Towards a Neurosymbolic Approach to the Weighted model Counting

Adriel Santos, Luca Naja, Matheus Uchôa, Edjard Mota, Son Tran, Artur Garcez
{avs, ldn, matheus.uchoa, edjard}@icomp.ufam.edu.br, son.tran@deakin.edu.au,a.garcez@city.ac.uk

Introduction

Our research delves into the Weighted Model Counting (WMC) problem, specifically exploring its connection with Logical Boltzmann Machines (LBM). While WMC extends the Boolean Satisfiability Problem (SAT) by quantifying and weighing possible truth-value assignments, our focus is on integrating LBM's neurosymbolic capabilities to better understand these probabilities. We introduce a novel concept of "reverse intent" in WMC, investigating how individual characteristics can influence the truth value of clauses in a Boolean formula. This approach, still in its early stages, aims to blend logic, machine learning, and probability theory to offer new insights into the complexities of WMC.

Related Work

This article extends the foundational work in theoretical computer science and logic, particularly focusing on the Boolean Satisfiability Problem (SAT) and the Weighted Model Counting (WMC) problem. It builds upon previous research, notably referenced in [1], by applying Bayesian methods to explore these challenges. The paper emphasizes the integration of Logical Boltzmann Machines (LBM) with SAT and WMC, linking abstract concepts with practical applications. A key aspect of this research is the shift from traditional clause-based analysis to an individual-focused approach, examining how unique characteristics of individuals influence the truth value of clauses in a Boolean formula. This approach represents an innovative expansion of the core concepts introduced in [1].

Conclusion

,

References

[1] S. N. Tran and A. d'Avila Garcez. Logical boltzmann machines for logical reasoning. 2021.

[2] Sang, Beame, and Kautz. Performing bayesian inference by weighted model counting. 2021.

[3] Salakhutdinov R. R. Hinton, G. E. 'deep boltzmann machines'. in proceedings of the twelfth international conference on artificial intelligence and statistics (aistats). 2009.

Acknowledgements

We thank UFAM for partially supporting this work.

Background and Theoretical Foundations

Boltzmann Machines are key stochastic neural networks in deep learning, linking statistical mechanics with information theory. They model complex distributions through symmetrically connected neurons representing binary states.

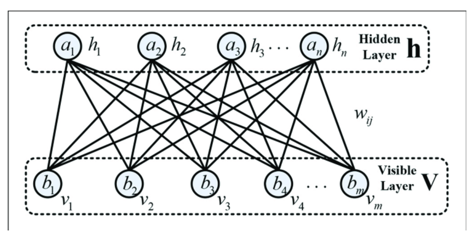

Figure 1: Basic structure of a RBM. Source: A Decoding Scheme for Incomplete Motor Imagery EEG With Deep Belief Network. Year: 2018.

- **Standard Boltzmann Machines:** The original, fully connected stochastic binary unit network, foundational to neural network research.
- **Restricted Boltzmann Machines (RBMs):** Two-layered (visible and hidden layers), no intra-layer connections. Key in deep learning for feature learning, dimensionality reduction, and classification.
- **Deep Boltzmann Machines (DBMs):** Multi-layered extension of RBMs for complex data representation and deeper learning, crucial in advanced deep learning.

Proposed Method and Application

Introduction to the Problem:
The Nixon Diamond Problem illustrates challenges in logical and probabilistic reasoning due to Nixon's conflicting identities as a Republican and a Quaker. Its resolution using Logical Boltzmann Machines highlights the effectiveness of these machines in handling complex logical structures and probabilistic scenarios, particularly in reconciling conflicting attributes in artificial intelligence systems. As demonstrated in [1], we can assign weights to each clause as shown. The following table presents the statements accompanied by each statement clause and an associated weight.

Statement	Clause	Weight
Nixon is a Republican	$n \to r$	1000
Nixon is a Quaker	$n \to q$	1000
Republicans not Pacifists	$r \to \neg p$	10
Quakers are Pacifists	$q \to p$	10

Conversion to SDNFs: Applying LBMs to the Nixon Diamond Problem involves converting weighted statements into Strict Disjunctive Normal Forms (SDNFs), where statements are expressed as ORs of AND clauses. This simplifies logical relationships for computational processing. As exemplified in [1], the SDFN are represented as, for example, n → r 1000 : $(n \wedge r) \vee (\neg n)$.
Building the RBM: This process involves using the unique conjunctive clauses and their corresponding weights derived from the SDNF conversion. Each clause, along with its weight, forms a part of the RBM's structure, serving as inputs to the machine. The RBM utilizes these inputs to analyze and infer the complex interrelationships between the different beliefs and statements. The RBM for this problem includes the following clauses with their weights:

Clause	Weight
$n \wedge r$	1000
$\neg n$	2000
$n \wedge q$	1000
$r \wedge \neg p$	10
$\neg r$	10
$q \wedge p$	10
$\neg q$	10

Weighting and Normalization: After converting the clauses, we can move towards introducing weights to the variables of the problem, progressively aligning it more closely with the Weighted Model Counting (WMC) problem. While the RBM assigns weights to clauses, we aim to correlate the weight of each clause with the individual weights of its elements through an empirical analysis. This approach seeks to deduce which elements are most influential and carry more weight in a decision-making process of an LBM, for instance. Considering that all other variables in the model are treated as logical conjunctions, which can be represented by multiplications based on the truth table principles of logical conjunctions, it's feasible to approach this as a problem of solving a linear system. In this context, the goal is to find values for the variables that satisfy the given sentences or clauses. Each clause, along with its associated weight, can be viewed as an equation in this linear system. The weights 2000 for ¬n, 10 for ¬r, and 10 for ¬q represent specific values that the equations must satisfy. The equations of this linear system is $n \times r = 1000; n \times q = 1000; r \times \neg p = 10; q \times p = 10$. Therefore, a possible solution is:

Variable	Value
n	100
r	10
q	1
p	10
$\neg p$	1

Probabilistic Analysis and Model Formulation: After obtaining the values of clauses and variables, we can further evolve our analysis and identify the probability relationships between the presented clauses. There are several ways to accomplish this, but given the values we have constructed in this example, we applied the technique of Normalization to probability distribution. Essentially, this ensures that all elements are positive and their sum equals 1, adhering to the properties of a probability distribution.

Clause	Weight
$n \wedge r$	0.24752475
$\neg n$	0.4950495
$n \wedge q$	0.24752475
$r \wedge \neg p$	0.00247525
$\neg r$	0.00247525
$q \wedge p$	0.00247525
$\neg q$	0.00247525

We can also apply this normalization technique to individual variables to achieve a probability distribution. 0.24752475: n; 0.4950495: r; 0.24752475: q; 0.00247525: p; 0.00247525: $\neg p$.
Based on the approach we've discussed, We see great promise in correlating a statistical method with a Logical Boltzmann Machine (LBM). This initiative involves assigning probabilistic values to logical statements, which significantly enriches the analysis of their impact within the LBM framework.