Operations Research Letters 42 (2014) 197–202

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

# A fully polynomial-time approximation scheme for approximating a sum of random variables

Jian Li *, Tianlin Shi

*Institute for Interdisciplinary Information Sciences, Tsinghua University, China*

ARTICLE INFO

*Article history:*
Received 29 October 2013
Received in revised form
11 February 2014
Accepted 11 February 2014
Available online 18 February 2014

*Keywords:*
Threshold probability
Tail probability
Approximate counting
Counting knapsack
FPTAS

ABSTRACT

Given $n$ independent integer-valued random variables $X_1, X_2, \ldots, X_n$ and an integer $C$, we study the fundamental problem of computing the probability that the sum $\boldsymbol{X} = X_1 + X_2 + \cdots + X_n$ is at most $C$. We assume that each random variable $X_i$ is implicitly given by an oracle $\mathcal{O}_i$, which given two input integers $n_1, n_2$ returns the probability of $n_1 \le X_i \le n_2$. We give the first deterministic fully polynomial-time approximation scheme (FPTAS) to estimate the probability up to a relative error of $1 \pm \epsilon$. Our algorithm is based on the technique for approximately counting knapsack solutions, developed in Gopalan et al. (2011).



## 1. Introduction

We study the following fundamental problem. The input consists of $n$ independent (not necessarily identically distributed) random integral variables $X_1, \ldots, X_n$ and an integer $C$. Our task is to compute the following probability value

$$F(C) = \Pr\left[\sum_{i=1}^{n} X_i \le C\right]. \tag{1}$$

It is well known that computing $F(C)$ is #P-hard (see e.g., [9]). The hardness of computing $F(C)$ has an essential impact in the area of stochastic optimization as many problems generalize and/or utilize this basic problem in one way or another, thus inheriting the #P-hardness. Although we can sometimes use for example the linearity of expectation to bypass the difficulty of computing $F(C)$, more than often no such simple trick is applicable, especially in the context of risk-aware stochastic optimization where people usually pay more attention to the tail probability than the expectation.

Despite the importance of the problem, surprisingly, no approximation algorithm with provable multiplicative factor is known. We note that we can easily obtain an additive PRAS (polynomial-time randomized approximation scheme) for this problem via the Monte-Carlo method: for each $i \in \{1, 2, \ldots, n\}$, generate $K$ independent samples $X_i^{(k)}$, $k = 1, 2, \ldots, K$, according to the distribution of $X_i$, and then use the empirical average

$$\widetilde{F}(C) = \frac{1}{K}\sum_{k=1}^{K} \mathbb{I}\left(\sum_{i=1}^{n} X_i^{(k)} \le C\right)$$

as the estimation of $F(C)$, where $\mathbb{I}(\cdot)$ is the indicator function. It is easy to see that $\widetilde{F}(C)$ is an unbiased estimator of $F(C)$. By standard Chernoff bound, one can see that with $K = \mathrm{poly}(1/\epsilon)$ samples, the estimate is within an additive error $\epsilon$ from the true value with constant probability (see e.g., [14]). To get a reasonable multiplicative approximation factor (say a constant close to 1), we need to set the additive error at the order of $F(C)$. So the number of samples needs to be $\mathrm{poly}(1/F(C))$, which can be exponentially large, when $F(C)$ is exponentially small. In certain application domains such as risk analysis, small probabilities (often associated with catastrophic losses) can be significant, thus demanding accurate estimations. Therefore, it is important to develop polynomial time approximation schemes for estimating such small probabilities.

**Assumptions.** Before presenting our main result, we need some notations and assumptions of the computation model. We assume that all random variables are discrete and the support of $X_i$, denoted as $\mathrm{supp}_i$, is finite and consists of only integers. Without loss of generality, we can assume all $X_i$s are nonnegative (i.e., $\mathrm{supp}_i \subseteq \mathbb{N}$) and $0 \in \mathrm{supp}_i$ for all $i$. To see why this is without loss of generality, simply consider the equivalent problem of computing $\Pr[\sum_{i=1}^{n}(X_i - \min X_i) \le C - \sum_{i=1}^{n} \min X_i]$, where $\min X_i$ is the

* Corresponding author.
*E-mail addresses:* lapordge@gmail.com, lijian83@mail.tsinghua.edu.cn (J. Li), stl501@gmail.com (T. Shi).

http://dx.doi.org/10.1016/j.orl.2014.02.004
0167-6377/

minimum value in $\text{supp}_i$. Under such an assumption, the problem is non-trivial only for $C > 0$. Moreover, we can assume that $\text{supp}_i \in [0, C+1]$ for all $i$ since we can place all mass in $[C+1, \infty)$ at the single point $C+1$, which does not affect the answer. The distribution of each random variable $X_i$ is implicitly given by an oracle $\mathcal{O}_i$, which given two input value $(n_1, n_2)$ returns the value $\Pr[n_1 \le X_i \le n_2]$ in constant time.

Our main result is a *fully polynomial-time approximation scheme* (FPTAS) for computing $F(C)$. For ease of notation, we use $(1 \pm \epsilon)F(C)$ to denote the interval $[(1-\epsilon)F(C), (1+\epsilon)F(C)]$. Let $\Delta = \prod_i \Pr[X_i = 0]$. Clearly $\Delta$ is a lower bound on the solution. Recall that we say there is an FPTAS for the problem, if for any positive constant $\epsilon > 0$, there is an algorithm which can produce an estimate $\widetilde{F}$ with $\widetilde{F} \in (1 \pm \epsilon)F(C)$ in $\text{poly}(n, \epsilon^{-1}, \log C, \log \frac{1}{\Delta})$ time (see e.g., [14]).

**Theorem 1.1.** *We are given $n$ independent nonnegative integer-valued random variables $X_1, \ldots, X_n$, a positive integer $C$, and a constant $\epsilon > 0$. Suppose that for all $i \in \{1, 2, \ldots, n\}$, $\text{supp}_i \subseteq [0, C+1]$, $0 \in \text{supp}_i$ and there is an oracle $\mathcal{O}_i$, which, upon two input integers $(n_1, n_2)$, returns the value $\Pr[n_1 \le X_i \le n_2]$ in constant time. There is an FPTAS for estimating $\Pr[\sum_{i=1}^n X_i \le C]$ and the running time is $O\big(\frac{n^3}{\epsilon^2} \log(\frac{1}{\Delta})^2 \log C\big)$.*

**Remark 1.** For simplicity of presentation, we assume in the above theorem a computation model in which any real arithmetic can be performed with perfect accuracy in constant time and the probability values returned by the oracle are reals, also with perfect accuracy. In Section 2.3, we show how to implement our algorithm in a computation model where only bit operations are allowed and the oracles also return numerical values with finite precision. We show that the bit complexity of the algorithm is still a (somewhat larger) polynomial.

**Remark 2.** Note that, the oracle assumption is weaker than assuming the explicit representations of the distributions (i.e., listing the probability mass at every point). In fact, if the input is the explicit representations of the distributions, we can preprocess the input in linear time so that each oracle call to $\mathcal{O}_i$ can be simulated in $O(\log |\text{supp}_i|)$ time. This can be done by computing the prefix sums $\Pr[X_i \le x]$ for all $x \in \text{supp}_i$ in $O(|\text{supp}_i|)$ time. Then for each oracle call $(n_1, n_2)$, we use binary search to find out the smallest value $x_1 \in \text{supp}_i$ that is no smaller than $n_1$ and the largest value $x_2 \in \text{supp}_i$ that is no larger than $n_2$ in $O(\log |\text{supp}_i|)$ time. Therefore, $\Pr[n_1 \le X_i \le n_2]$ is the same as $\Pr[x_1 \le X_i \le x_2]$, which can be computed from the prefix sums in constant time.

### 1.1. Related work

There is a large body of work on estimating or upper/lower-bounding the distribution of the sum of independent random variables. See e.g., [1,18,12,3,13]. Those works are based on analytic numerical methods (e.g., Edgeworth expansion, saddle point method) which either require specific families of distributions and/or do not provide any provable multiplicative approximation guarantees.

Our problem is a generalization of the counting knapsack problem. For the counting knapsack problem, Morris and Sinclair [15] obtained the first FPRAS (fully polynomial-time randomized approximation scheme) based on the Markov Chain Monte-Carlo (MCMC) method. Dyer [4] provided a completely different FPRAS based on dynamic programming. The first deterministic FPTAS is obtained by Gopalan et al. [7] (see also the journal version [19]).

Our problem is also closely related to the threshold probability maximization problem (see a general formulation in [11]). In this problem, we are given a ground set of items. Each feasible solution to the problem is a subset of the elements satisfying some property (this includes problems such as shortest path, minimum spanning tree, and minimum weight matching). Each element $b$ is associated with a random weight $X_b$. Our goal is to find a feasible set $S$ such that $\Pr[\sum_{b \in S} X_b \le C]$ is maximized, for a given threshold $C$. There is a large body of literature on the threshold probability maximization problem, especially for specific combinatorial problems and/or special distributions. For example, Nikolova, Kelner, Brand and Mitzenmacher [17] studied the corresponding shortest path version for Gaussian, Poisson and exponential distributions. Nikolova [16] extended this result to an FPTAS for any problem with Gaussian distributions, if the deterministic version of the problem has a polynomial-time algorithm. The minimum spanning tree version with Gaussian distributed edges has also been studied in [5]. For general discrete distributions, Li and Deshpande [10] obtained an additive PTAS if the deterministic version of the problem can be solved exactly in pseudo-polynomial time. Very recently, Li and Yuan [11] further generalized this result to the class of problems for which the multi-objective deterministic version admits a PTAS.

Our problem is also closely related to the fixed set version of the stochastic knapsack problem. In this problem, we are given a knapsack of capacity $C$ and a set of items with random sizes and profits. Their goal is to find a set of items with maximum total profit subject to the constraint that the overflow probability is at most a given parameter $\gamma$. Kleinberg, Rabani and Tardos [9] first considered the problem with Bernoulli-type distributions and provided a polynomial-time $O(\log 1/\gamma)$-approximation. Better results are known for specific distributions, such as exponentially distributions [6], Gaussian distributions [8,16]. For general discrete distributions, bi-criteria additive PTASes (i.e., the overflow probability constraint may be violated by an additive factor $\epsilon$ for any constant $\epsilon > 0$) are known via different techniques [2,10,11].

## 2. Algorithm

Our algorithm is based on dynamic programming. In Section 2.1, we provide the recursion of the dynamic program, which is largely based on the idea developed in [7,19], with some necessary adaptations. However, since the support of each random variable can be exponentially large, it is not immediately clear how the recursion can be implemented efficiently given the oracles. We address this issue in Section 2.2. In Section 2.3, we analyze the bit complexity of our algorithm.

### 2.1. The dynamic program

We first notice that $\Pr[\sum_{j=1}^i X_j \le C]$, for any $i \in \{1, 2, \ldots, n\}$, is a nondecreasing function of $C$. We consider its inverse function $\tau(i, a) : \{1, 2, \ldots, n\} \times \mathbb{R}_{\ge 0} \to \mathbb{N} \cup \{\pm\infty\}$, which is defined to

$$\tau(i,a) = \begin{cases} \min\left\{C \mid C \ge 0 \text{ and } \Pr\left[\sum_{j=1}^i X_j \le C\right] \ge a\right\}, & 0 < a \le 1; \\ +\infty, & a > 1; \\ -\infty, & a = 0. \end{cases}$$

It is easy to see that $\tau(i, a)$ is nondecreasing in $a$. The following simple lemma is needed. We omit the proof, which is straightforward.

**Lemma 2.1.** *Both of the following statements hold true.*

1. $\Pr\left[\sum_{i=1}^n X_i \le C\right] = \max\{a : \tau(n, a) \le C\}$.
2. $\tau(i, a) = 0$ *for any* $i \in \{1, 2, \ldots, n\}$ *and* $a \le \Delta$, *where* $\Delta = \prod_i \Pr[X_i = 0]$.