Model Counting and Sampling in First-Order Logic

Ondřej Kuželka[1]

[1]*Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic*

Abstract

First-order model counting (FOMC) asks to count models of a first-order logic sentence over a given set of domain elements. Its weighted variant, WFOMC, generalizes FOMC by assigning weights to the models and has applications among others in statistical relational learning. Several non-trivial classes of WFOMC problems that can be solved in time polynomial in the number of domain elements were identified in the literature over the past decade, since the two seminal paper by Van den Broeck (2011) and Van den Broeck, Meert and Darwiche (2014) established tractability of $\mathbf{FO}^2$ for WFOMC. This talk is about recent developments on WFOMC and the related problem of weighted first-order model sampling (WFOMS). We also discuss applications of WFOMC and WFOMS, such as automated solving of problems from enumerative combinatorics and elementary probability theory.

Keywords

Model counting, sampling, first-order Logic,

1. Weighted First-Order Model Counting

Weighted first-order model counting (WFOMC) is the task of computing the weighted sum of models of a given first-order logic sentence [1]. It has applications in statistical relational learning [2] and it is also relevant for probability theory and combinatorics [3, 4]. Formally, given a set of domain elements Δ and two functions $w(P)$ and $\overline{w}(P)$ mapping predicate symbols to real numbers, we define WFOMC as

$$\text{WFOMC}(\Psi, \Delta, w, \overline{w}) = \sum_{\omega:\omega \models \Psi} W(\omega, w, \overline{w}),$$

where $W(\omega)$, i.e. the weight of ω, is computed as

$$W(\omega, w, \overline{w}) = \prod_{\omega \models l} w(\text{pred}(l)) \prod_{\omega \models \neg l} \overline{w}(\text{pred}(l)).$$

Example 1 (Illustrating the notion of WFOMC). *Let us have a domain $\Delta = \{A, B\}$, and suppose that we only have two predicates heads and tails, with weights given as $w(heads) = 2$, $w(tails) = \overline{w}(heads) = \overline{w}(tails) = 1$. We will want to compute the WFOMC of the sentence $\Gamma = \forall x : (heads(x) \lor tails(x)) \land (\neg heads(x) \lor \neg tails(x))$. There are four models of Γ on the domain*

Δ: $\omega_1 = \{heads(A), heads(B)\}$, $\omega_2 = \{heads(A), tails(B)\}$, $\omega_3 = \{tails(A), heads(A)\}$ and $\omega_4 = \{tails(A), tails(B)\}$. The resulting first-order model count is $\mathrm{FOMC}(\Gamma, \Delta) = 4$ and the weighted model count is $\mathrm{WFOMC}(\Gamma, w, \overline{w}, \Delta) = 4 + 2 + 2 + 1 = 9$.

Example 2. *Consider the following variant of the Secret Santa game. There are n people attending the Christmas party. Each person is supposed to give presents to k other people and receive k presents.[1] The organizer decides to let everyone draw names from k hats each of which contains everyone's name once. The participants then come and each of them draws one name at random from each hat. We ask what is the probability that no one, at the same time, (i) draws their own name and (ii) draws the same name more than once. When $k = 1$, the solution is known since the 18th century and it is the ratio of the number of derangements (i.e., permutations without fixed points) and the number of permutations on n elements. For $k > 1$ the problem could still be solved by hand, but with non-trivial effort and with good knowledge of combinatorics. The same problem could, however, also be solved automatically using WFOMC in polynomial time because it can be encoded in the two-variable fragment of first-order logic with counting quantifiers $\boldsymbol{C}^2$ [5], which is known to be tractable [6] (what we mean by "tractable" is explained in Section 3). The $\boldsymbol{C}^2$ sentence with which we can achieve it is:*

$$\Psi = \bigwedge_{i=1}^{k} \left(\forall x \exists^{=1} y \ Draw_i(x, y) \land \forall y \exists^{=1} x \ Draw_i(x, y) \land \forall x \ \neg Draw_i(x, x) \right) \land$$

$$\bigwedge_{i=1}^{k} \bigwedge_{j<i} \left(\forall x \forall y \ \neg Draw_i(x, y) \lor \neg Draw_j(x, y) \right).$$

Here, $Draw_i(x, y)$ represents that x drew the name y from the hat i. To compute the desired probability, we just need to divide the WFOMC of the sentence Ψ on a domain of size n by $(n!)^k$ which is the total number of all possible draws (obviously, if we did not know factorials, we could compute this number by WFOMC as well).

2. Weighted First-Order Model Sampling

Another useful task one might need to solve is, given a first-order logic sentence, a domain and weighting functions, to sample a model of the sentence with probability proportional to its weight. This problem is known as *weighted first-order model sampling* (WFOMS) [7]. It has again applications in statistical relational learning, as it allows sampling from Markov logic networks [8, 9, 10], but its potential applications are much broader—it generally provides a principled approach for efficient sampling of combinatorial structures.

Example 3. *Suppose that after computing the probability of a successful draw in the Secret Santa game from Example 2 (a successful draw is one in which no one got their own name and no one got name of anyone else more than once), we conclude that the probability is too low for this procedure to be practical. Then we might want to sample an assignment of who should give presents to whom automatically. To achieve that we can sample models of the $\boldsymbol{C}^2$ sentence from Example 2 using WFOMS. It turns out that sampling is also tractable for sentences from the $\boldsymbol{C}^2$ fragment [10].*

---

[1] In the normal variant of the Secret Santa game, $k = 1$.

3. Tractability

An important question about WFOMC and WFOMS is: How hard are they? First, it is not difficult to observe that there is no hope for a WFOMC algorithm that would scale polynomially with the size of the first-order logic sentence even if the sentence was guaranteed to have just one logical variable (unless $\mathbf{P} = \#\mathbf{P}$). Therefore focus has been on identifying fragments of first-order logic sentences for which WFOMC can be computed in time polynomial in the size of the domain. The term coined for this type of tractability by Van den Broeck [2011] is *domain liftability*. In general, WFOMC and WFOMS are intractable even in this sense already for some sentences from $\mathbf{FO}^3$ [12, 9], however, there are non-trivial fragments of first-order logic which have been identified as tractable for WFOMC [13, 11, 14, 15, 16, 6, 17, 18, 19] and for WFOMS [7, 9], including the fragment $\mathbf{C}^2$ [6, 10].

4. Some Applications

Here we briefly mention several applications that go beyond the standard machine learning uses of WFOMC and WFOMS.

Combinatorics.   WFOMC is well-suited for solving problems from enumerative combinatorics.[2] The various tractable fragments of WFOMC can be used to count many other non-trivial structures as well, e.g., k-regular graphs, trees with k leaves etc. [3, 17]. The close connection between WFOMC and enumerative combinatorics was used for developing a method capable of generating a database of integer sequences with combinatorial interpretation [4] that we hope could complement the well known OEIS database [21].

Samplers.   Standard programming language libraries usually have some support for sampling. For instance, NumPy provides functionality for sampling simple structures such as permutations and combinations. However, when we need to sample a more complex structure, we typically need to develop a sampler from scratch. Having a declarative framework based on WFOMS would be a useful tool for programmers making their work easier. The main reason why there is not such a framework based on the current WFOMS algorithms [9, 10] yet is their practical performance, which is the focus of ongoing works. However, it is worth mentioning that, unsurprisingly, these algorithms already outperform state-of-the-art propositional model samplers on first-order sampling problems, as shown experimentally in [9].

Acknowledgments

---

[2]There are also other approaches to automating combinatorics within the broader context of *lifted inference*, most notably [20].

References

[1] G. Van den Broeck, N. Taghipour, W. Meert, J. Davis, L. De Raedt, Lifted probabilistic inference by first-order knowledge compilation, in: Proceedings of the Twenty-Second international joint conference on Artificial Intelligence, 2011, pp. 2178–2185.

[2] L. Getoor, B. Taskar, Introduction to statistical relational learning, volume 1, MIT press Cambridge, 2007.

[3] J. Barvínek, T. van Bremen, Y. Wang, F. Zelezný, O. Kuzelka, Automatic conjecturing of p-recursions using lifted inference, in: Inductive Logic Programming - 30th International Conference, ILP 2021, Proceedings, volume 13191 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 17–25. URL: https://doi.org/10.1007/978-3-030-97454-1_2. doi:10.1007/978-3-030-97454-1_2.

[4] M. Svatoš, P. Jung, J. Tóth, Y. Wang, O. Kuželka, On discovering interesting combinatorial integer sequences, in: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, 2023.

[5] E. Graedel, M. Otto, E. Rosen, Two-variable logic with counting is decidable, in: Proceedings of Twelfth Annual IEEE Symposium on Logic in Computer Science, IEEE, 1997, pp. 306–317.

[6] O. Kuželka, Weighted first-order model counting in the two-variable fragment with counting quantifiers, Journal of Artificial Intelligence Research 70 (2021) 1281–1307.

[7] Y. Wang, T. van Bremen, Y. Wang, O. Kuželka, Domain-lifted sampling for universal two-variable logic and extensions, in: Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22), AAAI Press, 2022.

[8] M. Richardson, P. Domingos, Markov logic networks, Machine Learning 62 (2006) 107–136.

[9] Y. Wang, J. Pu, Y. Wang, O. Kuželka, On exact sampling in the two-variable fragment of first-order logic, in: LICS '23: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, IEEE, 2023.

[10] Y. Wang, J. Pu, Y. Wang, O. Kuzelka, Lifted algorithms for symmetric weighted first-order model sampling, Artif. Intell. 331 (2024) 104114. URL: https://doi.org/10.1016/j.artint.2024.104114. doi:10.1016/J.ARTINT.2024.104114.

[11] G. Van den Broeck, On the completeness of first-order knowledge compilation for lifted probabilistic inference, in: Advances in Neural Information Processing Systems, 2011, pp. 1386–1394.

[12] P. Beame, G. Van den Broeck, E. Gribkoff, D. Suciu, Symmetric weighted first-order model counting, in: Proceedings of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, ACM, 2015, pp. 313–328.

[13] V. Gogate, P. M. Domingos, Probabilistic theorem proving, in: UAI 2011, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, Barcelona, Spain, July 14-17, 2011, 2011, pp. 256–265.

[14] G. Van den Broeck, W. Meert, A. Darwiche, Skolemization for weighted first-order model counting, in: Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR), 2014, pp. 1–10.

[15] S. M. Kazemi, A. Kimmig, G. V. den Broeck, D. Poole, New liftable classes for first-order probabilistic inference, in: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems, 2016, pp. 3117–3125.

[16] A. Kuusisto, C. Lutz, Weighted model counting beyond two-variable logic, in: Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, 2018, pp. 619–628.

[17] T. Van Bremen, O. Kuželka, Lifted inference with tree axioms, in: Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning, 2021, pp. 599–608.

[18] J. Tóth, O. Kuželka, Lifted inference with linear order axiom, in: Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI Press, 2023.

[19] S. Malhotra, D. Bizzaro, L. Serafini, Lifted inference beyond first-order logic, 2024. arXiv:2308.11738.

[20] P. Totis, J. Davis, L. De Raedt, A. Kimmig, Lifted reasoning for combinatorial counting, Journal of Artificial Intelligence Research 76 (2023) 1–58.

[21] OEIS Foundation Inc., The on-line encyclopedia of integer sequences, http://oeis.org, 2023. Accessed: 2023-05-01.