Comput Stat (2014) 29:1793–1798
DOI 10.1007/s00180-014-0506-1

SHORT NOTE

Permanents, α-permanents and Sinkhorn balancing

Francis Sullivan · Isabel Beichl

Received: 17 April 2013 / Accepted: 15 May 2014 / Published online: 28 June 2014

Abstract The method of Sinkhorn balancing that starts with a non-negative square matrix and iterates to produce a related doubly stochastic matrix has been used with some success to estimate the values of the permanent in some cases of physical interest. However, it is often claimed that Sinkhorn balancing is slow to converge and hence not useful for efficient computation. In this paper, we explain how some simple, low cost pre-processing allows one to guarantee that Sinkhorn balancing always converges linearly. We illustrate this approach by efficiently and accurately computing permanents and α-permanents of some previously studied matrices.

Keywords Matrix scaling · Doubly stochastic matrix · Sequential importance sampling

1 Introduction

The notion of Sinkhorn balancing for non-negative square matrices was introduced in Sinkhorn (1964) and some of the theory was developed in Knopp and Sinkhorn (1967). Our purpose here is to explain why, contrary to an apparently wide-spread opinion, Sinkhorn balancing is a tool useful when applying sequential importance sampling (SIS) to estimate the permanent and the alpha-permanent efficiently. As has been pointed out several times in the literature (Kou and McCullagh 2009; Liu 2001), when

F. Sullivan
IDA/CCS, 17100 Science Drive, Bowie, MD 20715, USA
e-mail: fran@super.org

I. Beichl (✉)
NIST, 100 Bureau Drive, Gaithersburg, MD 20899, USA
e-mail: isabel.beichl@nist.gov

used directly "out of the box" Sinkhorn balancing can be very slow to converge and hence is seemingly impractical for such computations. In fact, one can easily construct examples for which the rate of convergence is worse than logarithmic. However, by using some simple pre-processing that requires little more than depth-first search, a linear rate of convergence can always be guaranteed (Soules 1991; Tassa 2012). We will explain in Sect. 2. We explain the pre-processing in Sect. 3 and illustrate its use by reproducing some of the computational results reported in Kou and McCullagh (2009) along with information about the relative variance of our sampling via SIS for these examples in Sect. 4. As we shall see, basing importance sampling on Sinkhorn balancing provides a robust and efficient approach to this class of approximation problems.

2 Convergence of Sinkhorn balancing

The Sinkhorn balancing algorithm itself is simple to describe. One is given a non-negative matrix A and the aim is to derive a unique doubly-stochastic matrix B from A. To do this, we iterate: first divide each row of A by its sum giving a row-stochastic matrix. We call this row-balancing. Then divide the columns of the resulting matrix by their sums giving a column stochastic matrix. We call this column-balancing. Continue until convergence. The matrix B is unique in that it has non-zeros only at some (but perhaps not all) locations (i, j) where $a_{i,j} > 0$ and the non-zeros of B maximize a generalized form of entropy (Soules 1991). Here convergence for an $n \times n$ matrix, A, means that the row sums converge in L^∞ norm to the n-vector of all ones. This is assuming, of course, that columns are normalized after row sums. In other words, if we end by column balancing then the row sums should all be close to 1. That is,

$$||A_k \cdot e - e||_\infty < \epsilon$$

where A_k is the matrix that results from k iterations of row-balancing and column-balancing of A, e is the vector of all ones and ϵ is a small number. As a practical matter, doing n^2 iterations of row and column balancing, as described above, is usually more than enough to get convergence to a reasonable ϵ for the purposes of computing the permanent or α-permanent.

In Knopp and Sinkhorn (1967), it is shown that this iteration converges quickly if and only if every non-zero element of A has *support*, meaning that there exists a permutation of the rows and columns of A that element on the diagonal and places non-zeros on the other elements of the diagonal. In the case of unsupported elements, Sinkhorn balancing still converges but is not guaranteed to be quick.

Note that this fact in itself already points toward a connection between Sinkhorn balancing and computing the permanent because the number of such permutations is equal to the permanent in case A is a matrix of zeros and ones. If A is not a zero-one matrix we have that

$$|A| = \sum_{\sigma} \prod_{i=1}^{n} a_{i,\sigma(i)}$$

Here $|A|$ denotes the permanent and we sum over *supported* permutations σ, i.e. those that have all $a_{i,\sigma(i)}$ non-zero. For the case of the α-permanent where α is not equal to one, the formula is:

$$|A| = \sum_{\sigma} \alpha^{cyc(\sigma)} \prod_{i=1}^{n} a_{i,\sigma(i)}$$

where $cyc(\sigma)$ is the number of cycles in σ.

A second result found in Knopp and Sinkhorn (1967) is that there exist diagonal matrices D and E such that $B = D \cdot A \cdot E$ if and only if A has *total* support. Total support means that for every non-zero $a_{i,j}$ in A, there exists a permutation σ such that for some i, $\sigma(i) = j$ and

$$\prod_{i=1}^{n} a_{i,\sigma(i)} > 0$$

More generally, each such non-zero element in A is said to be supported and all others are non-supported. Only supported elements contribute to the permanent and B has non-zeros only at locations i, j where $a_{i,j}$ is supported. If all non-zeros are supported, A is said to have total support.

If A has total support then, by a result of Soules (1991), convergence is linear. But without total support, convergence can be extremely slow. If, for example, A is a matrix with ones on and above the main diagonal then only elements on the diagonal are supported and convergence is slower than logarithmic. (see Fig. 1).

3 Approximating the permanent and the α-permanent

Our strategy is to find all the supported elements and remove the non-supported ones before balancing and then use the entries of B to generate an importance function. To find the supported elements we use an algorithm introduced in Tassa (2012).

To explain Tassa's method for determining support, it is helpful to think of our matrix, A, in two ways: (1) as the adjacency matrix of an undirected bipartite graph, where the rows are one color and the columns another, and, (2) as the adjacency matrix of a directed graph where the arcs go from rows to columns.

If A is regarded as the adjacency matrix of a bipartite graph, the permanent is the number of perfect matchings of the graph. But, when regarded as a directed graph, the permanent of an $n \times n$ zero-one matrix A is equal to the number of distinct cycle covers of an associated directed graph G (Ben-Dor and Halevi 1993). The vertex set of G is the set $\{1, 2, \ldots n\}$ and there is a directed edge from vertex i to j if and only if $a_{ij} = 1$. (Note that a non-zero on the diagonal of A indicates a loop.) A cycle cover is a set of vertex-disjoint cycles that includes all of the vertices of G. Clearly a non-zero a_{ij} is supported if and only if it is in at least one cycle that is part of a cycle cover. (To see this, list the vertices in the order in which they occur in the cycles of the cycle cover. Because every vertex is included, this gives a permutation σ such that $\prod_{i=1}^{n} a_{i,\sigma(i)} > 0$.)

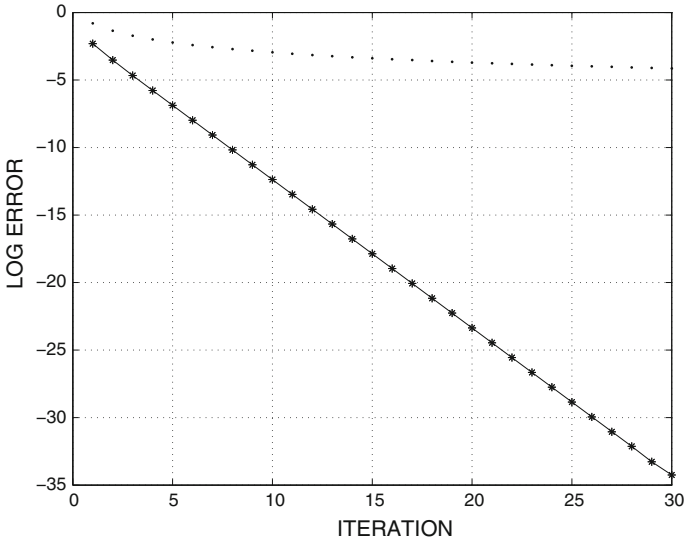

Fig. 1 Convergence of Sinkhorn balancing for a 7×7 matrix with 6 unsupported elements. *Line* indicates that support has been pre-determined. *No line* indicates without support pre-determined. Here "error" is $||A_k \cdot e - e||_\infty$ as in Sect. 2

If A is thought of as a bipartite graph, we first find a perfect matching using the Hopcroft-Karp algorithm. For an $n \times n$ matrix having m non-zeros, this requires $O(m\sqrt{n})$ operations. If there is no perfect matching, the permanent is zero and there is nothing to do. If there is a perfect matching, we can permute rows and columns so that the diagonal consists of all ones. Then, switching back to directed graphs, this means that one cycle cover consists entirely of loops. Other cycle covers must use at least one directed cycle that is not a loop, meaning they must use edges that are in strongly connected components of the graph obtained if we ignore the loops. Edges in strongly connected components can be determined using a single application of Tarjan's algorithm for finding strongly connected components (Tarjan 1972) which requires $O(m)$ operations. Thus, after running the Tarjan algorithm, if an edge is in a strongly connected component it is supported. Otherwise it is not.

In Beichl and Sullivan (1999) Sinkhorn balancing is used as the basis of a sequential importance sampling approach to estimation of permanents in order to estimate the so-called dimer covering constant. Use of Sinkhorn balancing is suggested by the following basic identity about the Sinkhorn balanced matrix, B, corresponding to a totally supported matrix A.

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

Here $|A_{i,j}|$ and $|B_{i,j}|$ denote the permanents of the i, j minors of A and B respectively. In very special cases, the ratios $|B_{i,j}|/|B|$ are all equal to one (Ando 1989) but in general they can be expected to be close to one.

Table 1 Relative errors and relative variance

Matrix	Size	Density (%)	Estimate	Relative error	Relative variance
A2	20×20	92.8	3.5285e+32	0.040	0.205
A3	15×15	100.0	1.4448e+22	0.010	0.252
A4	15×15	78.8	7.08694e+21	0.008	0.317
A5	20×20	100.0	3.2956e+49	0.002	0.187
A6	20×20	32	5.9738e+40	0.005	0.976

We generate a supported permutation sequentially as follows: first choose a column j from row one selected with probability $b_{1,j}$ (and thus importance $1/b_{i,j}$). Next delete row one and column j from A and repeat the process on the resulting matrix $(n-1) \times (n-1)$ matrix, etc. At each step, before balancing we zero out the non-supported entries of the current matrix. It is easy to see that we have

$$|A| = \mathcal{E}\left(\frac{\prod_{i=1}^{n} a_{i,\sigma(i)}}{\prod_{i=1}^{n} b_{i,\sigma(i)}}\right)$$

where $\mathcal{E}$ denotes expected value. Note that if the ratios $|B_{i,j}|/|B|$ were actually equal to one, a single sample would suffice to give an exact answer. In fact, because we will use the $b_{i,j}$ as the probability distribution from which we draw, the mean values of the ratios for the i, j selected will always equal one.

In the cases in which α is different from one we proceed similarly to the method used in Kou and McCullagh (2009). As the permutation is built, the current number of cycles is part of the developing importance function. We begin by choosing a row from column $j = 1$. But we change the importance function so that instead of selecting row i with probability $b_{i,j}$ we select with probability proportional to $\alpha^x b_{i,j}$ where $x = 1$ if selecting row i completes a cycle and $x = 0$ if not. The next column is chosen to have index i unless i has appeared before. In that case, a cycle has been completed and we select the next column at random from amount those still available.

4 Numerical results

In Table 1, we present a few numerical results to compare with those reported in Kou and McCullagh (2009). All computations were done using Matlab.[1] The matrices referred to as A2, . . ., A6 are taken from http://www.stat.uchicago.edu/~pmcc/reports/matrices

The errors are computed using the exact permanents given on the above web page. For a fixed matrix, a sample consists of a supported permutation σ and the importance

[1] Certain commercial equipment, instruments, or materials are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

value used at each step of building σ. The number of iterations to do Sinkhorn balancing was 20 and the number of samples drawn was 10,000 in all cases.

5 Conclusions

Sinkhorn balancing is an effective method for estimating the permanent and the α-permanent. In order to be efficient, it is necessary to pre-determine support which can be done in polynomial time.

References

Ando T (1989) Majorization, doubly stocastic matrices and comparison of eigenvalues. Linear Algebra Appl 115:163–248

Beichl I, Sullivan F (1999) Approximating the permanent via importance sampling with application to the dimer covering problem. J Comput Phys 149:128–147

Ben-Dor A, Halevi S (1993) Zero-one permanent is #p-complete, a simpler proof. In: Proceedings of the 2nd Israel symposium on the theory and computing systems, pp 108–117

Knopp P, Sinkhorn R (1967) Concerning non-negative matrices and doubly stochastic matrices. Pac J Math 21:343–348

Kou S, McCullagh P (2009) Approximating the α permanent. Biometrika 96:635–644

Liu JS (2001) Monte Carlo strategies in scientific computing. Springer, New York

Sinkhorn R (1964) A relationship between arbitrary positive matrices and doubly stochastic matrices. Ann Math Stat 35:876–879

Soules G (1991) The rate of convergence of sinkhorn balancing. Linear Algebra Appl 150:3–40

Tarjan R (1972) Depth-first search and linear graph algorithms. SIAM J Comput 1:146–160

Tassa T (2012) Finding all maximally-matchable edges in a bipartite graph. Theor Comput Sci 423:50–58