Approximation Algorithms for Max-Sum-Product Problems

Denis D. Mauá
Istituto "Dalle Molle" di Studi sul'Intelligenza Artificiale (IDSIA)
Switzerland
denis@idsia.ch

Abstract

Many tasks in probabilistic reasoning can be cast as max-sum-product problems, a hard class of combinatorial problems. We describe our results in obtaining a new approximation scheme for the problem, that can be turned into an anytime procedure. For many tasks, this scheme can be shown to be asymptotically the best possible heuristic.

1 Introduction

Many tasks in probabilistic reasoning can be cast as the computation of a max-sum-product of finite discrete functions, that is, a maximization of sum-marginals of a product of functions over finite domains. A prominent example is the selection of a *maximum a posteriori* (MAP) configuration in a discrete probabilistic graphical model [Mauá and de Campos, 2012]. We showed elsewhere that other tasks can also be cast in this framework, such as planning with (limited memory) influence diagrams [Mauá *et al.*, 2012a], and multiway sensitivity analysis of Bayesian networks [Mauá *et al.*, 2012b].

The max-sum-product problem inherits its computational hardness from MAP inference in discrete graphical models, which is NP-hard even if the correspnding domain graph is a tree and variables take on a bounded number of values; it is also inapproximable even when the treewidth of the domain graph is bounded [de Campos, 2011].

Park and Darwiche [2003] tackled the problem by systematically searching the space of solutions, using more easily computed upper bounds to potentially avoid an exhaustive search. Their algorithm however guarantees neither efficiency nor accuracy; despite these drawbacks, it is often competitive.

An alternative approach is to solve the problem in two stages, by first solving the sum-product part of the problem for every value of the maximization variables, and then solving the remaining max-product problem. Usually, both stages are solved by variable elimination. Since arbitrarily large functions might be created during variable elimination (in either stage), function approximations are necessary. Decther and Rish [2003] applied the mini-bucket scheme to approximate large functions created during a variable elimination solution of either the sum-product and the max-sum problems. While the complexity of the procedure can be controlled by the user, the final accuracy can be arbitrarily poor. Meek and Wexler [2011] generalized the mini-bucket scheme to arbitrary Multiplicative Approximation Schemes (MAS). Their approach also consists in approximating the functions generated during the variable elimination routine. While all the mentioned approaches are able to produce upper and lower bounds on the max-sum-product, they are unable to estimate beforehand the resources required to achieve a given accuracy. A different class of algorithms optimizes a surrogate global function by means of message passing in the underlying graph. Liu and Ihler [2011] devised a belief-propagation like algorithm that optimized an upper bound on the value of the max-sum-product. Jiang et al. [2011] proposed a similar approach, with different message functions and without the guarantee of upper bounding the true value. These approaches have polynomial time complexity, but are unable of refining the solution.

In this ongoing work we investigate multiplicative approximation schemes for the computation of max-sum-products such that (i) given a limited amount of computational resources (i.e., time and memory) finds bounds for the solution which are in the worst case independent of the input, and (ii) find a solution of a given accuracy in finite (but possibly exponential) time. Thus, the key feature of our schemes is to allow a trade-off between efficiency and accuracy. We show that for problems where the domain graph has bounded treewidth and variables take on a bounded number of values, the problem is approximable, and the scheme provides a fully polynomial time approximation scheme (FPTAS) that guarantees both accuracy (provided by the user) and efficiency (i.e., polynomial running time) [Mauá *et al.*, 2012b; 2012c; 2012b]. To our knowledge, this is the first result of a FPTAS in discrete probabilistic reasoning when the task is NP-hard. For problems with unbounded treewidth or arbitrarily large variable space cardinality, the schemes can provide either provably good approximations with exponential worst-case running time, or efficient approximation with arbitrarily poor accuracy. Experiments however show that often the scheme is able to find accurate solutions in feasible time [Mauá and de Campos, 2011; Mauá *et al.*, 2012a]. Furthermore, the solutions can be refined in an anytime fashion, that is, the accuracy of the algorithms improves monotonically as more time and memory are made available, and converges eventually to the global optimum [Mauá and de Campos, 2012].

2 Problem Statement

Given finite sets $X_1, \ldots, X_n$, an integer m smaller than n, and a collection of non-negative real-valued functions $\{f_\alpha : \alpha \in I\}$ over $X \stackrel{\text{def}}{=} X_1 \times \cdots \times X_n$, solve

$$\max_y \sum_z \prod_\alpha f_\alpha[y, z] \,,$$

where the maximization is over $X_1 \times \cdots \times X_m$ and the sum is over $X_{m+1} \times \cdots \times X_n$. The functions f_α are assumed to be defined locally, meaning that for each α there is a subset $S_\alpha \subseteq N \stackrel{\text{def}}{=} \{1, \ldots, n\}$ such that $f_\alpha[x] = f_\alpha[x']$ whenever x and x' agree on the coordinates in S_α. The *domain graph* of a collection of functions $\{f_\alpha\}$ is the graph $G = (N, E)$, where there is an edge $\{i, j\}$ in E if $\{i, j\} \subseteq S_\alpha$ for some α. The *treewidth* of the domain graph is the width of its minimum-width tree decomposition.

A *multiplicative approximation scheme* (MAS) for the max-sum-product problem is a family of algorithms $\{A_\epsilon : 0 < \epsilon < 1\}$ where each algorithm A_ϵ returns a solution y_ϵ such that

$$\max_y \sum_z \prod_\alpha f_\alpha[y, z] \leq (1 + \epsilon) \sum_z \prod_\alpha f_\alpha[y_\epsilon, z] \,.$$

A MAS is a *fully polynomial-time approximation scheme* (FPTAS) when each algorithm A_ϵ in the family runs in time polynomial in the size of the input (given in number of bits) and in $1/\epsilon$.

An anytime algorithm provides at each time t a solution y_t such that for $t' < t$ the the value of the objective function at $y_{t'}$ is no greater than the value of the objective function at y_t, and there is t such that y_t is an optimal solution. In other words, an anytime procedure provides a solution whose quality improves monotonically with time, and eventually converges to the optimal solution.

3 Contributions

1. We showed that MAP inference in discrete graphical models, selection of optimal strategies in (limited memory) influence diagrams, classification with ensemble of graphical models, and multiway sensitivity analysis in Bayesian networks can all be cast as a max-sum-product problem.
2. We devised a new algorithm that solves the max-sum-product exactly. Although the algorithm has exponential worst-case time complexity, we show in a testset of MAP inference, planning, and sensitivity analysis problems that it often runs in feasible time, and outperforms other methods.
3. We devised a new MAS for the max-sum-product. Given a desired accuracy, the algorithm estimates (before the computations) a worst-case bound on the amount of computational resources needed. Conversely, given a limit on the amount of computational resources, the algorithm estimates (beforehand) the worst-case accuracy of the algorithm. Experiments show that the estimate bounds are often loose, and the algorithm finds accurate solutions with few computational resources. The scheme can be turned into an anytime algorithm.
4. We proved that the MAS scheme is a FPTAS if the domain graph of the problem has bounded treewidth and all variables take on a bounded number of variables.
5. We showed that for planning with influence diagrams and multiway sensitivity analysis of Bayesian networks, there can be not FPTAS if either variables take on arbitrarily many values or the treewidth of the domain graph is unbounded. This was already known for the case of MAP inference in graphical models, so our FPTAS is the best heuristic for these problems (asymptotically).

Acknowledgments

This work was partially supported by the Swiss NSF grant no. 200020_134759/1.

References

[de Campos, 2011] C.P. de Campos. New complexity results for MAP in Bayesian networks. In *IJCAI 2011*, pages 2100–2106, 2011.

[Dechter and Rish, 2003] Rina Dechter and Irina Rish. Mini-buckets: A general scheme for bounded inference. *Journal of the ACM*, 50(2):107–153, 2003.

[Jiang *et al.*, 2011] J. Jiang, P. Rai, and H. Daume III. Message-passing for approximate MAP inference with latent variables. In *NIPS '11*, pages 1197–1205. 2011.

[Liu and Ihler, 2011] Q. Liu and A. Ihler. Variational algorithms for marginal MAP. In *UAI '11*, pages 453–462, 2011.

[Mauá and de Campos, 2011] D. D. Mauá and Cassio P. de Campos. Solving decision problems with limited information. In *NIPS '11*, pages 603–611. 2011.

[Mauá and de Campos, 2012] Denis Deratani Mauá and Cassio Polpo de Campos. Anytime marginal map inference. In *ICML '12*, 2012.

[Mauá *et al.*, 2012a] D. D. Mauá, C. P. de Campos, and M. Zaffalon. Solving limited memory influence diagrams. *Journal of Artificial Intelligence Research*, 44:97–140, 2012.

[Mauá *et al.*, 2012b] Denis D. Mauá, Cassio P. de Campos, and Marco Zaffalon. Updating credal networks is approximable in polynomial time. *International Journal of Approximate Reasoning*, 53(8):1183–1199, 2012.

[Mauá *et al.*, 2012c] Denis Deratani Mauá, Cassio Polpo de Campos, and Marco Zaffalon. The complexity of approximately solving influence diagrams. In *UAI '12*, pages 604–613, 2012.

[Meek and Wexler, 2011] C. Meek and Y. Wexler. Approximating max-sum-product problems using multiplicative error bounds. *Bayesian Statistics*, (9):439–472, 2011.

[Park and Darwiche, 2003] J. D. Park and A. Darwiche. Solving MAP exactly using systematic search. In *UAI '03*, pages 459–468, 2003.