Bound on the runtime of the even/odd contingency table counter

John Mount*

August 31, 2014

Abstract

We derive a bound on the run time and space utilization of the even/odd integer counting method described in [Mount, 2000].

1 Introduction

A section in Fast Unimodular counting suggests an even/odd decomposition for exactly counting the number non-negative integer solutions x to the linear system $Ax = b$. We derived and correct the claimed time and space complexity of the algorithm. Example code is given here: https://github.com/WinVector/Examples/tree/master/Count.

2 Problem

Suppose we are trying to count the number of non-negative integer solutions to the linear system $Ax = b$ where A is a m-row by n-column integer matrix, and b is an integer m-vector. We are going to assume that $x = 0$ is the unique solution to $Ax = 0$ (which is easy to check and a feature of our original application).

A section of [Mount, 2000] claimed an algorithm based on memoization and dividing solutions into a zero/one component and an even integral component. The claimed storage requirement was $n^m \log_2(B)$ sub-problems (where $B = 1 + \max_i |b_i|$). Due to space only an general argument is given. We give a precise proof that implies no more than $(n+1)^m \lceil \log_2(B) \rceil$ sub-problems need to be examined when A is a zero/one matrix (as it was in the source).

3 The Proof

Given a linear system $Ax = b$ as described above define the following:

*email: mailto:jmount@win-vector.com web: http://www.win-vector.com/

$$
\begin{aligned}
B &:= 1 + \max_i |b_i| && (1)\\
l &:= \min_{x \in \{0,1\}^n, i \in \{1,\cdots m\}} (Ax)_i && (2)\\
u &:= \max_{x \in \{0,1\}^n, i \in \{1,\cdots m\}} (Ax)_i && (3)\\
r &:= 1 + u - l && (4)\\
C_0 &:= \{b\} && (5)\\
C_i &:= \left\{ \frac{v - Az}{2} \middle| v \in C_{i-1}, z \in \{0,1\}^n \right\} \bigcap \mathbb{Z}^m \text{ for } i > 0 && (6)\\
h &:= \operatorname{argmin}_{i \in \mathbb{Z}^+} B/2^i < \frac{1}{2} && (7)
\end{aligned}
$$

In the original paper we use bounds $l \geq 0$ and $u \leq n$ (since A is a zero/one matrix with n-columns). In that application we have $r \leq n + 1$. The C_i are the sets of problems (new right-hand sides for $Ax = b'$) that are recursed on to solve the counting problem. What we want is a bound on:

$$
\text{size}(A) := \left| \bigcup_{i=0,1,2\cdots} C_i \right| \tag{8}
$$

We can show $\text{size}(A) \leq r^m(h+1)$ which in the original context establishes roughly $\text{size}(A) \leq (n+1)^m \lceil \log_2(B) \rceil$ (what the original paper should have claimed, using $h \approx \lceil \log_2(B) \rceil$).

Theorem 1. *size*$(A) \leq r^m(h+1)$

Proof. For $i \geq 0$ define:

$$
D_i := \left\{ \frac{b}{2^i} - \sum_{j=1}^{i} \frac{d_j}{2^j} \middle| d_j \in \{l, \cdots, r\}^m \right\} \bigcap \mathbb{Z}^m \tag{9}
$$

It can be shown that $|D_i| \leq r^m$ and that the D_i stop changing for $i \geq h$. So if we can show for all $i \geq 0$ we have $C_i \subseteq D_i$ then we are done.

Let's prove $C_i \subseteq D_i$ by induction on i. The base case $C_0 \subseteq D_0$ is obvious. So suppose $x \in C_i$. Then $x = \frac{v - Az}{2}$ for some $v \in C_{i-1}$ and $z \in \{0,1\}^n$. By our induction hypothesis $v \in D_{i-1}$ so $v = \frac{b}{2^{i-1}} - \sum_{j=1}^{i-1} \frac{d_j}{2^j}$ (where the d_j are all vectors in $\{l, \cdots, r\}^m$). Define $d_0 = Az$ and we have $x = \frac{b}{2^i} - \sum_{j=1}^{i} \frac{d_{j-1}}{2^j}$ which demonstrates $x \in D_i$ completing the proof. $\square$

4 An argument that doesn't work

Our proof works by ignoring a lot of the structure of the C_i (in fact we introduce a crude uniform bound on their size independent of i). A simple argument one might like to try positing that $|C_{i+1}| \leq (r/2)^m |C_i|$. This would argue the final C_h is no bigger than $((r/2)^m)^h$. Unfortunately this is treating the memoization (or dynamic programming) structures as if they were trees and there was no useful repetitions of right hand sides. So this simpler bound is ignoring far too much.

In fact for our problem we have $(((r/2)^m)^h = 2^{(\log_2(r)-1)mh} \approx 2^{(\log_2(n+1)-1)m \log_2(B)} = (\frac{n+1}{2} + B)^m > B^m$. But trivially when A is a zero/one matrix we are only interested in non-negative integral vectors $b' \leq b$, which there are no more than B^m of.

5 The nature of the bound

A bound stating we can compute the number of non-negative integer solutions of $Ax = b$ by inspecting no more than $(n+1)^m \lceil \log_2(B) \rceil$ sub-problems is pretty good. In particular it is linear in the logarithm of the magnitudes of the table margins (and not merely pseudo-polynomial or polynomial in the magnitudes of the table margins). Also for r-row by c column tables we have $n = rc$ and $m = r + c$, so we prefer a bound of the form $(n + 1)^m$ to something like 2^n.

References

[Mount, 2000] Mount, J. (2000). Fast unimodular counting. *Combinatorics, Probability and Computing*, 9:277–285.