Evaluation of DNF Formulas

Piotr Wojciechowski
LCSEE,
West Virginia University,
Morgantown, WV
{pwojciec@mix.wvu.edu}

Zola Donovan
Department of Mathematics,
West Virginia University,
Morgantown, WV
{zdonovan@mix.wvu.edu}

Abstract

This paper examines the Stochastic Boolean Function Evaluation (SBFE) problem for classes of DNF formulas. Although the SBFE problem is inapproximable for general DNF formulas, exact and approximate algorithms for monotone k-DNF and monotone k-term DNF formulas are presented. The SBFE problem for DNF formulas involves sequential tests to determine the value a DNF formula on an initially unknown input, $\mathbf{x} = (x_1, ..., x_n)$, where cost, c_i, associated with obtaining the value of independent bits, x_i, is given; and the probability, p_i, that each $x_i = 1$ is known. The goal is to minimize expected cost. Also presented is a proof on a lower bound result for evaluation of monotone CDNF formulas.

1 Introduction

Stochastic Boolean Function Evaluation (SBFE) is the problem of determining the value of a given Boolean function, ϕ, on an unknown input, $\mathbf{x}$, such that each bit x_i of $\mathbf{x}$ can only be determined by paying an associated cost, c_i. The unknown input, $\mathbf{x}$, is randomly generated such that the bits are independent, and the probability that $x_i = 1$, denoted p_i, is known. The goal is to minimize the expected cost of evaluation.

This paper studies the complexity of the SBFE problem, and considers both exact and approximate versions of the problem for k-DNF and k-term DNF formulas. It is known that the general SBFE problem is NP-hard for arbitrary DNF formulas since satisfiability is NP-hard [Gre06]. To show that the SBFE problem for k-DNF is NP-hard, even for $k = 2$, a simple reduction is used. A $\frac{4}{\rho}^k$ factor algorithm for evaluating monotone k-DNF is presented, along with a proof showing that the SBFE problem for monotone k-term DNF is solvable in polynomial time for some constant k.

Additionally, an approximation algorithm solving the SBFE problem for CDNF formulas (and decision trees) for the special case of unit costs, the uniform distribution, and monotone CDNF formulas is given [Kap05]. For k terms of the DNF, and d clauses in the CNF, the algorithm achieves an approximation guarantee of $O(\log kd)$ on the value of the expected certificate cost–a lower bound on the optimal solution. The algorithm alternates between two processes; one process attempts to achieve a 0-certificate, while the other attempts to achieve a 1-certificate. This "round robin" technique is modified to handle arbitrary costs with no change in the approximation factor, as the original algorithm handles only unit costs.

Finally, it is shown that the approximation guarantee of $O(\log kd)$, is close to optimal. The approximation factor must be $\Omega((\log kd)^{\varepsilon})$, for $0 < \varepsilon < 1$; which also implies that the (optimal) average depth of a decision tree computing a Boolean function can be exponentially larger than the average certificate size for that function, while the depth complexity of a decision tree for a function (worst-case) is at most quadratic [Buh02].

2 Statement of Problem

The Stochastic Boolean Function Evaluation (SBFE) problem is defined as follows. We are given a boolen formula $\phi(x_1, \ldots, x_n)$, the costs $c_1, \ldots, c_n \geq 0$ of determining the value of each x_i, and the probabilities $0 < p_1, \ldots, p_n < 1$ that each x_i is **True**. The goal of the problem is to determine, at minimum cost, the value of $\phi(\mathbf{x})$ for an unknown value

of $\mathbf{x}$. This $\mathbf{x}$ is randomly generated so that each x_i is independent and $P[x_i = \mathbf{True}] = p_i$. While $\mathbf{x}$ remains initially unknown the value of each x_i value can be revealed at a cost of c_i.

The order in which variables are revealed can vary depending on the values of the already revealed variables. The revealed variables are enough to determine the value of $\phi(\mathbf{x})$. Thus we have that for every $\mathbf{y}$ which agrees with $\mathbf{x}$ on the revealed variables, $\phi(\mathbf{y}) = \phi(\mathbf{x})$. This automatically leads us to the inapproximability of the general case. For example if $\phi(\mathbf{x})$ is an unsatisfiable CNF formula or a universal DNF formula then optimally no queries would be necessary. This is because in these cases the value of ϕ is constant over all values of $\mathbf{x}$. Any constant factor approximation has to yield the exact result in these cases and thus solve either an NP or coNP complete problem.

3 Motivation and Related Work

Stochastic Boolean Function Evaluation is applicable in the field of medicine. In this case the x_is correspond to medical tests performed on a given patient, and the boolean function $\phi(\mathbf{x})$ evaluates to **True** is the patient has a certain disease. SBFE can also be applied to query optimization in databases, where all stored values of $\mathbf{x}$ that satisfy $\phi(\mathbf{x})$ need to be located.

Certain special cases of the SBFE problem can be solved exactly in polynomial time. These include including read-once DNF formulas and k-of-n formulas [Ünl04]. An approximation factor of n can be obtained for arbitrary boolean formulas by testing the variables in increasing order of their costs. [Kap05]

Deshpande et al. explored a generic approach to developing approximation algorithms for SBFE problems, called the Q-value approach. It involves reducing the problem to an instance of Stochastic Submodular Set Cover. They proved that the Q-value approach does not yield a sublinear approximation bound for evaluating k-DNF formulas, even for $k = 2$. They also developed a new algorithm for solving Stochastic Submodular Set Cover, called Adaptive Dual Greedy, and used it to obtain a 3-approximation algorithm solving the SBFE problem for linear threshold formulas. [Des13]

The main application of the problem is in the optimization of database queries. In this we are in which databases a query ϕ evaluates to true. [Ibra84]

4 Critique

The introduction thoroughly outlines the contents of the paper. Included is a statement of the SBFE problem, and the goal of the problem. There is also a mention of areas like Operations Research and Learning Theory in which the SBFE problem is often studied. It is made clear that the focus of the paper is on the complexity of the SBFE problem for specific classes of DNF formulas with details about those classes, algorithms, and proofs that will be presented. Following is a discussion about how the Kaplan et al. algorithm is modified to obtain a near optimal approximation bound. Additionally, a statement referring readers to the omitted proofs of the results is also included in the introduction.

The second section, appropriately titled "Stochastic Boolean Function Evaluation," restates the SBFE problem by offering a more formal definition. Also given in this section, is a sufficient outline of a worst-case running time algorithm for the problem. Since the SBFE problem arises in many different application areas, it was nice to see a brief mention of two areas of application including medical diagnosis and query optimization. Also noted were a few algorithms which solve the SBFE problem exactly for a small number of classes of Boolean formulas, as well as the generic approach explored by Deshpande et al. to develop approximation algorithms for this problem.

The Preliminaries section succinctly defines many of the terms relevant to the SBFE problem, including but not limited to: *literal, term, clause, size* of a term or clause, *DNF, CNF, k*-term DNF, k-DNF, *size* of a DNF (CNF) formula, etc. This section is helpful, particularly for those not familiar with the SBFE problem. They also state the set cover problem, which is relevant given the reduction used to show that the SBFE problem for k-DNF and k-term DNF is NP-hard.

Next is a discussion of the hardness of the exact SBFE problem, given in the section "Hardness of the SBFE problem for monotone DNF". [Gre06] is cited for showing how the SBFE problem for CNF formulas is NP-hard. It is subsequently shown that if $P \neq NP$, the SBFE problem for DNF cannot be approximated to within any factor $\rho > 1$. Finally, this section presents an approach used by [Cox89] to show that the SBFE problem for DNF is still NP-hard whether or not the DNF is

monotone. While it is stated that Cox reduced from Knapsack, and the reduction presented is from Vertex-Cover, neither the Knapsack problem nor Vertex-Cover problem is stated or defined in the Preliminaries section.

The fifth section of the paper introduces the approximation algorithms used for the SPFE problem in the case of monotone k-DNF and k-term DNF formulas. This is achieved by alternating between two separate algorithms, one which tests for 1-certificates and the other which tests for 0-certificates. No pseudocode for these algorithms is given, nor is any given for the alternation procedure. Instead the paper provides descriptions of each of the necessary procedures.

The next section describes an exact polynomial time algorithm for the SBFE problem when restricted to k-term DNF. Similarly to the preceding section no pseudocode is provided and only outlines are given. The algorithm generalizes the work of [Gre06] general monotone DNF formulas instead of just those formulas with read-once refutations.

The final section of the paper describes the difference between the two measures of approximation bounds for the SBFE problem. The first of these is the optimal expected cost that can be achieved by a particular strategy while the other is the minimum expected evaluation cost of a given formula. The paper shows that this ration can be extremely large even when both evaluation cost and probability are uniform.

5 Conclusions

Thorough reduction to tautology of DNF formulas the general case of the SBFE problem is inapproximable unless $\mathbf{P} = \mathbf{NP}$. However when dealing with monotone k-DNF formulas the SBFE problem can be approximated in polynomial time with an approximation bound of $\frac{4}{\rho}^k$. Monotone k-term DNF formulas also have a polynomial time approximation algorithm, however, this algorithm has an approximation factor of $\max\{2 \cdot k, \frac{2}{\rho} \cdot (1 + \ln k)\}$. While the general cases of the SBFE problem for CNF and DNF formulas are inapproximable for constant factor approximations this result stems from the case when the optimal solution to the SBFE problem is 0. This situation might be overcome if we look at an $\alpha \cdot OPT + c$ approximation. Thus focusing of this type of approximation may generate approximations for more general forms of the SPFE problem.

References

[All13] Allen, S.; Hellerstein, L.; Kletenik, D.; and Ünlüyurt, T. 2013. Evaluation of DNF formulas. http://arxiv.org/abs/1310.3673

[Buh02] Buhrman, H., and Wolf, R. D. 1999. Complexity measures and decision tree complexity: A survey. *Theoretical Computer Science* 288:2002.

[Des13] Deshpande, A.; Hellerstein, L.; and Kletenik, D. 2013. Approximation algorithms for stochastic boolean function evaluation and stochastic submodular set cover. http://arxiv.org/abs/1303.0726

[Cox89] Cox, L.; Qiu, Y.; and Kuehner, W. 1989. Heuristic least-cost computation of discrete classification functions with uncertain argument values. *Annals of Operations Research* 21:129.

[Gre06] Greiner, R.; Hayward, R.; Jankowska, M.; and Molloy, M. 2006. Finding optimal satisficing strategies for and-or trees. *Artif. Intell.* 170(1):19-58.

[Ibra84] Ibaraki, T.; and Kameda, T. 1984. On the optimal nesting order for computing n-relational joins. *ACM Trans. Database Syst.* 9(3):482502.

[Kap05] Kaplan, H.; Kushilevitz, E.; and Mansour, Y. 2005. Learning with attribute costs. *STOC*, 356365.

[Ünl04] Ünlüyurt, T. 2004. Sequential testing of complex systems: a review. *Discrete Applied Mathematics* 142(1-3):189205.