Scaling matrices and counting the perfect matchings in graphs

Ioannis Panagiotas — **ENS de Lyon**
F. Dufossé — **Inria Grenoble**
K. Kaya — **Sabanci University, Istanbul**
B. Uçar — **CNRS and LIP, ENS de Lyon**

Abstract

We investigate efficient randomized methods for approximating the number of perfect matchings in bipartite graphs and general graphs. Our approach is based on assigning probabilities to edges. We show an unbiased estimator and bound its second moment. We also present practical experiments.

1 Introduction

We investigate efficient randomized methods for approximating the number of perfect matchings in bipartite graphs and general graphs. Our main tool is a sparse matrix scaling algorithm which we apply to the adjacency matrix of the given graph (bipartite or general) to assign probabilities to the edges and base random choices to these probabilities.

Counting perfect matchings in a bipartite graph is equivalent to computing the permanent of its adjacency matrix, which is defined as $Per(\mathbf{A}) = \sum_{\sigma} \prod a_{i,\sigma(i)}$, where the summation runs over all permutations σ of $1, \ldots, n$, which is #P-Complete [12]. While we focus on the case where $\mathbf{A}$ is a 0-1 matrix, our techniques are also applicable to matrices containing real values. Approximating the permanent is a well-studied problem. Jerrum et al. [5] discuss a Markov Chain Monte Carlo (MCMC) based approach which can provide a $(1+\epsilon)$-approximation for the permanent in fully polynomial time, with $\tilde{O}(n^{10})$ complexity. The MCMC based approach cannot be generalized easily to the general graph case (Štefankovic et al. [11] highlight the difficulties). Gurvits and Samorodnitsky [4] and Linial et al. [7] use matrix scaling to propose deterministic approximations with exponential guarantees (2^n and e^n respectively).

Our contribution is more on the practical side, as we describe an algorithm with significantly reduced complexity. We propose an alternative selection mechanism to the original algorithm due to Rasmussen [9]. The same approach is also studied by Beichl and Sullivan [1], but we offer some new insights as well as a more detailed experimental analysis.

We omit proofs and refer the reader to the full version of the manuscript, available in Hal [2].

2 Notation and background

Matrices are shown in bold upper case letters, e.g., $\mathbf{A}$. With $\mathbf{A}_{ij}$ we denote the submatrix of matrix $\mathbf{A}$ obtained by deleting the ith row and the jth column. The entries in the matrix are shown with lower-case letters and subscripts, e.g., $a_{i,j}$. Vectors are shown with bold, lower-case roman letters, e.g., $\mathbf{v}$. Components of a vector are shown with the same letter and subscripts, e.g., v_i. We use $\mathbb{E}[\cdot]$ to denote expectation. The base of the natural logarithm is shown with e.

We say that X' for a quantity P achieves (ϵ, δ)-approximation whenever $\Pr(|X' - P| \leq \epsilon P) \geq 1 - \delta$. In general, an (ϵ, δ) approximation can be achieved by simulating $O\left(\frac{\mathbb{E}[X^2]}{\mathbb{E}[X]^2} \cdot \frac{1}{\epsilon^2} \cdot \log(\frac{1}{\delta})\right)$ trials. For this reason, the fraction $\frac{\mathbb{E}[X^2]}{\mathbb{E}[X]^2}$ is called the critical ratio.

An $n \times n$ matrix is said to have *support* if there is a perfect matching in the associated bipartite graph. If all the non-zeros belong to at least one perfect matching then the matrix has *total support*. Any nonnegative matrix $\mathbf{A}$ with total support can be scaled with two (unique) positive diagonal matrices $\mathbf{R}$ and $\mathbf{C}$ such that $\mathbf{RAC}$ is doubly stochastic, that is each column and each row sums to one. The Sinkhorn-Knopp algorithm [10] is a well-known method for this purpose. When $\mathbf{A}$ is scaled with $\mathbf{R}$ and $\mathbf{C}$ to be doubly stochastic, $g(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{A} \mathbf{y} - \sum_{i=1}^{n} \ln x_i - \sum_{j=1}^{n} \ln y_j$ attains its minimum value for positive x_i and y_i at $\mathbf{x} = \mathrm{diag}(\mathbf{R})$ and $\mathbf{y} = \mathrm{diag}(\mathbf{C})$ [6, Proposition 2].

Lemma 1. *Let $\mathbf{A}$ be an $n \times n$ 0-1 matrix with total support. Consider $\mathbf{R}, \mathbf{C}$ diagonal scaling matrices such that $\mathbf{S} = \mathbf{RAC}$ is a doubly stochastic. Then, $Per(\mathbf{S}) = Per(\mathbf{A}) \cdot \prod_{i=1}^{n} r_i \cdot c_i$. In addition, $r_i \cdot c_j = \dfrac{Per(\mathbf{A}_{ij})}{Per(\mathbf{A})} \cdot \dfrac{Per(\mathbf{S})}{Per(\mathbf{S}_{ij})}$.*

3 Algorithm and analysis

The proposed algorithm to estimate the permanent is shown in Algorithm 1. It takes an $n \times n$, 0-1 matrix $\mathbf{A}$ (with nonzero permanent), and produces a random variable $X_{\mathbf{A}}$. The algorithm proceeds in n steps. Initially $X_{\mathbf{A}}$ is equal to one. At every step, the algorithm adds a nonzero entry to a matching, thus obtaining a perfect matching at the end. At step i, a nonzero entry in the ith row of $\mathbf{A}$ is chosen among those columns of $\mathbf{A}$ which have not been matched yet. That entry defines the matching edge for the ith row. The nonzeros are selected according to the values of the entries in the doubly stochastically scaled version of the remaining matrix. The random variable $X_{\mathbf{A}}$ is multiplied by the reciprocal of the value of the chosen nonzero. We discard the nonzeros in $\mathbf{A}^{(i)}$ that cannot be put into a perfect matching (with Dulmage-Mendelsohn decomposition [8]). The worst case run time of the algorithm can be bound as $\tilde{O}(mn^3 \ln n)$ with the Sinkhorn-Knopp algorithm used for scaling, where the terms including the deviation of the row/column sums from one are hidden, n is the size of the matrix and m is the number of nonzeros.

Algorithm 1 PERMANENT ESTIMATION

Require: $n \times n$, $0-1$ matrix $\mathbf{A}$
Ensure: Permanent estimate $X_{\mathbf{A}}$
1: $X_{\mathbf{A}} \leftarrow 1$
2: $\mathbf{A}^{(1)} \leftarrow \mathbf{A}$
3: **for** $i = 1$ **to** n **do**
4: Filter out those entries of $\mathbf{A}^{(i)}$ that cannot be put into a perfect matching
5: $[\mathbf{R}^{(\sigma,i)}, \mathbf{C}^{(\sigma,i)}] \leftarrow \textsc{Scale}(\mathbf{A}^{(i)})$
6: Pick a nonzero column j in the first row of $\mathbf{A}^{(i)}$ by using the probability density function $p_j = \frac{s_{1,j}}{\sum_{k \in \mathbf{A}^{(i)}(1,:)} s_{1,k}}$ for all nonzeros $a_{1,j}^{(i)}$ where $\mathbf{S} = \mathbf{R}^{(\sigma,i)} \mathbf{A}^{(i)} \mathbf{C}^{(\sigma,i)}$
7: $X_{\mathbf{A}} \leftarrow X_{\mathbf{A}} / p_j$
8: $\mathbf{A}^{(i+1)} \leftarrow \mathbf{A}_{1j}^{(i)}$ {delete the first row and the jth column of $\mathbf{A}^{(i)}$}

The algorithm identifies a perfect matching at the end. Let $\mathbf{R}^{(\sigma,i)}$ and $\mathbf{C}^{(\sigma,i)}$ denote the scaling matrices at the ith step of the algorithm, where the perfect matching σ is returned. Let also $\sigma_1^{(i)}$ denote the column chosen for the first row at the ith step.

Theorem 1. *Let $X_{\mathbf{A}}$ be a random variable returned by Algorithm 1 for the estimate of the permanent of an $n \times n$, 0–1 matrix $\mathbf{A}$. Then $\mathbb{E}[X_{\mathbf{A}}] = Per(\mathbf{A})$.*

Lemma 2. *Let* $\mathbf{A}$ *be* $n \times n$ *matrix such that* $a_{1,j} = 1$ *and* $\mathbf{A}_{1j}$ *be the minor after discarding entries not in any permutation. Let* $\mathbf{R}$, $\mathbf{C}$, $\mathbf{D}$, *and* $\mathbf{F}$ *be the positive diagonal matrices scaling* $\mathbf{A}$ *and* $\mathbf{A}_{1j}$ *respectively. Then,* $\prod_{i=2}^{n} r_i \cdot \prod_{i=1, i\neq j}^{n} c_i \leq e^{r_1 c_j - 1} \cdot \prod_{i=1}^{n-1} d_i \cdot f_i$

Theorem 2. *Let* $\mathbf{A}$ *be* $n \times n$ *matrix with total support, and* $\mathbf{RAC}$ *be its doubly stochastic scaling. Then*

$$\mathbb{E}[X_{\mathbf{A}}^2] \leq Per(\mathbf{A}) \cdot \frac{e^{v-n}}{\prod_i r_i \cdot c_i} \quad \textit{with } v \leq n \textit{ or} \quad \mathbb{E}[X_{\mathbf{A}}^2] \leq Per(\mathbf{A}) \cdot mean\Big(\sum_{\sigma} e^{\sum_j r_1^{(\sigma,j)} c_{\sigma_1^{(j)}}^{(\sigma,j)}}\Big) \cdot \frac{e^{-n}}{\prod_i r_i \cdot c_i} .$$

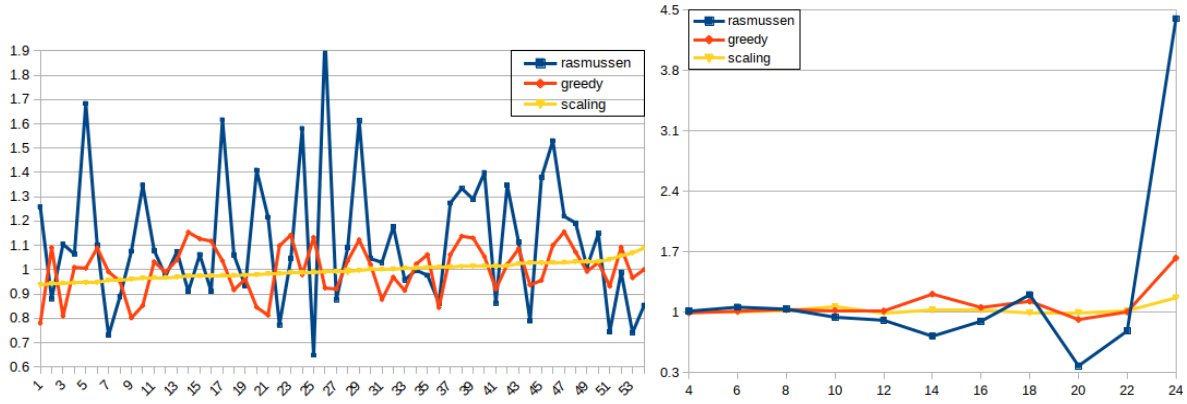
Rasmussen [9] investigates a simpler approach in which the selection probabilities are uniform, and no edges are discarded (so that the estimator can return zero). Beichl and Sullivan [1] propose the same algorithm and perform some analysis, where the $\mathbb{E}[X_{\mathbf{A}}^2]$ is bound using a notation to denote the average value of $\frac{1}{\prod_{i=1}^{n} r_1^{(\sigma,i)} c_{\sigma_1^{(i)}}^{(\sigma,i)}}$ over all perfect matchings. This analysis is valuable to show that scaling helps, but it does not yield computable bounds. Our analysis in Theorem 2 gives efficiently computable bounds. The variant of the algorithm for graphs has the same properties. In particular with X_G showing the random variable and $M(G)$ showing the number of perfect matchings in G, we have $\mathbb{E}[X_G^2] \leq M(G) \cdot \frac{1}{\prod_i s_i}$, where s_is are the diagonal entries of the scaling factors of the adjacency matrix of G. Fürer and Kasiviswanathan [3] discuss three randomized algorithms (Simple, REP, and Greedy) similar to ours. The one called Simple is a direct adaptation of Rasmussen's algorithm for graphs. REP extends Simple such that at certain points during the execution, it creates k copies where each copy selects its own neighbor; the procedure continues as before in each copy. The results of each copy are later combined. Greedy attempts to assign probabilities in a better way by setting probabilities inversely proportional to the column degrees.

4 Experiments and conclusion

To see how the proposed heuristic fares in practice, we compare it against the original estimator of Rasmussen as well as Greedy. Additionally for all three variants in this paper use a small heuristic and select at each step the row with the least nonzeros remaining. For each test, we take 1000 samples and report their mean. Initially we consider random matrices of size 40 and sparsity $\frac{4}{n}$. Such matrices can have large permanents (e.g., around 10^7) and are among the largest an exact algorithm can handle. The results which are summarized in Figure 1(a) showcase that our approach always has good performance. In contrast, Rasmussen's estimator often reports results that are a lot worse than the other two approaches (even if modified to avoid returning zeros). Greedy exhibits better performance than Rasmussen's, while our approach has better performance than Greedy (it has smaller and less frequent deviation from the value of the permanent).

To provide results with larger n, we focus on the class of matrices which represent grids and for which an exact formula for the permanent is known. The results presented in Figure 1(b) concur with the previous test, suggesting that the assignment of probabilities via scaling makes the overall procedure more reliable.

We have proposed a technique for approximating the permanent based on matrix scaling. We manage to prove loose yet computable upper bounds for $\mathbb{E}[X_A^2]$. The experimental analysis suggests that an improvement over previous similar methodologies is possible. Future work involves bounding the estimated factors for special graph classes.



(a) sparse 40×40 matrices (b) grid graphs

Figure 1: The approximation ratio for the three approaches in 54 random graphs with $n = 40$ and $4/n$ sparsity factor (in increasing order of the approximation ratio of our estimator), and on square grids of even length as the length increases.

References

[1] I. Beichl and F. Sullivan. Approximating the permanent via importance sampling with application to the dimer covering problem. *J. Comput. Phys.*, 149(1):128–147, 1999.

[2] F. Dufossé, K. Kaya, I. Panagiotas, and B. Uçar. Scaling matrices and counting perfect matchings in graphs. Technical Report RR-9161, Inria - Research Centre Grenoble – Rhône-Alpes, 2018.

[3] M. Fürer and S. P. Kasiviswanathan. Approximately counting perfect matchings in general graphs. In *ALENEX/ANALCO*, pages 263–272, 2005.

[4] L. Gurvits and A. Samorodnitsky. Bounds on the permanent and some applications. *arXiv preprint arXiv:1408.0976*, 2014.

[5] M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM*, 51(4):671–697, 2004.

[6] B. Kalantari and L. Khachiyan. On the complexity of nonnegative-matrix scaling. *Linear Algebra Appl.*, 240:87–103, 1996.

[7] N. Linial, A. Samorodnitsky, and A. Wigderson. A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents. *Combinatorica*, 20:545–568, 2000.

[8] A. Pothen and C.-J. Fan. Computing the block triangular form of a sparse matrix. *ACM T. Math. Software*, 16:303–324, 1990.

[9] L. E. Rasmussen. Approximating the permanent: A simple approach. *Random Struct. Algor.*, 5(2):349–361, 1994.

[10] R. Sinkhorn and P. Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific J. Math.*, 21:343–348, 1967.

[11] D. Štefankovič, E. Vigoda, and J. Wilmes. On counting perfect matchings in general graphs. *arXiv preprint arXiv:1712.07504*, 2017.

[12] L. G. Valiant. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8(2):189–201, 1979.