Investigate parameters of Fürer's algorithm

Capita Selecta: Artifical Intelligence (|H05N0a|)

Li Quan

April 26, 2011

In this report, we discuss the influence of the parameters in Fürer's algorithm for approximately counting random embeddings into random graphs [1]. The algorithm depends on a new graph decomposition called the ordered bipartite decomposition[1].

Because the algorithm Embeddings returns an unbiased estimator, use of Chebyshev's inequality implies that repeating the algorithm $O(\epsilon^{-2}\mathbb{E}_{\mathcal{A}}[Z^2]/\mathbb{E}_{\mathcal{A}}[Z]^2)$ times and taking the mean of the outputs results in an fraps for estimating $C_H(G) = C$.

Therefore, to evaluate the efficiency, the critical ratio $\mathbb{E}_{\mathcal{A}}[Z^2]/\mathbb{E}_{\mathcal{A}}[Z]^2$ must be evaluated (and the complexity of the stochastic experiment itself, i.e., the algorithm Embeddings). In the paper, the (related) easier ratio $\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[Z^2]]/\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[Z]]^2$ (critical ratio of averages) is bounded. The main result and the conditions are stated in the following lemma:

Lemma 1 (Main result [1, 2]). *Let H be a graph on n vertices. Let $e_H = \alpha N = \alpha(n)N$, and let $p = p(n) \in (0,1)$ with pN an integer ($N = \binom{n}{2}$). Let $\nu = \max\{2, \gamma\}$, with*

$$\gamma = \gamma(H) = \max_{3 \leq s \leq n} \{\max\{e_F : F \subseteq H, v_F = s\}/(s-2)\}.$$

Suppose the following conditions hold:

$$(1)\quad pN \to \infty \qquad (2)\quad np^{\nu}/\Delta^4 \to \infty \qquad (3)\quad \alpha^3 N p^{-2} \to 0.$$

Then, if H is bounded-degree by Δ, w.h.p. for a random graph $G \in \mathcal{G}(n,p)$, the critical ratio is polynomially bounded in n: $\mathbb{E}_{\mathcal{A}}[Z^2]/\mathbb{E}_{\mathcal{A}}[Z]^2 = O(\text{poly}(n))$, where poly($n$) depends on w and p.

[1]([1]) Informally, such a decomposition generates a labeling of vertices such that every edge is between vertices with different labels and for every very vertex all neighbors with a higher label have identitical labels. The labeling implicitely generates a sequence of bipartite graphs, and the crucial part is to ensure that each of the bipartite graphs is of small size. The size of the largest bipartite graph defines the width of the decomposition ($w = \max_i |E(H_i)| = \max_i e_{H_i}$).

This means that if H has a decomposition of bounded width w, for almost all graphs G, running the algorithm $\text{poly}(n)\epsilon^{-2}$ times and taking the mean, results in an $(1 \pm \epsilon)$-approximation for C, where $\text{poly}(n)$ is a polynomial in n depending on w and p. Because each run of the algorithm happens in polynomial time (as H is bounded-degree, the calculation of the number of embeddings can be done in polynomial time), the result is a fraps.

Clearly, w defines the efficiency of the algorithm: the larger it is, the more times the algorithm must be run to get a fraps for C.

Now we consider the number of vertices in a partition class of the decomposition $|V_i|$. V_i represents the set of vertices of H which get embedded into G during the ith stage of the algorithm (using the already constructed mapping of U_i). Intuitively, the number of vertices in the partition classes is somewhat related to the width of the decomposition (a trivial lower-bound was already given in the paper, more specifically $\Delta/2$).

Lemma 2. *Let $V_1, \ldots, V_l$ be a decomposition of a graph $H = (V_H, E_H)$. Let s and t denote the sizes of the two largest partition classes. A (worst-case) upper-bound for the width of the decomposition is st.*

Lemma 3. *Let $V_1, \ldots, V_l$ be a decomposition of a graph $H = (V_H, E_H)$ on n vertices. Let v_i denote the number of vertices in partition class V_i. An upper bound for the critical ratio of averages is then $(1 + \frac{c}{n})(1 + \frac{c}{n-v_1}) \ldots (1 + \frac{c}{n-\sum_{j<\ell} v_j})$.*

This means that n should be large enough and that most vertices should be found in the later partition classes. Intuitively, in earlier stages we do not want a large set of vertices to embed.

Lemma 1 and conditions (1)–(2) of Lemma 3 imply that the number of nodes of the pattern should be large enough: we want to observe "interesting" patterns. This also means that the considered network should be large enough to have some embeddings of the pattern (therefore, it should also have a relatively large number of edges), but of course it should not be too large with respect to the pattern as that would make the search space huge.

Condition (3) of Lemma 3 can be rewritten to $e_H^3 n^{-4} p^{-2} \to 0$: the number of edges of H should not be to large with respect to n. Again, the probability to have an edge between two nodes in the network p should be relatively large.

In conclusion, the algorithm works well when the count is not too small (as already stated before but it is just the essence of the algorithm): the number of vertices in both the pattern and network graph should be relatively large enough and the network graph should not be too sparse. As the decomposition width of the pattern graph plays an important role in the efficiency, it should be relatively sparse (but not too sparse, as that would be an uninteresting pattern).

A. Some examples

Consider the network graph G and the pattern graph H shown in Figure 1.

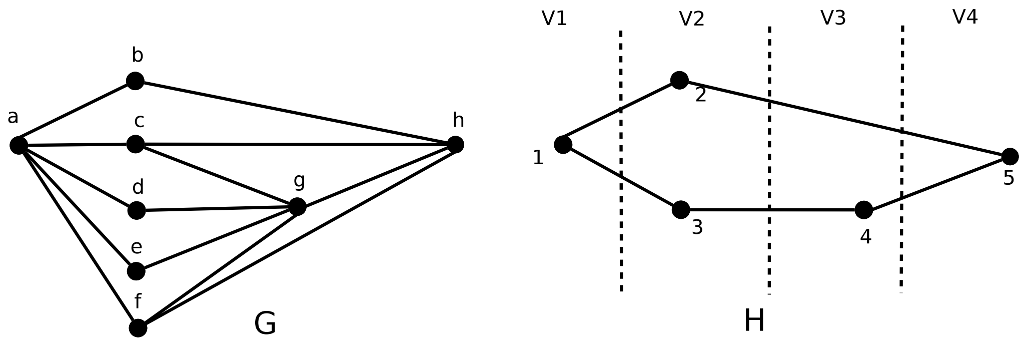

Figure 1: A network graph G and a pattern graph H with decomposition width 2.

We first show the algorithm for H where $U_i = N_H(V_i) \cap V^{i-1}$ and where $U_1 = \emptyset$; $U_2 = \{1\}$; $U_3 = \{3\}$ and $U_4 = \{2, 4\}$. Now we can easily construct the $H_i = H[U_i \cup V_i]$: $H_1 = H[\{1\}]$; $H_2 = H[\{1, 2, 3\}]$; $H_3 = H[\{3, 4\}]$ and $H_4 = H[\{4, 5\}]$. The width of the decomposition is $w = e_{H_2} = e_{H_4} = 2$. A succesful run of the Embeddings algorithm might go as follows:

- $i \leftarrow 1$
  - $G_f \leftarrow G[\{a, b, c, d, e, f, g, h\}]$
  - $X_1 \leftarrow 8$
  - $\varphi \leftarrow \{(1, a)\}$
  - $X \leftarrow 8$
  - $Mark(1) \leftarrow \{a\}$
- $i \leftarrow 2$
  - $G_f \leftarrow G[\{a, b, c, d, e, f, g, h\}]$
  - $X_2 \leftarrow 5 \times 4 = 20$
  - $\varphi \leftarrow \{(1, a); (2, b); (3, d)\}$
  - $X \leftarrow 8 \times 20 = 160$
  - $Mark(2) \leftarrow \{a, b, d\}$
- $i \leftarrow 3$
  - $G_f \leftarrow G[\{c, d, e, f, g, h\}]$
  - $X_3 \leftarrow 1$
  - $\varphi \leftarrow \{(1, a); (2, b); (3, c); (4, g)\}$
  - $X \leftarrow 160$
  - $Mark(3) \leftarrow \{a, b, d, g\}$
- $i \leftarrow 4$
  - $G_f \leftarrow G[\{b, c, e, f, g, h\}]$
  - $X_4 \leftarrow 1$
  - $\varphi \leftarrow \{(1, a); (2, b); (3, c); (4, g); (5, h)\}$
  - $X \leftarrow 160$
  - $Mark(4) \leftarrow \{a, b, c, g, h\}$

The algorithm returns $X/|aut(H)| = 160/(4!) = 20/3$, which means the embedding was found with probability $3/20 = 0.15$.

Now consider the pattern graph H in Figure 2, where $U_1 = \emptyset$; $U_2 = \{1\}$; $U_3 = \{2,3,4,5,6\}$ and $U_4 = \{2,7\}$, and, $H_1 = H[\{1\}]$; $H_2 = H[\{1,2,3,4,5,6\}]$; $H_3 = H[\{3,4,5,6,7\}]$ and $H_4 = H[\{2,7,8\}]$. The width of the decomposition is $w = \max_i e_{H_i} = e_{H_2} = 5$.

A succesful run of the Embeddings algorithm:

- $i \leftarrow 1$
  - $G_f \leftarrow G[\{a,b,c,d,e,f,g,h\}]$
  - $X_1 \leftarrow 8$
  - $\varphi \leftarrow \{(1,a)\}$
  - $X \leftarrow 8$
  - $Mark(1) \leftarrow \{a\}$
- $i \leftarrow 2$
  - $G_f \leftarrow G[\{a,b,c,d,e,f,g,h\}]$
  - $X_2 \leftarrow 5! = 120$
  - $\varphi \leftarrow \{(1,a);(2,b);(3,c);(4,d);(5,e);(6,f)\}$
  - $X \leftarrow 8 \times 120 = 960$
  - $Mark(2) \leftarrow \{a,b,c,d,e,f\}$
- $i \leftarrow 3$
  - $G_f \leftarrow G[\{b,c,d,e,f,g,h\}]$
  - $X_3 \leftarrow 1$
  - $\varphi \leftarrow \{(1,a);(2,b);(3,c);(4,d);(5,e);(6,f);(7,g)\}$
  - $X \leftarrow 960$
  - $Mark(3) \leftarrow \{a,b,c,d,e,f,g\}$
- $i \leftarrow 4$
  - $G_f \leftarrow G[\{b,g,h\}]$
  - $X_4 \leftarrow 1$
  - $\varphi \leftarrow \{(1,a);(2,b);(3,c);(4,d),(5,e);(6,f);(7,g);(8,h)\}$
  - $X \leftarrow 960$
  - $Mark(4) \leftarrow \{a,b,c,d,e,f,g,h\}$

The algorithm returns $X/|aut(H)| = 960/(4!) = 40$, which means the embedding was found with probability $1/40 = 0.025$, which means that this embedding has 6 times less probability of being found than the previous one.

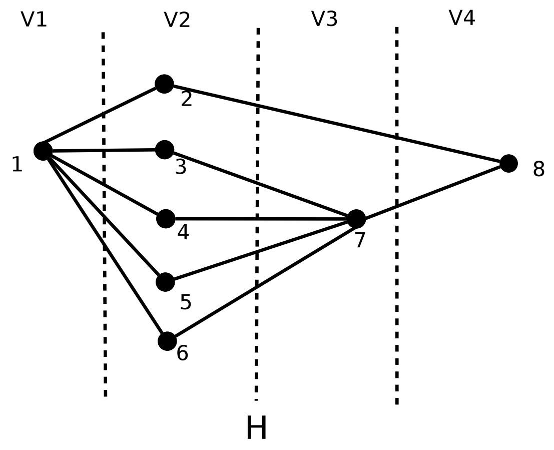

Figure 2: A pattern graph H with decomposition width 5.

B. Derivation

Proof of Lemma 2. The width is given by $w = \max_i |E(H_i)|$, where $H_i = H[U_i \cup V_i]$ with $U_i = N_H(V_i) \cap V^{i-1}$ and $V^i = \bigcup_{j \le i} V_j$. As the H_i are bipartite, the maximum number of edges is st (reached when the complete bipartite graph $K_{s,t}$ is considered).

□

Proof of Lemma 3. Using the proof of the main theorem in [1]. $V_1, \dots, V_\ell$ denotes a decomposition of H with width w. Consider the bipartite graph $H_i = (U_i, V_i, E_{H_i})$. Let $n_i = n - \sum_{j<i} v_j$, and $L_i = L_{H_i|U_i}(G_i)$, the number of embeddings of H_i in $G_i \in \mathcal{G}(n_i + u_i, p)$ where the mapping of the vertices in U_i to the distinguished vertices in G_i is given.

$$\frac{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[Z^2]]}{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[Z]]^2} = \prod_{i=1}^{\ell} \frac{\mathbb{E}[L_i^2]}{\mathbb{E}[L_i]^2} \le \prod_{i=1}^{\ell} (1 + \frac{c}{n_i})$$

In the paper the authors bound the last term to $O(n^c)$ by a telescoping argument (since $n = n_1 > n_2 > \dots > n_\ell$).

Clearly, we can derive a stricter (more complex) formulation of the upper bound by explicitely writing it in terms of the numbers of vertices in the partition classes v_i.

$$\begin{aligned}\prod_{i=1}^{\ell}(1 + \frac{c}{n_i}) &= (1 + \frac{c}{n_1})(1 + \frac{c}{n_2}) \dots (1 + \frac{c}{n_\ell}) \\ &= (1 + \frac{c}{n})(1 + \frac{c}{n - v_1}) \dots (1 + \frac{c}{n - \sum_{j<\ell} v_j})\end{aligned}$$

□

References

[1] M. Fürer and S. Prasad Kasiviswanathan. Approximately counting embeddings into random graphs. In *Proceedings of the 11th international workshop, APPROX 2008, and 12th international workshop, RANDOM 2008 on Approximation, Randomization and Combinatorial Optimization: Algorithms and Techniques*, APPROX '08 / RANDOM '08, pages 416–429, Berlin, Heidelberg, 2008. Springer-Verlag.

[2] O. Riordan. Spanning subgraphs of random graphs. *Combinatorics, Probability & Computing*, 9(2):125–148, 2000.