Outside Computation with Superior Functions

Parker Riley and **Daniel Gildea**
Department of Computer Science
University of Rochester
Rochester, NY 14627

Abstract

We show that a general algorithm for efficient computation of outside values under the minimum of superior functions framework proposed by Knuth (1977) would yield a subexponential time algorithm for SAT, violating the Strong Exponential Time Hypothesis (SETH).

1 Introduction

Weighted deduction systems are used in a number of NLP applications, including parsing for context-free grammars (Shieber et al., 1995; Sikkel, 1997; Nederhof, 2003; Eisner and Satta, 1999) and machine translation (Melamed et al., 2004; Lopez, 2009). In these applications, the inside-outside algorithm enables efficient calculation of the total weight of all derivations passing through a specific item in the weighted deduction system by computing tables of inside and outside values. Goodman (1999) develops a generalized inside-outside algorithm that can be used with any commutative semiring. Applying the sum-product semiring results in the standard inside-outside algorithm used as a subroutine in Expectation Maximization (Dempster et al., 1977). Applying the max-product semiring results in an efficient algorithm for finding, for example, the best tree that incorporates a specified constituent covering a specified span of the input string.

The minimum of superior functions framework of Knuth (1977) is an alternative to the semiring framework for analyzing weighted deduction systems. Knuth's framework is more general than semirings in that it allows more general functions to be used for combining the weights of subderivations. Knuth's framework has the advantage that it allows for best-first search with a generalization of Dijkstra's algorithm, as well as for A* search (Nederhof, 2003).

Given that Knuth's framework guarantees efficient inside computation, does it also guarantee

$$
\begin{array}{rl}
w_1\colon & [A, i, j] \\
w_2\colon & [B, j, k] \\
\hline
F_R(w_1, w_2)\colon & [S, i, k]
\end{array}
$$

Figure 1: A rule R for CFG parsing in weighted deduction notation for production $S \rightarrow A\,B$. The goal item for CFG parsing with start symbol S and sentence length n is $[S, 0, n]$.

efficient outside computation, allowing for a generalized inside-outside algorithm? In this paper, we answer this question in the negative. We prove that a general algorithm for efficient outside computation in this framework would imply the existence of a subexponential time algorithm for satisfiability of boolean formulas in conjunctive normal form (SAT), violating the Strong Exponential Time Hypothesis (SETH) (Impagliazzo and Paturi, 1999) which postulates that no such algorithms exist. This result may be counterintuitive, because one might expect efficient outside computation to be possible whenever efficient inside computation is possible. We believe this result to be the first formal hardness proof for outside computation in weighted deduction systems.

2 Background

A weighted deduction system (Nederhof, 2003) has rules of the form $\frac{X_1, \ldots, X_n}{Y}$ where $X_1, \ldots, X_n$ are items forming the antecedents of the rule and item Y is the consequent of the rule. A derivation of item X is a tree of rules where the antecedents of each rule are the consequents of its children, and X is the consequent of the root of the tree. The leaves of this tree are rules with zero antecedents, called axioms. Each rule R is also associated with a rule weight function F_R which takes as input the weights of the antecedents and calculates a new weight for the consequent. The weight of a derivation is the weight of the rule serving as the root of the tree, calculated by recursively evaluating the

rule weight functions F_R; that is, for a derivation D formed by applying rule R to antecedent derivations $D_1, ..., D_n$:

$$\text{weight}(D) = F_R(\text{weight}(D_1), ..., \text{weight}(D_n))$$

To show both the rule and the weights of the antecedents and consequent, we use a notation where each item's weight is written to its left. This is exemplified in Figure 1, which shows an example rule for CFG parsing with items of the form $[A, i, j]$, representing a subtree rooted by nonterminal A and spanning input tokens $i + 1$ through j.

One item in the weighted deduction system is designated as the goal item, and the fundamental problem is to calculate the total weight of all derivations of this item, where the total weight is calculated using a generalized sum operation, written $\bigoplus$. An extension of this is to calculate the total weight of all derivations of the goal item G that also contain item X, written $\gamma(X)$ (Goodman, 1999):

$$\gamma(X) = \bigoplus_{D:X,G \in D} \text{weight}(D)$$

where $X, G \in D$ means that item X and goal item G are each the consequent of some rule in D (for G, this is specifically the root rule). These γ values are a core component of the inside-outside Expectation Maximization (EM) algorithm for unsupervised probabilistic context-free grammar (PCFG) induction (Baker, 1979), where $\gamma(X)$ is calculated by combining a corresponding inside value (total weight of subtrees rooted at X) and outside value (cost of completing a derivation containing X). For the purposes of the EM algorithm, the $\oplus$ operation is standard addition, and F_R computes the product of its arguments. If we define the $\oplus$ operation to be $\max$, $\gamma(X)$ corresponds to the value of the best parse tree subject to the constraint that a particular constituent X be included. This value can be found by combining an inside and outside value, using the same procedure as is used for EM, but substituting $\max$ for addition.

Gildea (2020) discussed classes of weighted deduction system where computation of outside values (and by extension, γ values) can be done efficiently. Formally, they were interested in systems where $\gamma(X)$ can (or cannot) be calculated for every item X in time $O(|E|\gamma)$, where $|E|$ is the number of rules in the system, and $\gamma = \max_X |\gamma(X)|$ is the largest number of bits required to represent the

total weight of an item. They termed this "efficient outside computation."

One important class of weighted deduction system is the minimum of superior functions (Knuth, 1977). In this framework, each rule weight function F_R is a superior function, meaning that it is monotonically increasing in each argument and the result is always greater than or equal to each of its arguments. The generalized sum $\bigoplus$ in this framework used for calculating total weight is the minimum operation:

$$\gamma(X) = \bigoplus_{D:X,G \in D} \text{weight}(D)$$
$$= \min_{D:X,G \in D} \text{weight}(D)$$

Best-first search is possible in this framework using a generalization of Dijkstra's algorithm (Nederhof, 2003). It is interesting to ask whether efficient outside computation is always possible within this framework, and even more generally, whether the conditions necessary for best-first search are sufficient for efficient outside computation. The A* parsing system of Klein and Manning (2003) is an instance of the minimum of superior functions framework[1] that uses best-first search. Outside values are of particular interest for A* parsing because they can be used as admissible search heuristics (Pauls and Klein, 2009a), and to efficiently find the k best parses (Pauls and Klein, 2009b). When the function F_R simply takes a product of its arguments, as in Pauls and Klein (2009b), efficient outside computation is possible. In this paper, we address the question of whether this is guaranteed by the minimum of superior functions framework or merely an artifact of this particular system.

Gildea (2020) pointed out that there is no known efficient algorithm for outside computation in the minimum of superior functions framework. However, they did not present a formal hardness result. In this work, we prove that general efficient outside computation in this framework would yield a subquadratic time algorithm for the Orthogonal Vectors Problem, violating the Orthogonal Vectors Conjecture (Williams, 2005; Vassilevska Williams, 2015), which states that no such algorithms exist because their existence would violate the Strong Exponential Time Hypothesis (SETH) (Impagliazzo and Paturi, 1999) and yield a subexponential

---

[1]To see this, simply negate the log probabilities and replace max with min in their system. The superior function F_R is addition for all R.
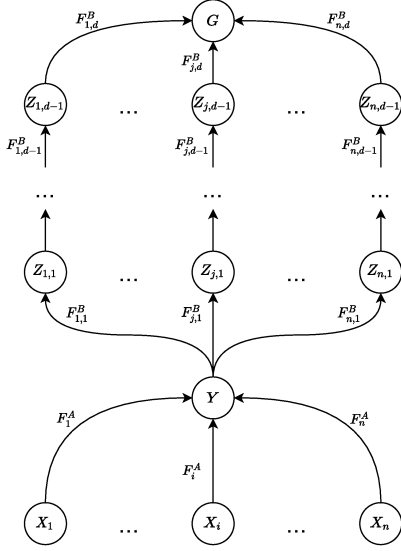
Figure 2: Graphical representation of the different paths through the weighted deduction system.

time algorithm for SAT. The Strong Exponential Time Hypothesis, a somewhat stronger assumption than P≠NP, is widely conjectured to be true, and has been used as an assumption in a number of recent hardness results, including the result that string edit distance cannot be computed in strongly subquadratic time, unless SETH is false (Backurs and Indyk, 2015).

3 Reduction

We begin with the Orthogonal Vectors Problem: given sets $A, B \subseteq \{0,1\}^d$ where $|A| = |B| = n$, determine whether there exist vectors $a \in A$ and $b \in B$ such that their dot product $a \cdot b = \sum_{k=1}^{d} a_k b_k$ is 0. We now reduce this problem to a weighted deduction system in the minimum of superior functions framework.

First, define n axiom items X_i, $i \in [1, n]$, and construct n corresponding rules R_i^A leading from X_i to item Y:

$$\frac{w: \quad X_i}{F_i^A(w): \quad Y}$$

where $F_i^A(w) = w + (d+1)i$. The weight for each axiom item X_i is defined to be 0. The intuition here is that the index i refers to a specific vector $A_i \in A$, and the resulting weight will allow later rule weight functions to identify the starting point for the derivation and thus which vector in A to compare to a vector in B. This is possible because all derivations from Y to the goal item will add no more than d to $F_i^A(\text{weight}(X_i)) = (d + 1)i$, making the value of i uniquely recoverable.

Next, we construct n rules $R_{j,1}^B, j \in [1, n]$ of the form:

$$\frac{w: \quad Y}{F_{j,1}^B(w): \quad Z_{j,1}}$$

where each $F_{j,1}^B$ is the rule weight function corresponding to the first dimension of vector $B_j \in B$. We define the rule weight functions used here and those in the upcoming rules in the following way:

$$F_{j,k}^B(w) = w + A_{\text{index}(w),k} B_{j,k}$$

where $\text{index}(w) = \lfloor \frac{w}{d+1} \rfloor$. Intuitively, these functions "look up" the choice of which vector A_i was used to begin the computation using $\text{index}(w)$, and multiply the k-th dimension of that vector with the k-th dimension of B_j.

Note that while item Y could be removed by defining a rule deriving each $Z_{j,1}$ from each X_i directly with an appropriately-defined rule weight function, this would require n^2 rules, whereas introducing the intermediate item Y provides the same connectivity with only $2n$ rules while using the weight to keep track of which X_i was chosen. This is important because our proof requires that the deduction system used for the reduction be constructed in subquadratic time.

Now we construct $n(d-2)$ rules $R_{j,k}^B, j \in [1, n]$, $k \in [2, d-1]$ of the form:

$$\frac{w: \quad Z_{j,k-1}}{F_{j,k}^B(w): \quad Z_{j,k}}$$

where $F_{j,k}^B$ was defined above. The intuition is that each family of $Z_{j,k}$ items for a particular j forms a chain that eventually covers all d dimensions of B_j.

So far we have not covered the final dimension of the B vectors, so we do so now by constructing n rules $R_{j,d}^B$ of the form:

$$\frac{w: \quad Z_{j,d-1}}{F_{j,d}^B(w): \quad G}$$

where G is the goal item of the weighted deduction system.

We now discuss properties of the resulting system, a graphical representation of which is presented in Figure 2.

Every computation begins at one of the axiom items X_i corresponding to A_i and always passes through Y. The computation then proceeds down one of n chains, each corresponding to a vector in B. Because the rule weight function $F_{j,k}^B(w)$ applied at each edge adds at most 1 to w and each

chain from Y to G consists of exactly d edges, the weight of any item in the chosen chain will never be more than d greater than the weight of the edge from the chosen X_i to Y. Because each of those edges' weights is a distinct multiple of $d+1$, the choice of the starting point (and corresponding vector in A) can be recovered by each $F_{j,k}^B$ in the chain using the $\mathrm{index}(w)$ function. This allows each chain to effectively calculate the dot product between its respective vector in B and the chosen vector in A.

In the superior function framework of Knuth (1977), the total weight of an item C (referred to as $\gamma(C)$) is the minimum weight over complete derivations D containing C and the goal item G:

$$\gamma(C) = \min_{D:C,G \in D} \mathrm{weight}(D)$$

where $\mathrm{weight}(D)$ is the result of the rule weight function for the (unique) rule producing G in derivation D.

For the purposes of the reduction, we are interested in the n total weights $\gamma(X_i)$. Note that every derivation containing X_i defines a path from X_i to G, and there are exactly n such paths for a given X_i: one for each chain from Y to G, each corresponding to a vector B_j. Recalling that $\mathrm{weight}(X_i) = 0$, we can rewrite $\gamma(X_i)$ as follows:

$$\gamma(X_i) = \min_j \left[\left(\bigcirc_{k=0}^{d-1} F_{j,d-k}^B \right) \left(F_i^A(\mathrm{weight}(X_i)) \right) \right]$$

$$= \min_j \left[\left(\bigcirc_{k=0}^{d-1} F_{j,d-k}^B \right) \left(F_i^A(0) \right) \right]$$

$$= \min_j \left[\left(\bigcirc_{k=0}^{d-1} F_{j,d-k}^B \right) ((d+1)i) \right]$$

$$= \min_j \left[\left(\sum_{k=0}^{d-1} A_{i,d-k} B_{j,d-k} \right) + (d+1)i \right]$$

$$= \min_j \left[(A_i \cdot B_j) + (d+1)i \right]$$

where $\bigcirc$ represents repeated function composition.

We can use the values of $\gamma(X_i)$ to solve the Orthogonal Vectors Problem. Because $A_i \cdot B_j$ is at most d, $\gamma(X_i)$ is evenly divisible by $(d+1)$ if and only if there is a vector in B that is orthogonal to A_i:

$$\gamma(X_i) \equiv 0 \pmod{d+1} \iff \exists j \, (A_i \cdot B_j = 0)$$

The complete algorithm for solving the problem using this technique is as follows:

1. Construct the deduction system as described above. ($O(nd)$ time)

2. Calculate $\gamma(X_i)$ for all $i \in [1, n]$.

3. Check whether $\gamma(X_i) \equiv 0 \pmod{d+1}$ for any i. If yes, then there exist vectors $A_i \in A$ and $B_j \in B$ such that $A_i \cdot B_j = 0$. Otherwise, no such vectors exist. ($O(n)$ time)

If all values $\gamma(X_i)$ could be calculated in linear time with respect to the number of edges $|E| \in O(nd)$, then the Orthogonal Vectors Problem could be solved in time $O(nd)$, violating the Orthogonal Vectors Conjecture which states that there is no strongly subquadratic time algorithm for this problem, and by extension violating the Strong Exponential Time Hypothesis (SETH) (Impagliazzo and Paturi, 1999). Because the proposed deduction system is an instance of the superior functions framework of Knuth (1977), we conclude that efficient outside computation is not possible in general under that framework unless SETH is false.

4 Conclusion

This work provides a formal proof that efficient outside computation is not possible in general for the minimum of superior functions framework (Knuth, 1977) (unless the Strong Exponential Time Hypothesis is false). This indicates that the conditions necessary for best-first search are not sufficient for efficient outside computation. It remains an open problem to characterize the class of functions for which best-first search and efficient outside computation are both always possible.

References

Arturs Backurs and Piotr Indyk. 2015. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). In *Proceedings of the forty-seventh annual ACM Symposium on Theory of Computing (STOC)*, pages 51–58.

J. K. Baker. 1979. Trainable grammars for speech recognition. In *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America*, pages 547–550.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the *EM* algorithm. *Journal of the Royal Statistical Society*, 39(1):1–21.

Jason Eisner and Giorgio Satta. 1999. Efficient parsing for bilexical context-free grammars and head automaton grammars. In *Proceedings of the 37th Annual Conference of the Association for Computational Linguistics (ACL-99)*, pages 457–464, College Park, Maryland.

Daniel Gildea. 2020. Efficient outside computation. *Computational Linguistics*, 46(4):745–762.

Joshua Goodman. 1999. Semiring parsing. *Computational Linguistics*, 25(4):573–605.

R. Impagliazzo and R. Paturi. 1999. Complexity of k-SAT. In *Proceedings of Fourteenth Annual IEEE Conference on Computational Complexity*, pages 237–240.

Dan Klein and Christopher D. Manning. 2003. A* parsing: Fast exact Viterbi parse selection. In *Proceedings of the 2003 Meeting of the North American chapter of the Association for Computational Linguistics (NAACL-03)*, pages 119–126.

Donald E. Knuth. 1977. A generalization of Dijkstra's algorithm. *Info. Proc. Letters*, 6(1):1–5.

Adam Lopez. 2009. Translation as weighted deduction. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 532–540, Athens, Greece.

I. Dan Melamed, Giorgio Satta, and Ben Wellington. 2004. Generalized multitext grammars. In *Proceedings of the 42nd Annual Conference of the Association for Computational Linguistics (ACL-04)*, pages 661–668, Barcelona, Spain.

M.-J. Nederhof. 2003. Weighted deductive parsing and Knuth's algorithm. *Computational Linguistics*, 29(1):135–144.

A. Pauls and D. Klein. 2009a. Hierarchical Search for Parsing. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 557–565.

A. Pauls and D. Klein. 2009b. K-Best A* Parsing. In *Proceedings of the 2009 Annual Conference of the Association for Computational Linguistics*, pages 958–966.

Stuart M. Shieber, Yves Schabes, and Fernando C. N. Pereira. 1995. Principles and implementation of deductive parsing. *The Journal of Logic Programming*, 24(1-2):3–36.

Klaas Sikkel. 1997. *Parsing Schemata*. Springer Verlag, Berlin.

Virginia Vassilevska Williams. 2015. Hardness of Easy Problems: Basing Hardness on Popular Conjectures such as the Strong Exponential Time Hypothesis (Invited Talk). In *10th International Symposium on Parameterized and Exact Computation (IPEC 2015)*, volume 43 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17–29, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Ryan Williams. 2005. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theor. Comput. Sci.*, 348(2):357–365.