_j = \Phi \mid \alpha_{j1} = \Phi \vee \alpha_{j2} = \Phi) = 1$.

Moreover, the isomorphism μ not only has to be structurally coherent but also has to fulfil the 2nd-order constraints shown in (7). The basic idea of these constraints is the satisfaction by an AG to be matched of the antagonism, occurrence and existence relations inferred from the set of AGs used to synthesise the FDG. However, the 2nd-order relations caused by FDG null vertices should not be taken into account, since they are artificially introduced in the extension of the FDG (see [1] for more details).

$$\begin{aligned} \overbrace{(A_\omega(\omega_i, \omega_j) = 1 \wedge p_i(\Phi) \neq 1 \wedge p_j(\Phi) \neq 1)}^{\text{FDG}} &\Rightarrow \overbrace{(\mathbf{a}_i = \Phi \vee \mathbf{a}_j = \Phi)}^{\text{AG}} \\ (O_\omega(\omega_i, \omega_j) = 1 \wedge p_i(\Phi) \neq 1 \wedge p_j(\Phi) \neq 1) &\Rightarrow (\mathbf{a}_i = \Phi \vee \mathbf{a}_j \neq \Phi) \\ (E_\omega(\omega_i, \omega_j) = 1 \wedge p_i(\Phi) \neq 1 \wedge p_j(\Phi) \neq 1) &\Rightarrow (\mathbf{a}_i \neq \Phi \vee \mathbf{a}_j \neq \Phi) \end{aligned} \quad (7)$$

The storage space of FDGs is $O(nN + mM + n^2 + m^2)$ where N and M are the number of elements of the domains Δ_v and Δ_e , respectively.

4 Second-Order Random-Graph Representation

We show next that the joint probability of an instantiation of the random elements in a RG can be approximated as follows:

$$p(\mathbf{d}_1, \dots, \mathbf{d}_s) \approx \prod_{i=1}^s p_i(\mathbf{d}_i) \prod_{i=1}^{s-1} \prod_{j=i+1}^s r_{ij}(\mathbf{d}_i, \mathbf{d}_j) \quad (8)$$

where $p_i(\mathbf{d}_i)$ are the marginal probabilities of the random elements γ_i , (vertices or arcs) and r_{ij} are the Peleg compatibility coefficients [2] that take into account both the marginal and 2nd-order joint probabilities,

$$r_{ij}(\mathbf{d}_i, \mathbf{d}_j) = \frac{\Pr(\gamma_i = \mathbf{d}_i \wedge \gamma_j = \mathbf{d}_j)}{p_i(\mathbf{d}_i) p_j(\mathbf{d}_j)} \quad (9)$$

The Peleg coefficient, with a non-negative range, is related to the “degree” of dependence between two random variables. If they are independent, the joint probability is defined as the product of the marginal ones, thus, $r_{ij} = 1$ (or a value close to 1 if the probability functions are estimated). If one of the marginal probabilities is null, the joint probability is also null. In this case, the indecisiveness $0/0$ is solved as 1, since this do not affect the global joint probability, which is null.

Eq. (8) is obtained by assuming independence in the conditional probabilities (section 4.1) and rearranging the joint probability expression with the Bayes rule (section 4.2)

4.1 Conditional Probabilities

The conditional density probability $p(\gamma_i / (\gamma_{i+1}, \dots, \gamma_s))$ of a random element γ_i is used to compute the joint density probability $p(\gamma_1, \dots, \gamma_s)$. Applying the Bayes rule to the conditional probability, the following expression holds,

$$p(\gamma_i / (\gamma_{i+1}, \dots, \gamma_s)) = \frac{p(\gamma_i) \cdot p((\gamma_{i+1}, \dots, \gamma_s) / \gamma_i)}{p(\gamma_{i+1}, \dots, \gamma_s)} \quad (10)$$

Due to the fact that this n -order probability can not be stored in practice, we have to suppose at this point that the conditioning random variables γ_{i+1} to γ_n are independent to each other. In that case, an estimate is given by

$$p(\gamma_i / (\gamma_{i+1}, \dots, \gamma_s)) = p(\gamma_i) \cdot \prod_{j=i+1}^s \frac{p(\gamma_j / \gamma_i)}{p(\gamma_j)} = p(\gamma_i) \cdot \prod_{j=i+1}^s \frac{p(\gamma_j, \gamma_i)}{p(\gamma_j) \cdot p(\gamma_i)} \quad (11)$$

Thus, if we use the Peleg compatibility coefficients then the conditional probability is,

$$\text{prob}(\gamma_i = \mathbf{d}_i / (\gamma_{i+1} = \mathbf{d}_{i+1}, \dots, \gamma_s = \mathbf{d}_s)) = p_i(\mathbf{d}_i) \cdot \prod_{j=i+1}^s r_{ij}(\mathbf{d}_i, \mathbf{d}_j) \quad (12)$$

4.2 Joint Probability

Using the Bayes theorem, the joint probability density function $p(\gamma_1, \dots, \gamma_s)$ can be split into the product of another joint probability function and a conditional one,

$$p(\gamma_1, \dots, \gamma_s) = p(\gamma_2, \dots, \gamma_s) \cdot p(\gamma_1 / (\gamma_2, \dots, \gamma_s)) \quad (13)$$

and applying $n-1$ times the same theorem on the remaining joint probability,

$$p(\gamma_1, \dots, \gamma_s) = p(\gamma_s) \cdot \prod_{i=1}^{s-1} p(\gamma_i / (\gamma_{i+1}, \dots, \gamma_s)) \quad (14)$$

If we use equation (12) to estimate the conditional probabilities, then the joint probability $p(\mathbf{d}_1, \dots, \mathbf{d}_s)$ can be estimated as $p^*(\mathbf{d}_1, \dots, \mathbf{d}_s)$ where,

$$p^*(\mathbf{d}_1, \dots, \mathbf{d}_s) = p_s(\mathbf{d}_s) \cdot \prod_{i=1}^{s-1} \left[p_i(\mathbf{d}_i) \cdot \prod_{j=i+1}^s r_{ij}(\mathbf{d}_i, \mathbf{d}_j) \right] \quad (15)$$

and introducing the first factor into the productory, we have

$$p^*(\mathbf{d}_1, \dots, \mathbf{d}_s) = \prod_{i=1}^s p_i(\mathbf{d}_i) \prod_{i=1}^{s-1} \prod_{j=i+1}^s r_{ij}(\mathbf{d}_i, \mathbf{d}_j) \quad (16)$$

In the approximations of the joint probability in the FDG and FORG approaches, random vertices and random arcs are treated separately, for this reason the above expression can be split considering vertices and arcs separately as follows

$$p^*(\mathbf{a}_1, \dots, \mathbf{a}_n, \mathbf{b}_1, \dots, \mathbf{b}_m) = \prod_{i=1}^n p_i(\mathbf{a}_i) \prod_{i=1}^m p_i(\mathbf{b}_i) \prod_{i=1}^{n-1} \prod_{j=i+1}^n r_{ij}(\mathbf{a}_i, \mathbf{a}_j) \prod_{i=1}^n \prod_{j=1}^m r_{ij}(\mathbf{a}_i, \mathbf{b}_j) \prod_{i=1}^{m-1} \prod_{j=i+1}^m r_{ij}(\mathbf{b}_i, \mathbf{b}_j) \quad (17)$$

5 Approximation of the Joint Probability by FORGs

In the FORG approach, the Peleg coefficients between vertices and between arcs do not influence on the computation of the joint probability. That is, by assumption 1 and 2 (section 3.1), $r_{ij}(\mathbf{a}_i, \mathbf{a}_j) = 1$ and $r_{ij}(\mathbf{b}_i, \mathbf{b}_j) = 1$ for all the vertices and arcs, respectively. On the contrary, assumption 3 (sec 3.1) makes that the probability on the arcs be conditioned on the values of the vertices that the arc connects, $q_j(\mathbf{b}_j \mid \mathbf{a}_{j1}, \mathbf{a}_{j2})$. In a similar deduction to that of section 4.3, and considering assumption 1, we arrive at the equivalence:

$$q_j(\mathbf{b}_j \mid \mathbf{a}_{j1}, \mathbf{a}_{j2}) = p_j(\mathbf{b}_j) r_{j1j}(\mathbf{a}_{j1}, \mathbf{b}_j) r_{j2j}(\mathbf{a}_{j2}, \mathbf{b}_j). \text{ Thus,}$$

$$P_R(G \mid \mu) = \prod_{i=1}^n p_i(\mathbf{a}_i) \prod_{j=1}^m p_j(\mathbf{b}_j) \prod_{j=1}^m \prod_{i=1, j_2} r_{ij}(\mathbf{a}_i, \mathbf{b}_j) \quad (18)$$

6 Approximation of the Joint Probability by FDGs

In the FDG approach, the 2nd-order probabilities between vertices can be estimated from the marginal probabilities and the 2nd-order relations as follows (a similar expression is obtained for the arcs),

$$\begin{aligned} \Pr(\alpha_i = \mathbf{a}_i \wedge \alpha_j = \mathbf{a}_j) &= 0 \text{ if Condition }^*_{2nd} \\ \Pr(\alpha_i = \mathbf{a}_i \wedge \alpha_j = \mathbf{a}_j) &\approx p_i(\mathbf{a}_i) p_j(\mathbf{a}_j) \text{ otherwise} \end{aligned} \quad (19)$$

where the Condition $^*_{2nd}$ is

$$^*_{2nd} : \begin{cases} (A_\omega(\omega_i, \omega_j) \wedge \mathbf{a}_i \neq \Phi \wedge \mathbf{a}_j \neq \Phi) \vee (O_\omega(\omega_i, \omega_j) \wedge \mathbf{a}_i \neq \Phi \wedge \mathbf{a}_j = \Phi) \vee \\ (O_\omega(\omega_j, \omega_i) \wedge \mathbf{a}_i = \Phi \wedge \mathbf{a}_j \neq \Phi) \vee (E_\omega(\omega_i, \omega_j) \wedge \mathbf{a}_i = \Phi \wedge \mathbf{a}_j = \Phi) \end{cases} \quad (20)$$

Note that, in the first case, it can be assured that the joint probability is null, but in the second case, we assume that the random elements are independent and the probability is estimated as a product of the marginal ones.

Thus, the Peleg coefficients are simplified as r_{ij}' , using equation (7) and (19),

$$r_{ij}'(\mathbf{a}_i, \mathbf{a}_j) = \begin{cases} 0 & \text{if } \alpha_{j_1} \wedge \alpha_{j_2} \wedge p_i(\mathbf{a}_i) \neq 0 \wedge p_j(\mathbf{a}_j) \neq 0 \\ 1 & \text{otherwise} \end{cases} \quad (21)$$

Moreover, due to the independence assumption 2 (sec 3.2), it is not possible to have a non-null arc and a null vertex as one of its endpoints in an outcome graph. Thus, we have $p(\alpha_{j_1} = \Phi \wedge \beta_j \neq \Phi) = 0$ and $p(\alpha_{j_2} = \Phi \wedge \beta_j \neq \Phi) = 0$. In the other cases, by assumption 1, they are assumed to be independent and so computed as the product of the marginal ones. The Peleg coefficients between vertices and arcs are simplified as

$$r_{ij}''(\mathbf{a}_i, \mathbf{b}_j) = \begin{cases} 0 & \text{if } (i = j_1 \vee j_2) \wedge \mathbf{a}_i = \Phi \wedge \mathbf{b}_j \neq \Phi \\ 1 & \text{otherwise} \end{cases} \quad (22)$$

The final expression of the joint probability of an outcome AG with respect to an FDG is

$$P_R(G|\mu) = \prod_{i=1}^n p_i(\mathbf{a}_i) \prod_{j=1}^m p_j(\mathbf{b}_j) \prod_{i=1}^n \prod_{j=1}^m r_{ij}'(\mathbf{a}_i, \mathbf{a}_j) \prod_{i=1}^m \prod_{j=1}^m r_{ij}''(\mathbf{b}_i, \mathbf{b}_j) \prod_{j=1}^m \prod_{i=j_1, j_2} r_{ij}''(\mathbf{a}_i, \mathbf{b}_j) \quad (23)$$

7 Conclusions and Future Work

We have presented a general formulation of an approximation of the joint probability of random elements in a RG, that describes a set of AGs, based on 2nd-order probabilities and marginal ones. We have seen that the FORG and FDG approaches are two specific cases (simplifications) of this general 2nd-order formulation. In both cases, the marginal probabilities of the random vertices and arcs are considered but the difference between them is in how are the 2nd-order relations between vertices or arcs estimated. FORGs keep only the 2nd-order probability between arcs and their extreme vertices, since the other joint probabilities are estimated as a product of the marginal ones. On the contrary, FDGs keep only a qualitative and structural information of the 2nd-order probabilities between all the vertices and arcs. If we compare both methods, FORGs have local (arc and endpoint vertex) 2nd-order semantic knowledge of the set of AGs but do not use any 2nd-order structural information of the set of the AGs. FDGs do not keep any 2nd-order semantic information but include the 2nd-order structural information of all the set of AGs. For this reason, the storage space of FORGs increases to the square on the size of the random-element domain but the FDGs increases to the square on the number of vertices and arcs.

However, the most important implication of the given general formulation of the 2nd-order random graph representation is that it opens the door to the development of other probabilistic graph approaches, either full 2nd-order or not. In addition, it is interesting to study empirically the relation between the amount of data to be kept in the model and the recognition ratio and run time in several applications. That is, to know in which applications is worthwhile to use explicitly the 2nd-order probabilities through the Peleg coefficients or is enough to estimate them by other ways less costly in space requirements, such as FORGs and FDGs. Moreover, a distance between the structural model and an AG can be defined. This is left for future research.

References

1. R. Alquézar, F. Serratosa, A. Sanfeliu, "Distance between Attributed Graphs and Function-Described Graphs relaxing 2nd order restrictions". *Proc. SSPR'2000 and SPR'2000*, Barcelona, Spain, Springer LNCS-1876, pp. 277-286, 2000.
2. S. Peleg and A. Rosenfeld, "Determining compatibility coefficients for curve enhancement relaxation processes", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 8, pp. 548-555, 1978.
3. Sanfeliu, F. Serratosa and R. Alquezar, "Clustering of attributed graphs and unsupervised synthesis of function-described graphs", *Proceedings ICPR'2000, 15th Int. Conf. on Pattern Recog.*, Barcelona, Spain, Vol.2, pp. 1026-1029, 2000.
4. K. Sengupta and K. Boyer, "Organizing large structural modelbases", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 321-332, 1995.
5. F. Serratosa, R. Alquezar and A. Sanfeliu, "Efficient algorithms for matching attributed graphs and function-described graphs", in *Proceedings ICPR'2000, 15th Int. Conf. on Pattern Recognition*, Barcelona, Spain, Vol.2, pp. 871-876, 2000.
6. A.K.C. Wong and M. You, "Entropy and distance of random graphs with application to structural pattern recognition", *IEEE Trans. on PAMI.*, vol. 7, pp. 599-609, 1985.
7. H. Bunke, "Error-tolerant graph matching: a formal framework and algorithms". *Proc. Workshops SSPR'98 & SPR'98*, Sydney, Australia, Springer LNCS-1451, pp.1-14, 1998.
8. D.S. Seong, H.S. Kim & K.H. Park, "Incremental Clustering of Attributed Graphs", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, pp. 1399-1411, 1993.
9. 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.