Approximating the Permanent via Importance Sampling with Application to the Dimer Covering Problem

by
Isabel Beichl
NIST, Gaithersburg, MD 20899
isabel@cam.nist.gov
and
Francis Sullivan
IDA/Center for Computing Sciences, Bowie, MD 20715
fran@super.org

Abstract

We estimate the asymptotic growth rate of the number of dimer covers of a cubic lattice. Our estimate $\lambda_3 = 0.4466 \pm 0.0006$ is consistent with the lower bound obtained by Hammersley and the more recent improved upper bound obtained by Ciucu.

It is well known that computing λ is equivalent to computing the permanent of a certain 0-1 matrix. We describe an extremely efficient Monte Carlo algorithm for approximating the permanent. Previous work on Monte Carlo approaches includes the pioneering results of Jerrum and Sinclair, who use a rapidly mixing random walk. Our method is inspired by results of Soules on convergence of Sinkhorn balancing. We use balancing to generate an importance function that allows us to do direct random sampling, rather than a random walk that converges to a limiting distribution.

1. Introduction

We describe a new method for approximating the permanent of a 0-1 matrix based on the Monte Carlo technique of importance sampling. We use this method to solve the dimer covering problem in 2 and 3 dimensions, that is, to estimate the asymptotic behavior of the number of dimer coverings of a regular rectangular lattice. We use an excellent "importance function" that is readily available and fairly easy to compute. The importance function is the result of applying the technique of Sinkhorn balancing to a particular 0-1 matrix and to some of its minors. It is known that Sinkhorn balancing converges quickly for a completely supported matrix [23], [16] and, as we shall see, it is also easy to insure that all matrices encountered are completely supported. (All terminology is defined in the next section.)

The dimer covering problem is a special case of the more general monomer-dimer problem, where one wishes to count the number of ways of covering a lattice with both monomers and dimers. An extension of our method applies to the more general monomer-dimer setting, but the computations are much more elaborate. For the present paper, our specific goal is to estimate the asymptotic value:

$$\lambda_d = \lim_{m\to\infty} \frac{\log(perm(A_d(m)))}{m^d}$$

where d is the dimension (d is 2 or 3 in our case), and $perm(A_d(m))$ is the permanent of a particular $(m^d/2) \times (m^d/2)$ matrix, to be described shortly. Our calculation of λ_2 reproduces known analytic results. Our calculation for λ_3 gives $\lambda_3 = 0.4466$ with error bars of ± 0.0006.

The organization of the rest of the paper is as follows: in section 2 we give a history of the problem, in section 3 we explain importance sampling, in section 4 we explain Sinkhorn balancing and its relationship with the permanent, section 5 gives the details of our permanent approximation algorithm with subsections on the variance and programming details, and finally our results for λ_2 and λ_3 are in section 6.

2. Background and history of the problem

We define a *brick* to be a d-dimensional ($d \geq 2$) rectangular parallelepiped with sides whose lengths are integers. A dimer is a brick whose volume is 2. An m-brick is a brick with m units on each side. The number of different ways to fill an m-brick with dimers we call F_d. It is well-known that F_d grows exponentially with the volume of the m-brick, that is with m^d in all dimensions d, and it was proved in [7] that

$$\lim_{m \to \infty} \frac{\log(F_d)}{m^d}$$

exists. The limit will be denoted λ_d. Figure 1 shows one way to fill a 6-brick with dimers in *2-d*. To estimate F_d, we would need to find how many ways this is possible for all m-bricks.

There is extensive literature concerning calculation of λ_d. For $d = 2$, Fisher, Kastelyn and Temperley [25], [13] gave an analytic solution,

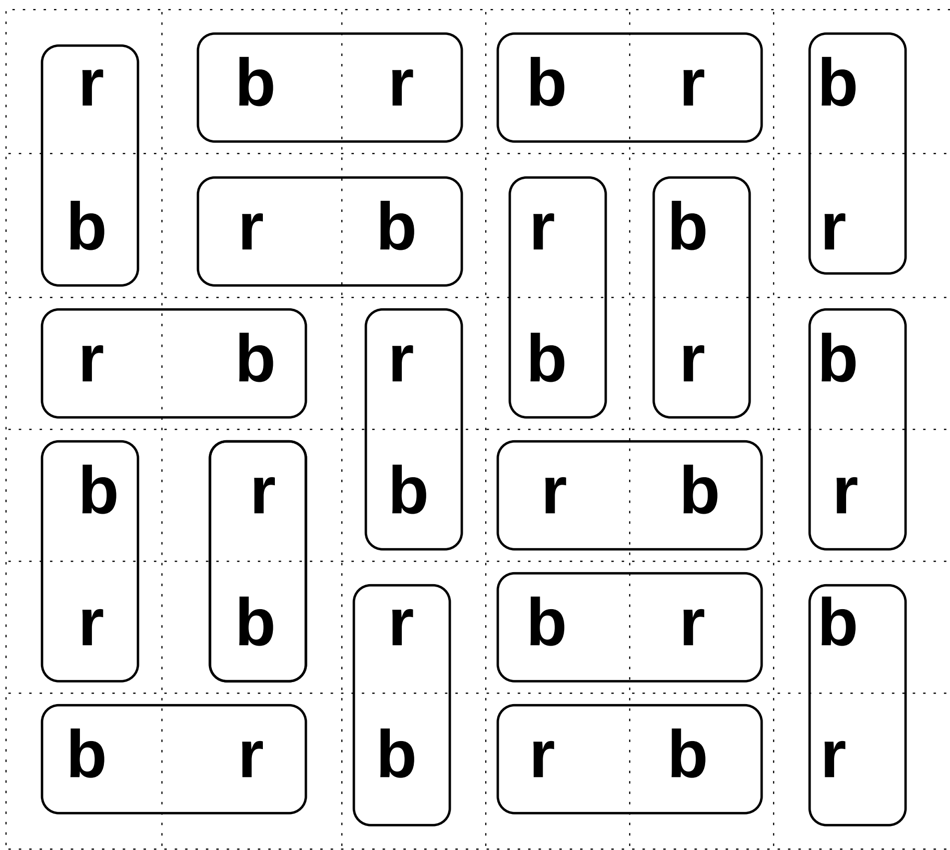
$$\lambda_2 = 0.29156090 \cdots$$

but their method does not extend to dimension 3. The early paper by Fowler and Rushbrooke [5] gave rigorous bounds $0 \leq \lambda_3 \leq 0.54931$. The upper bound was improved by Minc [17] to 0.54827 and recently by Ciucu [4] to 0.463017.

Because λ_d is a non-decreasing function of d, a lower bound for λ_3 is $\lambda_2 = 0.29156$. This was improved by Hammersley [6] to 0.418347 and by Priezzhev [19] to 0.419989. A conjecture by Schrijver and Valiant [24] on lower bounds for permanents would imply, as noted by Minc [18], that $\lambda_3 \geq 0.440075$.

It was known [6] that computing λ_d is equivalent to finding the permanent of a certain matrix A. Here's what this means: Let us consider $d = 2$ first. Think of the m-brick as a checkerboard with m squares on a side. We label the squares red and black separately as in Figure 1. There are $m^2/2$ red squares and $m^2/2$ black squares. We then form an incidence matrix A_m of size $m^2/2 \times m^2/2$ with a 1 in position i, j if red square i touches black square j, and 0 otherwise. One dimer covering of the checkerboard means a "path" through A, that is, a selection of exactly one non-zero element in each row and column. The dimer covering problem then translates into finding the number of paths through the matrix A. We assume a periodic boundary condition (= toroidal) so that every square has exactly 4 neighbors (north,

Figure 1: Figure 1: Example of one dimer covering of red (r) and black (b) sites.



south, east and west) and thus, the matrix A would have exactly 4 ones in every row and column. When $d = 3$, the m-brick is a cube with side m, volume m^3, and the matrix A is $m^3/2 \times m^3/2$ with exactly 6 ones in every row and column, because each red or black cube has neighbors on the top and bottom in addition to north, south, east and west. We will let $a_{i,j}$ denote the $(i,j)^{th}$ element of A. Unless noted otherwise, we will assume that $d = 3$ for the rest of this paper.

The number of paths though A, can be written as

$$\sum_{\sigma} \prod_{i=1}^{m^3/2} a_{i,\sigma(i)}.$$

This is the definition of the *permanent* of A, which we will also write as $|A|$. Here σ ranges over $\mathcal{S}_{m^3/2}$, the set of all permutations on $m^3/2$ letters. Note that the only terms that contribute to the sum are for those permutations σ where all $a_{i,\sigma(i)}$ are non-zero, which is exactly when the set of those $a_{i,\sigma(i)}$ form a path through the matrix. Note also, that the permanent of a matrix is similar to the determinant but it lacks the alternating signs of the terms. Computing the permanent is known to be difficult [20].

At first glance, both the permanent and the determinant seem to require $O(n!)$ operations. However, the determinant is an alternating multi-linear form and classic methods of linear algebra apply, giving algorithms to do the evaluation in $O(n^3)$ operations. Computing the permanent, however, really *is* intractable. The best known algorithm for computing the permanent exactly due to Ryser [20] requires $O(n * 2^n)$ operations. This is not a practical method for this problem because the size of the matrices necessary to find λ_3 are greater than 2000×2000 for those m's necessary to obtain statistically reliable estimates of the limit.

The matrix A arising from the monomer-dimer problem has a special, highly regular structure. Fisher and Temperley[25], and independently Kastelyn [13] used the special structure to reduce evaluating the permanent to linear algebra for the *2-d* case. It is known, however, that the *3-d* case cannot yield to the linear algebra approach. For details see Hammersley [6].

Approximating the permanent also has been studied by many authors. Karmarkar, Karp, Lipton, Lovasz and Luby [14] gave a permanent approximation algorithm whose runtime grows exponentially with matrix size. Jerrum and Sinclair [11] developed a random-walk algorithm that runs in polynomial time for some important classes of matrices. An excellent source of results using random walks can be found in the series of papers by Jerrum and Sinclair [11]. In [12], possible application is described. These methods should ultimately give an independent approximation for λ_3.

3. Importance sampling

We use direct random sampling using an importance function rather than the more widely used Markov chain random walk approach. Importance sampling is a form of Monte Carlo sampling designed to reduce the variance of the estimators for a given size sample [8].

Suppose that we want to estimate some mean value

$$\bar{f} = \frac{1}{N} \sum f(X_i)$$

where the X_i are chosen uniformly. The same result can be obtained by choosing the X_i from some non-uniform probability distribution *if* we multiply the sample by the inverse of the probability of picking that sample. If "choosing X uniformly and averging" is an unbiased estimator then "choosing X with probability $p(X)$ and averaging $f(X)/p(X)$" is also an unbiased estimator. In other words, suppose we require that the probability of picking X_i is $p(X_i)$, then, in the long run, for large N, the fraction of terms in the estimator that used X_i would be $p(X_i)$ and the mean would contain $f(X_i) * p(X_i)$. But to get the same effect as sampling uniformly instead of a term like $f(X_i)$ you would need to use $f(X_i)/p(X_i)$ giving for large N

$$\frac{1}{N} \sum_{X_i \epsilon U} f(X_i) = \frac{1}{N} \sum_{X_i \epsilon U} \frac{f(X_i)}{p(X_i)} * p(X_i) \approx \frac{1}{N} \sum_{X_i \epsilon P} \frac{f(X_i)}{p(X_i)}$$

where $X_i \epsilon U$ means X_i is chosen uniformly, and $X_i \epsilon P$ means X_i is chosen with probability $p(X_i)$.

The power of the method comes from a judicious choice of $p(X)$. The variance is reduced if $p(x)$ is chosen to be similar in shape to $f(x)/\bar{f}$. To explain our application of importance sampling we consider first a naive approach. Assume we choose σ uniformly at random from $\mathcal{S}_n$. Let $\prod a_{i,\sigma(i)}$ be the product of matrix elements along the path determined by σ. If all matrix elements chosen are equal to 1, then the value 1 is obtained by that permutation, σ. The ratio of number of samples that produce a 1, to the total number of samples will be same as the ratio of the permanent to the total number of permutations possible, namely $|A|/n!$ where n is the matrix size. To see this, suppose that $S(\sigma)$ is defined to be $\prod a_{i,\sigma(i)}$. Taking the mean of the $S(\sigma)$ and multiplying by $n!$, we get in the limit

$$n! * \frac{1}{N} * \sum S(\sigma) \rightarrow perm(A) = |A|$$

because σ is chosen uniformly and hence the probability that any random $S(\sigma)$ is equal to one is equal to the fraction of all permutations that have $S(\sigma) = 1$, i.e.

$$prob\,\{S(\sigma) = 1\} = |A|/n!.$$

However, convergence to the mean will be extremely slow, because the variance is huge, namely,

$$var^2(n! * S(\sigma)) = \frac{1}{N} \sum (n! * \prod a_{i,\sigma(i)})^2 - |A|^2$$

$$= (n!)^2 |A|/n! - |A|^2$$

$$= |A|^2 (n!/|A| - 1)$$

In our case, even the *relative* variance is an enormous number. From the work cited earlier, it is known that for an $m \times m \times m$ grid, the number of coverings satisfies the rough approximation:

$$\frac{\log(|A|)}{m^3} \sim 0.45$$

so that

$$|A| \leq (\sqrt{e})^{m^3}$$

a number vastly smaller than the factorial $(m^3!)$.

Instead of this naive method, we choose a permutation by choosing one element at a time from successive rows, using an estimate of the percentage of the paths that go through that element,(i.e. the probability that a path goes through that matrix location) given that we know the part of the path that had already been selected. We notice that if, for each A, we could evaluate the following matrix, which we will call the *matrix balance, m-bal(A)* then we would have a perfect estimate of the percentage of paths passing through each position (i, j). Here $A_{i,j}$ denotes the minor of A obtained by deleting row i and column j from the original matrix A.

$$\begin{matrix} \frac{a_{1,1}|A_{1,1}|}{|A|} & \frac{a_{1,2}|A_{1,2}|}{|A|} & \cdots & \frac{a_{1,n}|A_{1,n}|}{|A|} \\ \frac{a_{2,1}|A_{2,1}|}{|A|} & \frac{a_{2,2}|A_{2,2}|}{|A|} & \cdots & \frac{a_{2,n}|A_{2,n}|}{|A|} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{a_{n,1}|A_{n,1}|}{|A|} & \frac{a_{n,2}|A_{n,2}|}{|A|} & \cdots & \frac{a_{n,n}|A_{n,n}|}{|A|} \end{matrix}$$

Clearly, the $(i, j)^{th}$ element contains the percentage of paths that go through location (i, j). The map $A \to$ *m-bal(A)* is known as the Bregman map and has been studied by Bregman [3], Bapat [2] and Linial, Smorodnitsky and Wigderson [16]. Note that *m-bal*(A) is *doubly-stochastic*, that is, its entries are all non-negative, and all the rows sum to 1 as do all the columns. We choose a "perfect" path through the matrix as follows:

(1) In row 1, we select column j with probability

$$\frac{a_{1,j}|A_{1,j}|}{|A|}.$$

Notice that if we do select column j, then not only must $a_{1,j} \neq 0$ but also the permanent of the minor $|A_{1,j}|$ must be non-zero and so we know that there *must* be a path in the minor that along with $a_{1,j}$ gives a path in the matrix A. As terminology, $a_{i,j}$ is *supported* if there exists a path in A that goes through position (i, j). So $a_{i,j}$ is supported if and only if *m-bal*$(A)_{i,j} \neq 0$.

(2) Look at the minor $A_{1,j}$ obtained by deleting row 1 and column j from the original matrix A and matrix-balance $A_{1,j}$. In the first row (obtained from the second row of the original A matrix), an element will be look like

$$\frac{a_{2,k}|A_{(1,j),(2,k)}|}{|A_{1,j}|}$$

where $|A_{(1,j),(2,k)}|$ is the permanent of the $(n-2) \times (n-2)$ minor of A obtained by deleting row 1, column j and row 2, column k. Select one particular column k of row 2 with probability

$$\frac{a_{2,k}|A_{(1,j),(2,k)}|}{|A_{1,j}|}$$

(3) Again take the minor of A deleting rows 1 and 2 and columns j and k. Minor balance that minor. Select another column, as above.

$\vdots$

(n) Continue in this way, until there is a 1×1 matrix left. This final matrix must be a non-zero because at the previous (and every) stage of this procedure we select a non-zero element with non-zero probability. But that element is an element of A (which thus must be non-zero) times a permanent of a minor which is also non-zero.

So, if we were able to get the minor-balance of a matrix easily we would have $|A|$ exactly, namely the product of the inverse probabilities chosen at each stage. That is,

$$|A| = \frac{|A|}{|A_{1,j_1}|} * \frac{|A_{1,j_1}|}{|A_{(1,j_1)(2,j_2)}|} * \frac{|A_{(1,j_1)(2,j_2)}|}{|A_{(1,j_1)(2,j_2)(3,j_3)}|} * \cdots * |A_{(1,j_1)\cdots(n-1,j_{n-1})}|$$

Unfortunately, knowing m-$bal(A)$ is equivalent to knowing $|A|$ and so computing it is intractible [10].

There is an approximation to m-$bal(A)$ available in the Sinkhorn balance of A [22], [23]. In principle, one can Sinkhorn balance a matrix in polynomial time [16].

4. Sinkhorn Balancing

The Sinkhorn balance of a matrix, $B = s\text{-}bal(A)$, is a doubly stochastic matrix obtained from A by multiplying by diagonal matrices D and E, $B = D{*}A{*}E$. Sinkhorn [22] discusses why this is possible and Soules [23] gives convergence information. For more information on the mathematics in Sinkhorn balancing see [1], [15] In practice, instead of determining D and E directly, we Sinkhorn balance by first dividing all rows of A by their sum. This makes A row stochastic but possibly not column-stochastic. So divide all columns by their sum. This makes A column stochastic, but possibly not row stochastic. Continuing alternatively with rows and columns converges to $s\text{-}bal(A)$. Sinkhorn balancing is thus an application of the method of iterating projections in Hilbert space [16], [21].

Soules [23] shows that Sinkhorn balancing converges linearly when all elements are supported. We take advantage of this by pre-determining the unsupported elements and setting the Sinkhorn balance of those elements to zero (what unsupported elements would converge to) before doing the balancing computation. By an unsupported element we mean an element (i, j) of the matrix that is on *no* path. Recall that a path in a matrix is a selection of one non-zero element from every row in the matrix, so that every column occurs exactly once. Hopcroft & Karp [9] have an $O(n^{5/2})$ algorithm for finding paths through a matrix. It is possible to alter their algorithm so that after finding one path in the matrix deciding if another element is on a path is very easy.

4.1 Why *s-bal* approximates *m-bal*

Linial, Samorodnitsky and Wigderson [16] show that

$$|A| \leq |A|/|B| \leq e^n |A|$$

where $B = s\text{-}bal(A)$ for A $n \times n$. In other words, they show that $|A|$ can be approximated by $|A|/|B|$, no matter what the entries of A are. So if we apply this to both $|A_{i,j}|$ and $|A|$ we get that

$$\frac{|A_{i,j}|}{|A|} \approx \frac{|A_{i,j}|/|B_{i,j}|}{|A|/|B|}$$

We show that

$$\frac{|A_{i,j}|/|B_{i,j}|}{|A|/|B|} = b_{i,j}$$

which gives the desired approximation

$$\frac{|A_{i,j}|}{|A|} \approx b_{i,j}$$

This equaltiy follows from the following two simple observations.

1. The product along *any* path through B equals $|B|/|A|$.
   Proof: $B = s\text{-}bal(A)$, so $B = D * A * E$ where D and E are diagonal. So, $|B| = |D| * |A| * |E|$ and hence, for any permutation σ

   $$\frac{|B|}{|A|} = |D| * |E| = \prod_{1 \leq i \leq n} d_i \prod_{1 \leq j \leq n} e_j = \prod_{1 \leq i \leq n} d_i e_{\sigma(i)}$$

   On the other hand,

   $$\prod_{1 \leq i \leq n} b_{i,\sigma(i)} = \prod_{1 \leq i \leq n} d_i a_{i,\sigma(i)} e_{\sigma(i)} = \prod_{1 \leq i \leq n} d_i e_{\sigma(i)}$$

   where the last equality holds because A is a 0-1 matrix.

2. For all i, j,

   $$\frac{|A_{i,j}|}{|A|} = b_{i,j} \frac{|B_{i,j}|}{|B|}$$

   Proof:

   $$b_{i,j}|B_{i,j}| = b_{i,j} \left(\sum_{\bar{\sigma}} \prod_{k \neq i} b_{k,\bar{\sigma}(k)} \right)$$

   where the sum ranges over all permutations $\bar{\sigma}$ omitting i and j. Hence

   $$b_{i,j}|B_{i,j}| = \sum_{\sigma} \prod_{k=1}^{n} b_{k,\sigma(k)}$$

   where the σ range over all permutations where $\sigma(i) = j$. Therefore,

   $$b_{i,j}|B_{i,j}| = |A_{i,j}| * \frac{|B|}{|A|}$$

   because there are $|A_{i,j}|$ such product terms and by (1), each one is equal to $|B|/|A|$.

5. Procedure

So here then is the procedure.

(1) Sinkhorn balance A. This produces a doubly stochastic matrix B with $(i, j)^{th}$ element $b_{i,j}$. Because row 1 sums to 1, we can select a column , j_1, with probability b_{1,j_1}. We will write this simply as b_{j_1} because at each stage we will always select from the first row of the Sinkhorn balanced matrix.

(2) Sinkhorn balance the $(n-1) \times (n-1)$ minor A_{1,j_1} giving matrix $B^{(1)}$ with elements $b^{(1)}_{i,j}$, select a column j_2, with probability $b^{(1)}_{j_2}$ where we use the first row of $B^{(1)}$ as probabilities.

$$\vdots$$

(k) Sinkhorn balance the $(n-k+1) \times (n-k+1)$ minor of A obtained by removing the first (k-1) rows and columns $j_1, \cdots, j_{k-1}$ from A. Call the resulting matrix $B^{(k-1)}$. Select a column j_k from the first row of $B^{(k-1)}$ with probability $b^{(k-1)}_{j_k}$.

$$\vdots$$

(*n-1*) Continue until there is only a 1×1 matrix left.

Then $(1, j_1), (2, j_2), \cdots, (n, j_n)$ is a path of 1's in A. The value of $|A|$ is then approximated by the mean of terms like

$$\frac{1}{b_{j_1}} * \frac{1}{b^{(1)}_{j_2}} * \frac{1}{b^{(2)}_{j_3}} * \cdots * \frac{1}{b^{(n-2)}_{j_{n-1}}}$$

Notice that this is the same as

$$\frac{a_{1,j_1}}{b_{j_1}} * \frac{a_{2,j_2}}{b^{(1)}_{j_2}} * \frac{a_{3,j_3}}{b^{(2)}_{j_3}} * \cdots * \frac{a_{n-1,j_{n-1}}}{b^{(n-2)}_{j_{n-1}}} = \frac{f(X)}{p(X)}$$

5.1 Our Method as Importance Sampling

In our case, the sample space is $\mathcal{S}_n$ the set of permutations on n letters where n is the size of the matrix we need to evaluate. For dimension 3, n is $m^3/2$ and so, for example, for an $18 \times 18 \times 18$ cube the matrix will have size 2916×2916 and $\mathcal{S}_n = \mathcal{S}_{2916}$ will contain 2916! elements. Our estimator $f(X)/p(X)$ is the characteristic function $f(X)$ of a matrix path of A in $\mathcal{S}_n$ multiplied by $1/p(X)$, which is obtained from the product of estimates of the relative number of paths at each stage of the selection. In other words $f(\sigma) = 1$ if $a_{1,\sigma(1)}, a_{2,\sigma(2)}, \cdots, a_{n,\sigma(n)}$ are all 1's in the matrix A and $p(\sigma) = \prod b^{(k)}_{\sigma(k)}$. $\sigma(1) = j_1, \sigma(2) = j_2, \sigma(3) = j_3 \cdots$. Note that we compute the importance function as we proceed.

This is a subtle but important point. The Sinkhorn balance is not used as the estimator but rather as the importance factor for selecting and scaling the estimate. The mean converges to the permanent because the naive method converges to the permanent. How good or bad the Sinkhorn balance is as an estimate of the minor balance does not affect what the approximation converges to but rather how fast it converges. (In our case it converged fast enough to get extremely good error bars.)

5.2 : Variance

The variance, var, in this calculation is by definition

$$var^2 = \frac{1}{N}\sum(s^2 - |A|^2)$$

where N is the number of samples and s is the value obtained from a single sample, namely $p(\sigma)^{-1}$where the σ is a permutation on n letters whose selection is described in the section 5. So

$$s^2 = \left(\frac{1}{p(\sigma)}\right)^2$$

and

$$var^2 = \frac{1}{N}\sum\left(\frac{1}{p(\sigma)}\right)^2 - |A|^2.$$

The probability that a particular path σ will be chosen is $p(\sigma)$; so, in the long run

$$var^2 \to \sum_{\sigma}\frac{1}{(p(\sigma))^2} * p(\sigma) - |A|^2$$

where σ ranges over all supported paths. Simplifying,

$$var^2 \to \sum_{\sigma}\frac{1}{p(\sigma)} - |A|^2$$

$$= |A| * \left(\frac{1}{|A|} \sum_{\sigma} \frac{1}{p(\sigma)} - |A| \right)$$

But this is

$$|A| * \left(\left\langle \frac{1}{p(\sigma)} \right\rangle - |A| \right)$$

because there are $|A|$ paths. Here $\langle . \rangle$ denotes the average over *all* paths, not just paths chosen by importance sampling. Thus we have

$$var^2 = |A|^2 * \left(\frac{1}{|A|} * \left\langle \frac{1}{p(\sigma)} \right\rangle - 1 \right)$$

and the relative variance is

$$\frac{var^2}{|A|^2} = \frac{1}{|A|} \left\langle \frac{1}{p(\sigma)} \right\rangle - 1$$

If we think of $p(\sigma)$ as a probability distribution on the paths σ, we note that the relative variance is determined by the degree to which $p(\sigma)$ approximates the 'perfect' distribution which would give each path weight equal to $1/|A|$.

It is also worth noting what the variance would be if we just chose $1/b_j$ uniformly. Without importance sampling var would be

$$\frac{1}{|A|} \left\langle \frac{1}{p(\sigma)^2} \right\rangle - 1$$

a very large quantity.

5.3 : Datastructures used in permanent software

The matrix A used in this calculation is sparse. It contains $6 * n$ non-zero elements where A is $n \times n$. We make use of the sparsity by maintaining instead of the matrix A, a row-matrix and a column-matrix, which contain respectively for a given row, the column numbers of the next non-zero element and for a given column the row numbers for the next non-zero element. The actual values of the elements in the Sinkhorn balancing are kept as one array with pointers into it from the row-matrix and the column matrix.

Sinkhorn balancing a minor involves two steps, row balancing and column balancing. For our datastructure row balancing is straightforward. Column balancing is more elaborate. Because we need to keep a record of which columns of A are to be used in the minor we wish to balance, we maintain this information by using a linked list of the active columns. When we delete a column (by choosing it in a path) we do so by marking it as deleted and then on the next traversal of the active-column-list we do the delete from the list.

5.4 : Computing $log(|A|)$ when $|A|$ is not representable

One of the challenges in this calculation is that $|A|$ is not representable in floating point when the matrix size is large. For example, $\lambda_3^{(14)}$ is around 0.45, so $\log(|A_{14}|) = 0.45 * 14^2/2 \approx 617$. Thus $|A| \approx e^{617}$. However, the logs of the individual samples *are* representable and we must use these instead of actual estimates of the permanent. We want to estimate

$$\lambda_3^{(m)} = log(Perm)/m^3$$

where

$$Perm \approx \left\langle \frac{1}{p} \right\rangle = \frac{1}{N} \sum_{i=1}^{N} \prod_{j} \frac{1}{b_j}.$$

But

$$\prod \frac{1}{b_j} = \exp\left(\log \prod \frac{1}{b_j}\right) = \exp\left(\sum (-1)\log(b_j)\right)$$

All our samples of $\sum(-1)\log(b_j)$ are approximately the same size. Let C be the smallest integer contained in all of these terms. That is, C is the minimum over all samples of

$$\left\lfloor \sum (-1)\log(p_i) \right\rfloor$$

So,

$$\prod \frac{1}{b_j} = \exp(C + R_i)$$

where R_i is the remainder after C is subtracted from the log of the estimate of sample i. So,

$$\frac{1}{N} \sum_{i=1}^{N} \prod_{j} \frac{1}{b_j} = \frac{1}{N} \left(\sum_{i=1}^{N} \exp(C) * \exp(R_i) \right)$$

$$
\begin{aligned}
&= \exp(C)\frac{1}{N}\sum_i \exp(R_i) \\
&= \exp(C) * \langle \exp(R_i) \rangle .
\end{aligned}
$$

Thus,

$$
\begin{aligned}
\lambda_3^{(m)} &= \log(|A|)/m^3 \\
&\approx \log(\exp(C) * \langle \exp(R_i) \rangle)/m^3 \\
&= C + \log\left(\langle \exp(R_i) \rangle\right)/m^3
\end{aligned}
$$

Thus, it is possible to get the log of the average permanent even though each permanent is not representable by using the average of exp of the remainder which *is* representable.

6. Observations and Results

Figure 2 shows calculations for λ_2, the case known by previous analytic results to be 0.29156090. Our result agrees extremely well with the known value.

Figure 3 shows calculations for λ_3. For both 2 and 3 dimensions, we fit the input points with a quadratic. $y = \alpha + \beta/x^2$ where x are our values m and the corresponding y are the values $log(|A_m|)/m^d$, $d = 2, 3$. The α's , the limiting values, are our approximations to λ_2 and λ_3. For λ_3 then we get 0.4466 ± 0.0006.

It is interesting to observe that on the whole, for different paths, the same probabilities occur and in approximately the same proportions. In other words, the probability distribution $p(\sigma)$ is observed to concentrate near the uniform distribution $1/|A|$.

7.0 Acknowledgement

We wish to thank Dr. Eleazer Bromberg for many useful discussions and insights on the permanent. We also wish to thank Dr. George Soules for introducing us to the importance of importance sampling.

Figure 2: Figure 2: $\lambda_2^{(m)}$ values, fit with $y = \alpha + \beta/x^2$.

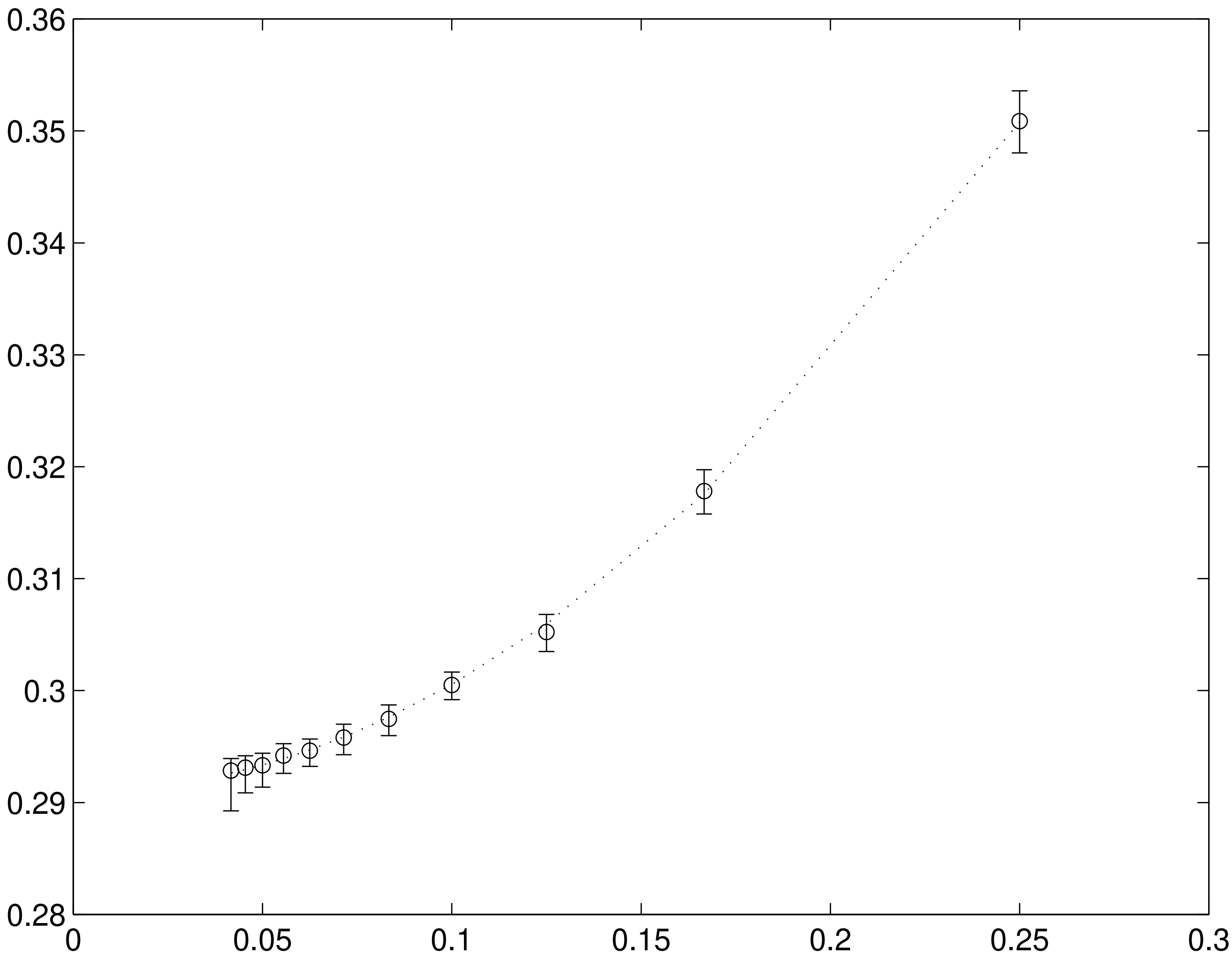
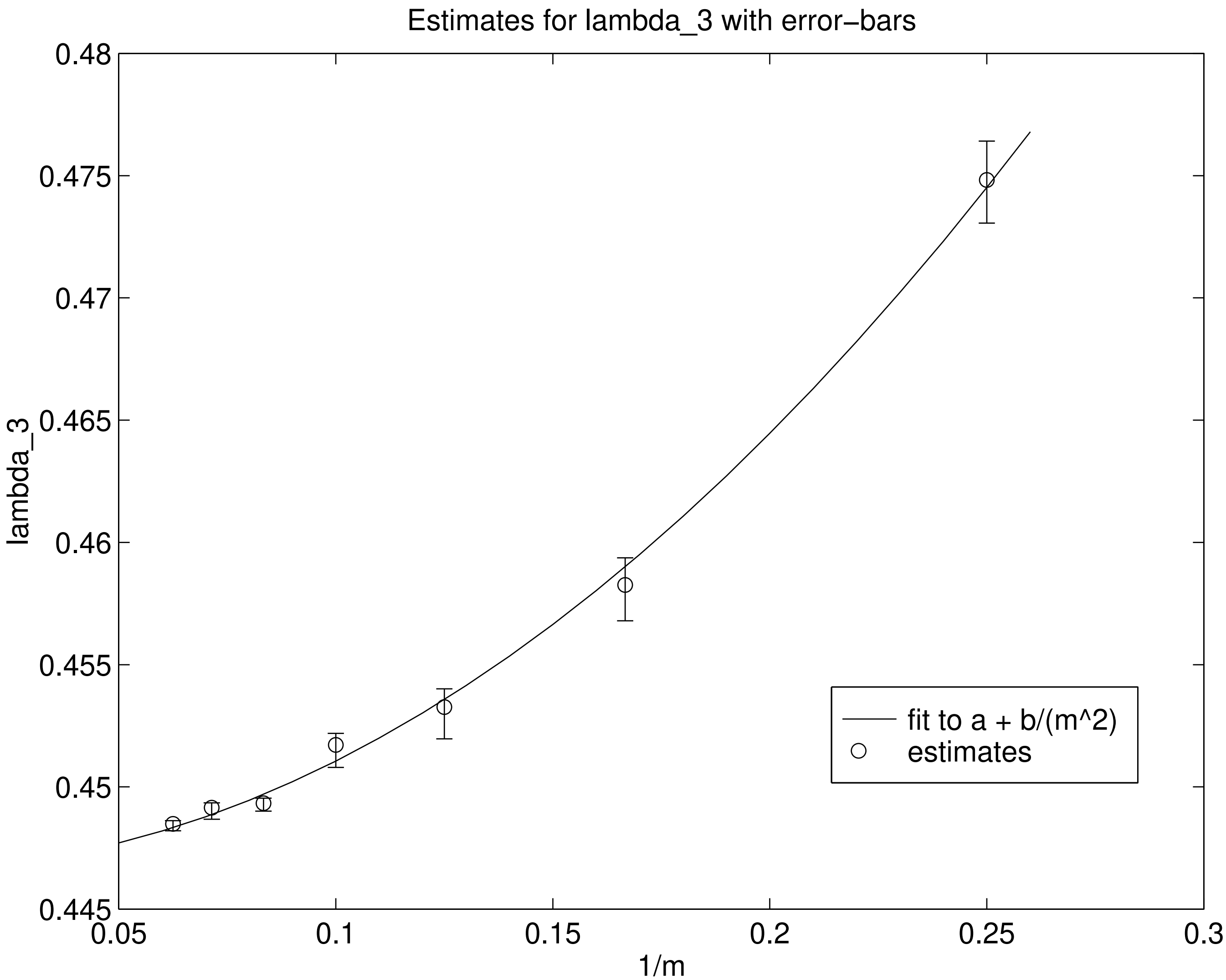


Figure 3: Figure 3: $\lambda_3^{(m)}$ values, fit with $y = \alpha + \beta/x^2$.



References.

1 Ando, T. "Majorization, Doubly Stochastic Matrices and Comparison of Eigenvalues" *Linear Algebra & Applic.* **118**, 163 (1989).

2 Bapat, R. B., "Applications of an inequality in information theory to matrices", *Linear Algebra & Applic.* **78** 107 (1986).

3 Bregman, L. M., "Proof of convergence of *Sheleikhovskii's method for a problem with transportation constraints", Zh. vychsl. Mat. mat. Fiz.* **7** 147 (1967).

4 Ciucu, M. "An improved upper bound for the three-dimensional dimer problem", to appear.

5 Fowler, R. H. & Rushbrooke, G. S. "Statistical theory of perfect solutions", *Trans. Faraday Soc.* **33** 1271 (1937).

6 Hammersley, J. M, "An improved lower bound for the multidimensional dimer problem", *Proc. Camb. Phil. Soc.* **64**, 455 (1968).

7 Hammersley, J. M. "Existence theorems and Monte Carlo methods for the monomer-dimer problem. *Research papers in statistics: Festschrift for J. Neyman* (1966), pp. 125-146.

8 Hammersley, J. M. & Handscomb, D. C. *Monte Carlo Methods* (Wiley, New York, 1964).

9 Hopcroft, J. & Karp, R., "An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs.", *SIAM J. Comp.* **2**, 225 (1973).

10 Jerrum, M.R. "Two-dimensional Monomer-Dimer Systems are Computationally Intractible", *J. Stat. Phys.* **48**, 121 (1987).

11 Jerrum, M. R. & Sinclair, A. J., "Approximating the Permanent" *SIAM J. Computing* **18**, 1149 (1989).

12 Kenyon, C., Randall, D. & Sinclair, A. "Approximating the Number of Monomer-Dimer Coverings of a Lattice" *J. Stat. Phys.* **83** 637 (1996).

13 Kastelyn, P. W., "The statistics of dimers on a lattice." *Physica* **27** 1209 (1961).

14 Karmarkar, N., Karp, R. M., Lipton, R., Lovasz, L. & Luby, M. "A Monte Carlo algorithm for estimating the permanent", *SIAM J. Computing* **22**, 284 (1993).

15 London, D. "On matrices with double stochastic pattern", *J. Math. Anal. Appl.* **34**, 648 (1971).

16 Linial, N., Samorodnitsky,A. & Wigderson, A., "A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents", to appear.

17 Minc, H. "An upper bound for the multidimensional dimer problem", *Math. Proc. Cambridge Philos. Soc.*, **83**, 461 (1978).

18 Minc, H. Review of the paper "On lower bounds for permanents" by A. Schrijver and W. G. Valiant, *Math Reviews* (1982) article 82a:15004,63.

19 Priezzhev, V. B., "The statistics of dimers on a three-dimensional lattice II. An improved lower bound", *J. Stat. Phys.*, 829 (1981).

20 Ryser, H., *Combinatorial Mathematics* (Math. Assoc. Amer. Carus Monographs, 1963) **14**.

21 Sullivan, F. and Atlestam, B. "A descent method with smooth rotund spaces with applications to approximation in L_p", *J. Math. Anal. and Applic.* **48** (1974).

22 Sinkhorn, R., "A relationship between arbitrary positive matrices and double stochastic matrices" *Annals of Math. Stat.* **35**, 876 (1964).

23 Soules, G. W., "The rate of convergence of Sinkhorn balancing" *Linear Algebra & Applic.* **150**, 3 (1991).

24 Schrijver, A. and Valiant, W. G., "On lower bounds for permanents" *Nederl. Akad. Wetensch. Indag. Math.* **42**, 425 (1980).

25 Temperley, H. N. V. & Fisher, M. E. "Dimer problem in statistical mechanics - an exact result." *Philos. Mag.* **6**, 1061 (1961).