Sampling and Counting Acyclic Orientations in Chordal Graphs (Student Abstract)

Wenbo Sun[*]

Rochester Institute of Technology
Rochester, NY, U.S.A.
ws3109@rit.edu

Abstract

Sampling of chordal graphs and various types of acyclic orientations over chordal graphs plays a central role in several AI applications such as causal structure learning. For a given undirected graph, an acyclic orientation is an assignment of directions to all of its edges which makes the resulting directed graph cycle-free. Sampling is often closely related to the corresponding counting problem. Counting of acyclic orientations of a given chordal graph can be done in polynomial time, but the previously known techniques do not seem to lead to a corresponding (efficient) sampler. In this work, we propose a dynamic programming framework which yields a counter and a uniform sampler, both of which run in (essentially) linear time. An interesting feature of our sampler is that it is a stand-alone algorithm that, unlike other DP-based samplers, does not need any preprocessing which determines the corresponding counts.

Introduction

Sampling and counting of different types of acyclic orientations over chordal graphs attracted attention in several AI research areas, for example in structure learning of Bayesian networks (Ganian, Hamm, and Talvitie 2020; Ghassami et al. 2019; Talvitie and Koivisto 2019; Wienöbst, Bannach, and Liskiewicz 2020). A graph is chordal if each of its cycles of length at least four has an edge that connects two nonadjacent vertices in that cycle. Counting and sampling itself is a hot research area in the theoretical community. Counting of acyclic orientations is a special case of the Tutte polynomial of a graph, which plays an important role in graph theory and it is related to some important graph quantities such as the chromatic number (number of colorings) and the number of strong orientations. Computing these quantities is usually NP-hard on general graphs, so calculating them efficiently under some constraints, such as counting over special types of graphs, is a natural research direction. It has been proved that the number of acyclic orientations is equal to the evaluation of the chromatic polynomial at -1 (Stanley 1973). Remarkably, the calculation of the chromatic polynomial can be done in polynomial time on chordal graphs (Naor, Naor, and Schäffer 1989), and, therefore, the number of acyclic orientations can be computed in polynomial time in this graph class. But, to the best of our knowledge, the problem is not known to be self-reducible and, prior to our work, the existence of an efficient uniform random sampler for acyclic orientations in chordal graphs was unknown.

In this work we present a dynamic programming approach over the clique tree structure of chordal graphs, which yields an almost linear time counter and a linear time uniform sampler for acyclic orientations in this graph class.

Preliminaries: Chordal Graphs, Clique Trees

For a graph G, we use $G[U]$ to denote the graph induced in G on the vertex set $U \subseteq V(G)$. Every chordal graph G can be represented by a clique tree T_G where $V(T_G)$ is the set of maximal cliques of G and the tree satisfies the induced subtree property: For every vertex $v \in V(G)$, the induced subgraph $T_G\left[A_v\right]$ is connected, where A_v is the set of maximal cliques of G containing v. Let T_{G,C_r} be the clique tree T_G rooted at a maximal clique C_r. If G is clear from the context, we will simply write T_{C_r}. We denote by $T_{C_r,C}$ the subtree of T_{C_r} containing C and its descendants; we write T_C if C_r is clear from the context.

Each clique C in T_{C_r} can be partitioned into a separator set $\operatorname{Sep}(C) = C \cap \operatorname{Parent}(C)$ and a residual set $\operatorname{Res}(C) = C \setminus \operatorname{Sep}(C)$, where Parent (C) is the parent clique of C in T_{C_r} (if $C = C_r$, then $\operatorname{Parent}(C) = \emptyset$). The following properties hold:

- For each vertex v in G, there is a unique clique C_v that contains v in its residual set. This implies that $\left|V\left(T_G\right)\right| \leq |V(G)|$ and that C_v is the root of $T_{C_r}\left[A_v\right]$; we denote this rooted subtree by T_{C_v}. All other cliques in T_{C_v} that contain v have it in their separator set.
- For a clique C let $D(C)$ be the set of vertices in the descendant cliques of C in T_{C_r} except $\operatorname{Sep}(C)$, i.e., $D(C) := \cup_{C' \in V(T_C)} C' - \operatorname{Sep}(C)$. Let $A(C)$ be the vertices in the cliques not in T_C except $\operatorname{Sep}(C)$, i.e., $A(C) := \cup_{C' \in V(T_{C_r}) - V(T_C)} C' - \operatorname{Sep}(C)$. The separator $\operatorname{Sep}(C)$ separates $A(C)$ and $D(C)$ in G : there is no edge with one endpoint in $A(C)$ and the other in $D(C)$.
- Construction of a clique tree for a connected chordal graph can be done in time $O(|E(G)|)$.

We use $G\left[T_C\right]$ for the subgraph induced by the vertices that belong to cliques in T_C i.e., $G\left[T_C\right] := G\left[\cup_{C' \in V(T_C)} C'\right]$.

[*] Partially supported by NSF award 1819546.

We also define the following subgraph of $G\left[T_C\right]$: Let $\hat{G}\left[T_C\right]$ be $G\left[T_C\right]$ with the edges within the separator set $\operatorname{Sep}(C)$ removed, i.e., $\hat{G}\left[T_C\right] := G\left[T_C\right] - E(G[\operatorname{Sep}(C)])$.

Our Contribution

The main contribution of our work is summarized in Theorem 3 and in Algorithm 1, which generates a uniformly random acyclic orientation for the given chordal graph. In other words, each orientation is generated with probability $1/|\Omega|$, where Ω is the set of all acyclic orientations of the graph. The proof of Theorem 3 relies on the following two lemmas (their proofs, as well as the full proof of the theorem, are available upon request, and will be included in the author's thesis).

Lemma 1. *Let C be a clique in the rooted clique tree T and let $C_1, C_2, \ldots, C_d$ be its children cliques. The edge sets of the graphs $\hat{G}\left[T_{C_i}\right], i = 1, \ldots, d$, are mutually disjoint.*

Lemma 2. *Let G be a connected chordal graph and let T be a rooted clique tree of G. For a clique C in T and an acyclic orientation σ over C, let $\mathrm{AO}\left(T_C, \sigma\right)$ be the set of acyclic orientations on $G\left[T_C\right]$ that are consistent with σ. For any C and any two acyclic orientations σ_1 and σ_2 over C, we have*

$$\left|\mathrm{AO}\left(T_C, \sigma_1\right)\right| = \left|\mathrm{AO}\left(T_C, \sigma_2\right)\right|.$$

In order to make the running times of our algorithms more readable, we assume that each arithmetic operation takes a constant time. This is, of course, a bit optimistic, since the ultimate number of orientations can be as high as 2^m for a graph with m edges, and, therefore, the true running time of each arithmetic operation adds a factor of about $m \operatorname{polylog}(m)$. We use $\tilde{O}()$ notation to indicate that this factor is omitted from our running time estimate.

Theorem 3. *Let G be a connected chordal graph. The number of its acyclic orientations can be calculated in $\tilde{O}(|V(G)|) + O(|E(G)|)$ time.*

Proof sketch. Let T be a clique tree of G rooted at a clique C_r. For a clique C in T, we define $\mathrm{AO}\left(T_C\right)$ as the number of acyclic orientations of $G\left[T_C\right]$ under the assumption that the orientation of the edges of $G[\operatorname{Sep}(C)]$ has been fixed. Then, $\mathrm{AO}\left(T_{C_r}\right)$ computes the overall number of acyclic orientations of G, since $\operatorname{Sep}\left(C_r\right) = \emptyset$. We show how to compute $\mathrm{AO}\left(T_C\right)$ by dynamic programming over the clique tree:

$$\mathrm{AO}\left(T_C\right) = \frac{|C|!}{|\operatorname{Sep}(C)|!} \prod_{C_{i=1}^{d}} \mathrm{AO}\left(T_{C_i}\right), \qquad (1)$$

where $C_1, \ldots, C_d$ are the children cliques of C in T (and $d = 0$ if C is a leaf of T). Let $\sigma_{\operatorname{Sep}(C)}$ be the given orientation of $G[\operatorname{Sep}(C)]$, we extend it to an acyclic orientation σ_C over C by picking an orientation from the $\frac{|C|!}{|\operatorname{Sep}(C)|!}$ candidates. The calculation is obviously correct when C is a leaf. When C is a non-leaf, we fix an arbitrary σ_C and let $\sigma_{\operatorname{Sep}(C_i)}$ be the orientation restricted to $G\left[\operatorname{Sep}\left(C_i\right)\right]$, and we use A_i to denote the set of acyclic orientations of $G\left[T_{C_i}\right]$ consistent with $\sigma_{\operatorname{Sep}(C_i)}$. By Lemma 1, we can prove that there is a bijection between $\mathrm{AO}\left(T_C, \sigma_C\right)$ and $A_1 \times \cdots \times A_d$.Then, by the inductive hypothesis, the correct calculation of $\left|\mathrm{AO}\left(T_{C_i}\right)\right|$ yields that $\prod_{C_{i=1}^{d}} \mathrm{AO}\left(T_{C_i}\right) = \left|\mathrm{AO}\left(T_C, \sigma_C\right)\right|$, and by Lemma 2 we know that $\left|\mathrm{AO}\left(T_C, \sigma_C\right)\right|$ does not depend on the specific orientation of σ_C, and there are $\frac{|C|!}{|\operatorname{Sep}(C)|!}$ possible σ_C's, yielding the expression 1.

The construction of T costs $O(|V(G)|)$, and each clique C of T is processed exactly once and it costs $O\left(\deg_T(C) + 1\right)$. Hence the running time is $\tilde{O}(|V(G)|) + O(|E(G)|)$. □

Algorithm 1: Sample acyclic orientations uniformly at random

```
Construct a clique tree of the input graph and randomly pick a clique $C_r$ as the root. Choose a uniformly random ordering of the vertices in $C_r$.
Let $C$ be a current clique that we are processing in a depth-first search manner starting with $C_r$. Pick a uniformly random ordering of $C$ that is consistent with $G[\operatorname{Sep}(C)]$ by choosing a random ordering $\pi$ of $C$ and replacing the relative order of $\operatorname{Sep}(C)$ in by the given ordering.
Output the oriented graph.
```

The proof of Theorem 3 naturally yields a uniform sampler of acyclic orientations (see Algorithm 1), which runs in time $O(|E(G)|)$ for any input chordal graph G.

Future Plans

We view this work as a first step in the direction of counting and sampling different types of graph orientations on chordal graphs, such as bipolar orientations, sink-free orientations, and strong orientations.

References

Ganian, R.; Hamm, T.; and Talvitie, T. 2020. An Efficient Algorithm for Counting Markov Equivalent DAGs. In *AAAI*, 10136–10143.

Ghassami, A.; Salehkaleybar, S.; Kiyavash, N.; and Zhang, K. 2019. Counting and sampling from Markov equivalent DAGs using clique trees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3664–3671.

Naor, J.; Naor, M.; and Schäffer, A. A. 1989. Fast parallel algorithms for chordal graphs. *SIAM Journal on Computing*, 18(2): 327–349.

Stanley, R. P. 1973. Acyclic orientations of graphs. *Discrete Mathematics*, 5(2): 171–178.

Talvitie, T.; and Koivisto, M. 2019. Counting and sampling Markov equivalent directed acyclic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 7984–7991.

Wienöbst, M.; Bannach, M.; and Liskiewicz, M. 2020. Polynomial-Time Algorithms for Counting and Sampling Markov Equivalent DAGs. *arXiv preprint arXiv:2012.09679.*