Randomized Algorithms

Lecture 10: Updated February 9, 2006

Lecturer: Eric Vigoda *Original Scribe: Ivona Bezáková*

Approximately Counting KNAPSACK Solutions

In this lecture, we study the counting and sampling versions of the knapsack problem. This is a further example (as with Eucledian TSP in the last lecture) of the power of simple dynamic programming based approaches. It also highlights the intimate connection between approximate counting and random sampling.

Here is the problem statement.

#KNAPSACK: We are given n objects, together with each object $i \in [n]$ (recall $[n] = \{1, 2, \ldots, n\}$) we have its *integer* weight w_i, and the total weight W our knapsack can hold. Our objective is to find the number of subsets $S \subseteq [n]$ such that $\sum_{i \in S} w_i \leq W$. We call the sets satisfying this weight constraint *feasible*, and denote them as $\mathcal{S}$. Thus, we are computing $|\mathcal{S}|$. A geometric interpretation of the problem is the following. Recall, the n-dimensional hypercube has vertex set $\{0,1\}^n$ and edges connecting vertices which differ in exactly one coordinate. The constraint $\sum_i w_i \leq W$ defines a hyperplane. Thus, we are counting the number of points of the hypercube that intersect the hyperplane.

This problem is sometimes referred to as the zero-one #KNAPSACK. #KNAPSACK is known to be $\#P$-complete, therefore we focus on approximation schemes.

We present an fpras for #KNAPSACK due to Dyer [1]. An *FPRAS* (fully polynomial randomized approximation scheme), is a randomized algorithm, which for all $0 < \epsilon, \delta < 1$, computes OUT satisfying:

$$OUT(1-\epsilon) \leq |\mathcal{S}| \leq OUT(1+\epsilon)$$

with probability $\geq 1 - \delta$, and running in time polynomial in n, $1/\epsilon$, and $\log(1/\delta)$.

The first fpras was due to Morris and Sinclair [2]. They used a more standard Markov chain Monte Carlo approach, but the analysis of their algorithm was considerably more complicated than Dyer's approach.

Dyer first presents an easy algorithm to approximate $|\mathcal{S}|$ within a polynomial in n factor. Using this sampler, it is then straightforward to estimate $|\mathcal{S}|$ arbitrarily close.

First we present a pseudo-polynomial solution for #KNAPSACK, i.e., the running time depends on W. Let $F(j,k)$ be the number of sets $T \subseteq [j]$ such that $\sum_{i \in T} w_i \leq k$. Thus, $|\mathcal{S}| = F(n, W)$.

Now we have an easy recurrence for $F(j,k)$. Either: we include the j-th item in the knapsack and our available weight goes from k to $k-w_j$; or we don't include the j-th item and the available weight stays the same. Hence, we have:

$$\begin{aligned} F(j,k) &= F(j-1,k) + F(j-1,k-w_j) \qquad \text{if } j \geq 1 \\ F(0,k) &= \begin{cases} 1 & \text{if } k \geq 0 \\ 0 & \text{if } k < 0 \end{cases} \end{aligned}$$

Using dynamic programming we find $F(n,W)$ in time $O(nW)$.

We don't want the running time to depend on W. Hence, we'll rescale the weights so that W scales down to something which is polynomial in n. The scaling will round the weights w_i down to integers, thereby altering the set of feasible solutions. In fact, the rounding will increase the number of solutions, but by at most a factor of $(n+1)$.

Scaling. Without loss of generality, assume the weights are sorted: $0 \leq w_1 \leq w_2 \leq \cdots \leq w_n \leq W$. Let the scaled-down weights be $\omega_i := \lfloor \frac{n^2}{W} w_i \rfloor$, and the new total weight $\Omega := n^2$. Let Φ be the solution set of this new instance of #KNAPSACK. Using the above dynamic programming we can compute $|\Phi|$ in time $O(n^3)$.

Note, if $S \in \mathcal{S}$, i.e., S is a feasible set of the original problem, then

$$\sum_{i \in S} \omega_i \leq \frac{n^2}{W} \sum_{i \in S} w_i \leq \frac{n^2}{W} W \leq n^2,$$

and then S is also solution to the new instance. Therefore, $\mathcal{S} \subseteq \Phi$.

Comparing $|\Phi|$ with $|\mathcal{S}|$. We have just shown $|\mathcal{S}| \leq |\Phi|$. We will now define a map $f : \Phi \to \mathcal{S}$ such that for every $S \in \mathcal{S}$ there will be at most $(n+1)$ sets in Φ mapping to it. This implies $|\Phi| \leq (n+1)|\mathcal{S}|$. Therefore in $O(n^3)$ time we can compute $|\Phi|$ which satisfies:

$$\frac{|\Phi|}{n+1} \leq |\mathcal{S}| \leq |\Phi| \tag{10.1}$$

It remains to define f. For $S \in \mathcal{S}$, let $f(S) := S$.

Consider $S \in \Phi \setminus \mathcal{S}$. Let K be the largest integer k such that $w_k \leq W/n$. Note, any $S' \subseteq [K]$ is a solution to the original problem (i.e. $S' \in \mathcal{S}$). In particular, for such a $S' \subseteq [K]$ we have

$$\sum_{i \in S'} w_i \leq \sum_i \frac{W}{n} \leq W.$$

Hence, $S' \in \mathcal{S}$. Therefore, for $S \in \Phi \setminus \mathcal{S}$, we know $S \not\subseteq [K]$. Let ℓ be the index of the heaviest element in S. Note, $\ell > K$. Then we define $f(S) := S \setminus \{\ell\}$. We need to show that $f(S) \in \mathcal{S}$. To simplify notation, let

$$\delta_i := w_i \frac{n^2}{W} - \omega_i,$$

denote the rounding error. Thus,

$$w_i = \frac{W}{n^2}\left(\omega_i + \delta_i\right),$$

and $0 \leq \delta_i \leq 1$. Finally, we need to show that $f(S) \in (S)$. We have:

$$\begin{aligned}
\sum_{i \in f(S)} w_i &= \frac{W}{n^2} \sum_{i \in f(S)} (\omega_i + \delta_i) \\
&= \frac{W}{n^2}\left(\sum_{i \in S} \omega_i - \omega_\ell + \sum_{i \in f(S)} \delta_i\right) \\
&\leq \frac{W}{n^2}\left(\sum_{i \in S} \omega_i - \omega_\ell + n\right) \\
&\leq \frac{W}{n^2} \sum_{i \in S} \omega_i \quad \text{since } \omega_\ell > W/n \\
&\leq \frac{W}{n^2}\left(n^2\right) \quad \text{since } S \in \Phi \\
&= W,
\end{aligned}$$

Therefore, $f(S) \in \mathcal{S}$ and f is properly defined from Φ to $\mathcal{S}$.

How many sets map to a given set $S \in \mathcal{S}$? Let ℓ be the index of the heaviest element in S. Then, by the definition of f, the set S maps to itself via the first rule defining f, and at most $n - \ell$ sets map to S via the second rule. This sums to at most n sets mapping to any $S \in \mathcal{S}$. This proves (10.1).

Sampling. We now use the above dynamic programming approach to sample uniformly at random from Φ. We will trace back a solution using the table F. We start with the empty set and we will add elements according to the probability distribution defined by F.

1. Let $S := \emptyset$, $j := n$, and $k := n^2$.
2. While $j > 0$ do:
   (a) With probability $F(j-1, k-\omega_j)/F(j,k)$:

   $$\text{Set } S = S \cup \{j\} \text{ and set } k = k - \omega_j.$$

   (b) Decrease j by one.

It is easy to see that this algorithm samples sets uniformly from Φ. Given access to F, the above algorithm takes $O(n)$ time to generate a sample. Thus, for t samples it takes $O(n^3)$

time to compute F and then $O(tn)$ time to generate t random samples from Φ, therefore it takes $O(n^3 + tn)$ time in total.

Counting. Let $p := |\mathcal{S}|/|\Phi|$. Note, we know $p \geq 1/n$. We will use sampling to estimate p. Generating $\frac{n}{\epsilon^2}\log(1/\delta)$ sets from Φ and counting the proportion that fall in $\mathcal{S}$, by Chernoff bounds, we then have an estimate of p that is within a multiplicative factor $(1 \pm \varepsilon)$ with probability $\geq 1 - \delta$. Note, the total running time is $O(n^3 + \varepsilon^{-2}n^2\log(1/\delta))$.

Finally, recall

$$|\mathcal{S}| = p|\Phi|$$

We know $|\Phi|$ and we have an estimate of p which is within a factor $(1 \pm \epsilon)$ with probability $\geq 1 - \delta$. Hence, we have an estimate of $|\mathcal{S}|$ with the same guarantees. This completes the description of the fpras.

Improvements and generalizations. Using randomized rounding, Dyer improved the running time to $O\left(\left(n^{5/2}\sqrt{\log(1/\varepsilon)} + n^2\varepsilon^{-2}\right)\log(1/\delta)\right)$. The m-constraint #KNAPSACK has m weights for each object and m total weights. The objective is to find the number of sets satisfying all m weight constraints. Dyer's approach carries through, resulting in $O(n^{2m+1} + \varepsilon^{-2}n^{m+1})$ running time. In the general (non zero-one) #KNAPSACK problem one wants to find the number of all feasible *multi*sets. Again, Dyer's approach gives $O(n^{2m+3} + \varepsilon^{-2}n^{m+3})$ time bound.

Open Problems: Is there a deterministic algorithm to estimate $|\mathcal{S}|$ arbitrarily close? The above approach computes $|\Phi|$ deterministically, and this yields a $\sqrt{n}$ approximation to $|\mathcal{S}|$. We only use randomness to boost the approximation factor.

References

[1] M. Dyer. *Approximate Counting by Dynamic Programming*, Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC), 693-699, 2003.

[2] B. Morris and A.J. Sinclair. Random walks on truncated cubes and sampling 0-1 knapsack solutions. *SIAM J. Comput.*, 34(1):195-226, 2004.