A Combinatorial Approach to Weighted Model Counting in the Two Variable Fragment with Cardinality Constraints

AIxIA 2021, Underline Science Inc.

Authors: Sagar Malhotra, Luciano Serafini
DOI: 10.48448/sf9b-gp87
Available from: https://underline.io/lecture/42130-a-combinatorial-approach-to-weighted-model-counting-in-the-two-variable-fragment-with-cardinality-constraints

Lecture Transcript

thank you very much. And we can proceed with the next talk which is a combination of a combinatorial approach to weight. The model counting in the two variable fragment with cardinality constraints by cigar malhotra and luciano Serafini. And the speaker is cigar.

Hello everyone. Thank you Viviana for the introduction. That saves me 30 seconds. All right. Uh So so yes. Today I'll be talking about combinatorial approach to weight and module counting in the two variable fragment. And this is joint work with my supervisor, luciano Serafini. So let's look at first why written model counting is an important problem. And with model counting is mostly used for travel stick influence. And the setup is this. So let's say you have a set of worlds. A probability space Where you can have multiple models. And each model has some real valued weight given by the weight function. W in equation one. Now the probability of a world is defined by the weight of the world, divided by the sum of the weight of all the worlds. And this sum is also known as partition function. The probability of a logical formula is defined as the sum of all the models which satisfy the logical formula divided by the sum of the weight of all the models that exist in this world.

So in this presentation we will focus on the first order logic variant of this problem where the formula that you have is enforced order logic and what you want to figure out is the weighted sum of all the models of this first order logic formula.

Okay and usually this problem is a sharpie. Hard so like it's harder than the more the anti heart problems even for for example for too sad where the satisfy ability is in polynomial time. The model counting is a sharpie. So we want to restrict ourselves somehow to efficient fragments and that is done by restricting W. Two a weight which is independent of the domain elements in the first order logic and restricting the fragments of functional logic. So you can't figure out fragments of functional logic where you there there there there is speed time, very functional model counting. And these fragments are called the main lift table. And today we will focus on one such fragment which is called fo two Which is abbreviation for forced all the logic with two variables. So if you do language you can have a set of Medicaid's and a set of variables and I mean this set of variables are just two variables X and Y. And you have some domain in which the models are interpreted an example is the number of

and the photographs that you can have and and the the possible and all the possible undirected graphs at a model of this formula which is basically cemetery.

Now the question is how many undirected graphs exist over end nodes. Okay. And that's what we're going to address today. So we're going to address this problem in general and I hope you're not scared by the slide. This slide is there to just give your intuition of what we are going to face. So let's say you have an arbitrary formula for all X. For all Y. Five X. Y. Like the cemetery formula introducing the previous slide. And you want to count the number of models of this formula over end domain elements. Okay. So the way we do it is we first enumerate all the possible ways that you can realize the unitary properties and I will later formalize what are the unity properties. But let's say the sky elements of the domain realize a certain you know your property. I then there are N choose K. Uh the multnomah coefficient. The red multinational coefficient. Number of ways of realizing these unity properties. The green multi normal coefficient which has in it the triple H I J. Okay, tells you how the H I J V pair of elements such that C has a unity property. I and D has unity property. J can realize the binary property. V. Okay, I will formalize later. What are the Unitarian binary properties and how this counting works? But right now I just want you to get an intuition of what we're going to talk about. So the red and the green multi normal coefficients are just enumerating all the models over the judiciary and the binary properties. The N I J V select the models which are satisfy a ble which which satisfy the formula phi X. Y. Okay. So N I J V is an indicator variable, which is zero or one? Depending on whether the the pair cds satisfies the binary property's entailed by five X. Y. Okay. And K Ij here are just the possible ways in which you can combine the juniper tree, the elements which have the I I attended property and the elements which have the guillotine europe the way you can combine them. Okay, so let's look deeper into it. So let me first formalize what is A unity property, a unity property formally. And logic is a one type and a one type is a conjunction of maximally consistent set of literal, which have only one variables. So for example, in the example previously introduced of the graphs, you have only one binary predicated are Okay, so the one types are not Rx sx and Rx sex. So there are 21 types,

not our xx and our access. And they're maximally consistent because you cannot add any atom any item with only one variable to them. Such that they are still consistent. So they are maximally consistent. Okay, We see a certain domain element realizes of one type if for that domain element, the one type is true. So for example, if you have the graph and a certain node does not have a self loop, then it realizes the one type, alpha one. It has not our xx. Okay.

Ah moving on. So how do I do enumeration over the one types. Okay, so the key idea here is that in a given model? Okay, a certain domain element can only realize one of the possible one types. Okay, so for example, if you have a graph, okay then a certain node

can have a self loop or not have a self loop. They are mutually exclusive choices and each node has to have that. All right. So, so if I want to count the number of graphs such that you have given notes which have self loops and K two notes rich. So the cable notes which do not have self loops and K two notes which have self loops. Then that number is given by this expression here. Okay, so there's a slight added in the slides. Caven notes which do not have self loops and K two notes which have self loops. So this idea can be generalized of enumerating although unity properties and the possible ways that K. I. Consul for each unit property alpha X. If you have K I constants Then the possible such enumeration. The possible ways of realizing these unity properties is given by equation six. So basically question sexism multi normal coefficient. So all the K II some to end. So what you have in the numerator is in factorial. The number of ordering of the of all the domain elements but you have to divide by K I factorial for each property I because you do not want to uh count more? The exchange ability? The exchangeable elements. Okay, so if two elements let's say are both red. So if you have a colored graph and two notes are both read, you do not want to count their exchange as a different enumeration. Right? So this is what this expression counts. The possible ways of realizing the january properties. Now, What about binary properties? Okay, so in logic, binary properties are formalized as two tables. So two tables. Our conjunction of maximally consistent set of liberals With exactly two variables.

Uh So here, in a photo language I introduced earlier we have only one binary predicate are so the maximally consistent set of liberals are given here so that the two tables are given by beta one, beta two and beta food. These are the only possible two tables. And again, the key idea here is that in a given model,

uh

a pair of domain elements. Okay, given that they have some unity properties can only realize one binary property, so they cannot have more than one binary property. Okay, so let's see how such an enumeration can work. So let's say you have K one elements which have the first unity property and K two elements which have the second unit property or they satisfy alpha one X. The one type and they satisfy the K two elements to satisfy alpha two works to one type. And now let's say I want to count all possible models where where I have H 123 pairs of constraints. Is that each of these a paired satisfied the Viet binary property. Okay, so I have basically this this V index goes over four possible binary properties. Right? So, basically And this enumeration is given by the situation here. So cave and K two are the possible pairings that you can make between the unity properties. Okay. And given these unity properties, you want to distribute the unity the unity pairings to the two binary properties. All right. And that is given by the expression on the right. So you have possible pairings and again you do not want to overcount for the exchanges. So hence you divide by the number of pairs which relies the one ordered towards all the three or the 4th two tables. So this idea again can be generalist. And now we have the way a way of enumerating all possible models Over January in binary properties or over one types and two tables. So the red multi normal coefficients allows us to enumerate all the possible realizations off The one types and given the enumeration of the one types. You pick a pair. Each pair of the unity properties or the one types. And given this pair, you you and you enumerate the possible binary assignments of this pair, which is given by the green multi normal coefficient. Okay, so K IJ is basically telling us the K I G expression here is basically telling us what are the possible ways of the pairings. Okay

now moving on. What how does the formula impact this intimidation? And that's what we will discuss now. So now we have we know how to enumerate all the models as you know in binary properties but you already know this number. But we want so we want to be selective about the models. You want to select only those models which are models off for our next for all right. Fire. Fx Y. Okay. And this is done by introducing an indicator variable V which is one if and only if. Okay. The conjunction of the unity and the binary property is the one who has to test and the two tables and this this expression Okay, intuitively what it means. Is that the formula for all X. For all by this this force urologic formula allows us to admit only certain types of properties. It discards the other ones. Okay. And when it discards the other one we warned the N I. J. So when when when an assaulting model those discarded properties are realized we want that model to be thrown away to have zero weight. Okay.

And this is expressed an equation aid. So basically all these things combined together. Give us this formula. So you first and you married in the red multi normal coefficient over the unity properties. Given this an immigration You enumerate over the binary properties. Okay. And now that you have all possible enumeration, you check whether assaulting enumeration is consistent with your formula or not. And that is given by the N I J V. So whenever you're enumeration is not consistent with the formula U n I J V will have some um there will be some N I J V. Of in that enumeration which is zero and hence the contribution of the entire model or that an entire setting will go to zero. And when they are all consistent when all the properties are consistent with the model, all the any Jvs will be one.

Moving on now we will talk about cardinality constraints. So what are cardinality constraints? So cardinality constraints. Our constraints on the number of times assault and eradicate can be realized in the model. So for example if you have a simple labeled graph. Okay. So like the previous one I introduced but something simple means that it cannot have a self loop.

And now you want to say I only only want the graphs with exactly mm edges. This can be expressed as a cardinality constraints

given here. So what you do is you just say that the predicate are has a cardinality to women. You will automatically have only those models where the correspondent where the corresponding graph is exactly M edges.

Okay, so the way you can compute

the fossil model counting with cardinality constraint is by introducing linear arithmetic constraints on K one K two and H I J V. S. Okay. And one example is given in the red equation hill

moving forward.

We will talk about freight and model counting. So so far we have talked about just model counting then. How do you introduce weights? And it's a pretty simple idea. So you you see that vision model counting has a specific goes over the kh the possible instance ations of K. S. And the edge vectors. Okay. And I express it as this in this abbreviated form of F K H N I J V. What I can do is I can write an arbitrary weight function which goes from each of the institution of the que and the H vectors to a positive numbers are okay and hence and then I can associate this positive number as a multiplication way to the F K H. To get the rated for shoulder model counting. Okay, so this weight function is quite expressive and actually this weight function is strictly more expressive than symmetric weight functions which are just multiplication waits for the Medicaid's. And this new class of weight functions can be expressed with just real numbers. Whereas previously we needed to use complex numbers too express these weights.

I will finish my condition here. Thanks a lot.

Thank you, cigar. Hello. Hello. Hi. So I know that there might be some questions on the chat. I will wait for seeing them. There might be a small delay. Otherwise I have a couple. I can start then I will give the floor to the audience. I had the chance to have a look at your paper and I understood that your formulation could be useful for probabilistic inference in statistical relation learning. Could you spend one word on this?

Alright, okay. So basically

ah basically the probabilistic inference framework,

all all vision model countings. All probabilistic inference actually in Beijing networks as well can be reduced to vision model counting. And this was works of darvish Along in 2003 I believe. And the first order logic setup allows us to go to statistical relation learning specifically where you have first order logic formulas. So what you do is that you can actually uh express mark of logic network distributions using Reddit for shoulder model counting. And uh I cannot I have not written the distribution the reduction, so I cannot explain it so easily. But let's say you have a formula F of X. Y. And microblogging network with the weight W. Okay. So what you can do is you can introduce a new predicate P of Xy and you're right B of X. Y. If and only if F of X Y. Okay. And the weight of P of xy when it occurs positively Is exponential of W&1 otherwise. And the weight of all other predicates is one. So the weighted model counting of this uh formula Okay, will correspond exactly to the distribution by the market logic network. So Okay I'm sorry just so this you see this expression here the equation do Okay. No I do not.

Okay. Yes. So the term on the numerator is just the weight of the model and the term in the denominator is just a sermon. The weighted model counting that I just explained to you. I'm sorry, I should have written that maybe this reduction.

I'm not an expert of the topic. So it was just a curiosity. You aren't scared even without going into the technicalities and if there are no questions from the audience it seems that they are not. Maybe this is the really most knife question I could do if you found this polynomial approach but I was wondering whether you did implement some of the techniques that you just discussed. The neurosis rights, okay. Yes. Uh and uh they usually Okay, so I should, I'm sorry, I should have said earlier, the polynomial techniques exist by given and drug as well already. So they use some sort of let's say DbN NFS but for sort of logic variant of BDNF to which are some uh probably sorry, the dead diagrams for the generate the trees for the model counting what we our contribution is having a probabilistic frame or having a combinatorial framework for this, having formulas for this basically. So our contribution is somewhere between a I and community audits. So we have closed form formulas and anyone can just look at them and implement and we have implemented many and there do work efficiently, but I would say that maybe DDN NF sometimes beat our formulas, so our formulas and more about let's say theoretical clarity. Okay, okay. This is a beautiful to understand as the motivation for your approach. Okay.

Just to finish, sorry, a much extended work of this. This line has been accepted triple Ai. Today we got the acceptance. I hope you read that as well. Thanks a lot. Okay, Thanks a lot. And let me thank the speaker and we can