Approximate Counting via Stratified Sampling: the 3d hard-sphere entropy constant

Isabel Beichl [*]   Francis Sullivan [†]

October 10, 2003

We have used a stratified sampling version of Knuth's algorithm [1] for estimating size of backtrack trees to attack several difficult approximate counting problems, including the problem of estimating the hard-sphere entropy for a three-dimensional grid lattice.

A number of years ago Knuth developed an approximate counting method in order to estimate the running time of a back-track program without actually performing the entire backtrack. The underlying idea is simple. Recall that any backtrack can be thought of as a search of a tree which backs up to the first ancestor node having an available choice whenever it is blocked, and continues doing this until an "answer" is found or all nodes have been examined. If we imagine that the backtrack is a tree (not necessarily balanced) then this amounts to a depth first traversal of the tree that stops at a node satisfying some pre-specified condition. In many situations, it is useful to be able to estimate how much work will be done before the answer is found, i.e. to estimate how large the tree is without actually traversing it.

If the backtrack search generated a perfect binary tree, the size of the search is easy to estimate. Just determine d, the depth of the tree and assume that the whole tree must be traversed before finding the answer. The amount of

[*] NIST, 100 Bureau Dr., Gaithersburg, MD 20899 isabel.beichl@nist.gov

[†] IDA/CCS, 17100 Science Dr., Bowie, MD 20715 fran@super.org

work is then 2^{d+1}, the number of nodes in the tree. To determine d, just walk down one branch of the tree and count the number of steps to reach the leaf.

Amazingly, a simple and obvious-seeming generalization of this idea works in much more general situations where the tree is not binary or even fixed degree and the depth is not uniform. A "sample" is a traversal of any path of the tree, stopping when a leaf is reached. At each step k choose at random among the n_k children of the current node and record n_k. After traversing a path, the estimate obtained for the number of nodes is given by:

$$c_{tot} = n_0(1 + n_1(1 + n_2(1 + \ldots)))$$

Averaging values of c_{tot} over sufficiently many samples gives the estimate.

Knuth's method can be thought of as an application of importance sampling. For one sample, the number of possible choices for the first k levels of the tree is $c_{k-1} = n_0 n_1 \ldots n_{k-1}$ and, because of the uniform choice at each level, the probability of making that particular sequence of choices is $1/c_n = 1/(n_0 n_1 \ldots n_{k-1})$. If the choices were made using some other, non-uniform probabilities p_i, then the estimate for number of leaves at the k^{th} step would be $c_{k-1} = 1/(p_0 p_1 \ldots p_{k-1})$

Non-uniform probabilities have been use with great success in several problems, including approximating the permanent of a zero-one matrix and counting the number of monomer-dimer covers of a three-dimensional cubic lattice. In these cases, the backtrack that is sampled is an algorithm for finding a perfect matchings. The probabilities used were based on Sinkhorn balancing which, in this application, made use of the fact that the graph was bi-partite. In more general situations, however, it is not clear how to generate useful non-uniform probabilities. For example, it is possible to use Knuth's basic method to estimate the number of independent node sets in an undirected graph, but there is no obvious probability to use except uniform weights.

We have used a more general development of Knuth's fundamental idea that was introduced by [2]. Instead of sampling by following one path to a leaf, each sample follows several intersecting paths, chosen according to a classification of nodes into several strata. For example, if the backtrack comes from some type of search of a graph, then nodes can be grouped in strata based on number of neighbors. At each step, one sample node is taken from each stratum and weights are accumulated based on the history of the node. Actually implementing this method presented interesting challenges in design of data structures and averaging techniques for avoiding the use of extended precision.

References

[1] Chen, Pang-Chieh “Heuristic sampling: A method for prediciting the performance of tree searching programs”, *SIAM Journal on Computing* **21** (1992), 295-315.

[2] Knuth, Donald E. “Estimating the Efficiency of Backtrack Programs”, *Selected Papers on Analysis of Algorithms*, CSLI Publications, Stanford, California, (2000).