The permanent of an \( n \times n \) matrix \( A \) is defined as \( \text{per}(A) \triangleq \sum_{\pi \in S_n} \prod_{i=1}^{n} a_{i\pi(i)} \), where \( S_n \) is the set of all permutations over \( \{1, \ldots, n\} \). Computing \( \text{per}(A) \) is a hard problem, even finding \( \text{sign}(\text{per}(A)) \) is \( \text{NP} \)-complete. The question of approximating the permanent of a matrix has applications in various domains including statistical physics, where it is used in the Ising model, and in quantum physics and quantum computing. Computing the permanent is one of the candidate problems to prove Quantum supremacy.

In the specific case when \( a_{i,j} \in \{0,1\} \) and \( A \) is the adjacency matrix of a general graph \( G \), \( \text{per}(A) \) is equal to the number of cycle covers. For a bipartite graph \( G(U,V) \) let \( A = \{a_{ij}\} \) be the binary matrix where \( a_{ij} = 1 \) if and only if vertex \( i \) in \( U \) is adjacent to vertex \( j \) in \( V \). Then, \( \text{per}(A) \) is equal to number of perfect matching of \( G \). Computing \( \text{per}(A) \) is \#P-hard \cite{Valiant}. In fact, Valiant introduced the notion of \#P-hardness for this particular problem. There are both deterministic and randomized algorithms for approximating the permanent. Today, our focus will be on deterministic approaches.

## 1 Useful Inequalities

We will be using 2 key inequalities regarding the permanent of a matrix. We begin with one conjectured by Van Der Waerden, later proved to be correct in several papers. In the next lecture, we will see an elegant proof of this lower bound by Gurvits using real stable polynomials.

**Theorem 1 (Van Der Waerden Inequality)** If \( A \) is a doubly stochastic matrix then

\[
e^{-n} \sim \frac{n!}{n^n} \leq \text{per}(A) \leq 1.
\]

**Proof:** We prove the upper bound by noting that since the entries of \( A \) are non-zero we have:

\[
\text{per}(A) = \sum_{\pi \in S_n} \prod_{i=1}^{n} a_{i\pi(i)} \leq \prod_{i=1}^{n} \sum_{j=1}^{n} a_{ij} = 1.
\]

\( \square \)

**Remark 2** This upper bound is satisfied when \( A \) is the identity matrix. The lower bound is satisfied when all entries of \( A \) are equal to \( \frac{1}{n} \), where \( n \) is the dimension of the matrix.

**Remark 3** Restricting matrix \( A \) to have a bounded number of non-zero entries can result in better lower bounds.

**Remark 4** The reason the lower bound is bounded away from zero is that doubly stochastic matrices are convex combination of permutation matrices.

Next, the Bregman-Minc inequality gives us a non-trivial upper bound for \( \text{per}(A) \).
Theorem 5 (Bregman-Minc Inequality) Let $A \in \{0, 1\}^{n \times n}$ be a square binary matrix and let $r_i \triangleq \sum_{j=1}^{n} a_{i,j}$ for all $i$, then

$$\text{per}(A) \leq \prod_{i=1}^{n} (r_i!)^{1/r_i}.$$  

Example 1 For a 2-regular bipartite graph $G(X, Y)$ with $|X| = |Y| = n$, let $A$ be the matrix with $a_{i,j} = 1$ if and only if vertex $i \in X$ is connected to vertex $j \in Y$ (the adjacency matrix). Then, the Bregman-Minc inequality gives the following bound:

$$\text{per}(A) \leq 2^{n/2}.$$  

The lower bound on $\text{per}(A)$ is achievable when $G$ is a cycle, and the upper bound is achievable when $G$ is collection of $n/2$ cycles of length 4.

Example 2 For an $r$-regular bipartite graph $G$, where $r = \alpha n$, the Bregman-Minc inequality gives:

$$\text{per}(A) \leq (r)^{n/r} \sim (\sqrt{2\pi r})^{\alpha/2} (\frac{r}{e})^n.$$  

Let

$$D \triangleq \begin{bmatrix} \frac{1}{\sqrt{r}} & \cdots & \frac{1}{\sqrt{r}} \end{bmatrix}$$

then we have that $DAD$ is a doubly stochastic matrix. Therefore, by the Van Der Waerden inequality, $r^{-n} \text{per}(A) \geq e^{-n}$. Note that in the case that $\alpha = O(1)$, the gap between Van Der Waerden and Bregman-Minc inequality is polynomial in $n$.

2 Sequential Importance Sampling

Given a rooted tree $T$ we can sample one of its leaves by starting from the root and iteratively choosing a child of the current node uniformly at random until we reach a leaf. A sequential importance sampling algorithm outputs the product of the branching factor as the weight of the corresponding leaf.

For a given leaf $X$, let $br(X)$ be branching factor of $X$, which is equal to the product of the out-degree of all nodes on the path from the root to $X$. Note that $1/br(X)$ is the probability of reaching $X$ with the proposed algorithm. So, $\{1/br(X)\}$ form a probability distribution.

Proposition 6 Let $T$ be a rooted tree and let $L$ be the set of its leaves. Define $br(X)$ for leaf $X$ as above. Then

$$\sum_{X \in L} \frac{1}{br(X)} = 1.$$  

Also, we have the following

Proposition 7 With notation similar to Proposition 6 let $N = |L|$, then we have,

$$\mathbb{E}[br(X)] = N.$$
Proof:

\[
\mathbb{E}[br(X)] = \sum_{X \in L} \frac{1}{br(X)} \cdot br(X) = N
\]

Consider the case where we are given a bipartite graph \( G(U, V) \) and want to sample one of its perfect matchings (if it has any) uniformly at random. Since the problem of counting the number of matchings is \(#P\)-hard, we can only give an approximate algorithm.

One way to generate a perfect matching of a bipartite graph is to sequentially construct a partial matching until we have a perfect matching. To do so, we iterate over vertices in \( U \), where for vertex \( u \in U \) we find all of \( u \)'s neighbors \( v \) such that after matching \( v \) to \( u \) the resulting partial matching is part of some perfect matching. We call such neighbors \( v \) valid.

Algorithm 1: Sequential importance sampling for generating a perfect matching

1. **Input:** Bipartite graph \( G(U, V) \);
2. Find a perfect matching \( X \) of graph \( G \);
3. Let \( \pi \) be random permutation on \( U \);
4. Let \( X = \emptyset \), \( B = 1 \);
5. for \( i = 1, \ldots, n \) do
6. \hspace{1em} Let \( \tilde{N}(\pi(i)) \) be the set of valid neighbors of \( \pi(i) \);
7. \hspace{1em} Choose vertex \( j \) from \( \tilde{N}(\pi(i)) \) uniformly at random;
8. \hspace{1em} \( X = X \cup (\pi(i), j) \);
9. \hspace{1em} \( B = B \cdot |\tilde{N}(\pi(i))| \);
10. \hspace{1em} Remove \( \pi(i) \) and \( j \) from \( G \);
11. end
12. **Output:** matching \( X \), and branching factor \( B \).

**Lemma 8** Given a bipartite graph \( G \), assume \( M \) is the set of all perfect matching of \( G \), so that \( |M| = N \). Let \( X \) be the (random) matching that is chosen by a run of Algorithm 1. Let \( Y \) be a matching that is chosen uniformly at random from the set of all perfect matchings in \( G \). If we consider their branching factors, then we have that

\[
\mathbb{E}( \log(br(X))) \leq \log(N) \leq \mathbb{E}( \log(br(Y)))
\]

**Proof:** First, we prove the upper bound on \( \mathbb{E}( \log(br(X))) \):

\[
\mathbb{E}( \log(br(X))) = - \sum_{X \in M} \frac{1}{br(X)} \cdot \log \left( \frac{1}{br(X)} \right) \\
\leq \left( \sum_{X \in M} - \frac{1}{br(X)} \right) \cdot \log \left( \frac{1}{N} \sum_{X \in M} \frac{1}{br(X)} \right) \\
= \log(N),
\]

where the middle inequality is due to Jensen’s inequality for the concave function \( x \log(x) \) and the last equality is true because of Remark 6.

To prove the upper bound we use Jensen’s inequality for the concave function \( \log(x) \):
\[ \mathbb{E}(\log(\text{br}(Y))) = -\sum_{Y \in \mathcal{M}} \frac{1}{N} \log \left( \frac{1}{\text{br}(Y)} \right) \geq \log \left( \frac{1}{N} \sum_{Y \in \mathcal{M}} \frac{1}{\text{br}(Y)} \right) = \log(N). \]

Now, we are ready to prove the Bregman-Minc inequality.

**Proof:** Let \( G(U, V) \) be the bipartite graph corresponding to a binary matrix \( A \) (adjacency matrix). By Lemma 8 we know that \( \log(N) \leq \mathbb{E}(\log \text{br}(Y)) \) where the expectation is over the randomness in \( Y \), which is selected uniformly at random from the set of all perfect matchings. Fix a matching \( M \) and a permutation \( \pi \) on vertices in \( U \). Let \( a(M, \pi, i) \) be the number of valid neighbors of vertex \( u = \pi(i) \) at step \( i \) of Algorithm 1 where \( \pi(1), \ldots, \pi(i-1) \) are matched as in \( M \). Each vertex in the neighborhood of \( u \), \( N(u) \), by definition must have a match through \( M \). If \( u \) appears after \( j \) of these vertices with matches in \( N(u) \), then defining the degree of vertex \( u \) in \( G \) as \( r_u \) we have that at most \( r_u - j \) of \( u \)'s neighbors are left unmatched. Therefore, in this worst case scenario \( a(M, \pi, i) \leq r_{\pi(i)} - j \). Now, note that when the permutation \( \pi \) is chosen uniformly at random, \( u \) appears in each position with equal probability. Hence,

\[ \mathbb{E}_{\pi} \log(a(M, \pi, i)) \leq \frac{1}{r_{\pi(i)}} \sum_{j=0}^{r_{\pi(i)}-1} \log(r_{\pi(i)} - j) = \frac{\log(r_{\pi(i)}!)}{r_{\pi(i)}}. \]

Putting together the above arguments we have that:

\[ \log N \leq \mathbb{E}_{\pi} \mathbb{E}_M(\log \text{br}(Y)) = \mathbb{E}_M \mathbb{E}_{\pi} \sum_{i=1}^{n} \log(a(M, \pi, i)) \leq \mathbb{E}_M \sum_{i=1}^{n} \mathbb{E}_{\pi} \log(a(M, \pi, i)) \leq \mathbb{E}_M \sum_{i=1}^{n} \frac{1}{r_{\pi(i)}} \sum_{j=0}^{r_{\pi(i)}-1} \log(r_{\pi(i)} - j) = \mathbb{E}_M \sum_{i=1}^{n} \frac{\log(r_{\pi(i)}!)}{r_{\pi(i)}} = \sum_{i=1}^{n} \frac{\log(r_{\pi(i)}!)}{r_{\pi(i)}}. \]

Which gives us the desired result. \( \square \)
3 Convex Optimization

Given a matrix $A$ with positive entries, is there a way to scale the rows and columns of $A$ to get a doubly stochastic matrix? In other words, do there exist $\alpha_1, \ldots, \alpha_n$, and $\beta_1, \ldots, \beta_n$ so that $[\alpha_i a_{i,j} \beta_j]$ form a doubly stochastic matrix? In this section we consider this question, and the examine how to find these coefficients $\{\alpha\}$ and $\{\beta\}$ if they exist.

**Theorem 9** If $A$ is a matrix with positive entries, and $\text{per}(A) \neq 0$ then there exist diagonal matrices $D_\alpha \triangleq \text{diag}(\alpha_1, \ldots, \alpha_n)$ and $D_\beta \triangleq \text{diag}(\beta_1, \ldots, \beta_n)$ so that $D_\alpha A D_\beta$ is a doubly stochastic matrix.

**Remark 10** When all entries of $A$ are strictly positive, by iteratively normalizing rows then columns to sum to 1, we will converge to a doubly stochastic version of $A$ (that is, find $D_\alpha$ and $D_\beta$) \[1\].

Next, we show an alternate way of finding $D_\alpha$ and $D_\beta$, by solving a convex program.

**Proof:**

$$
\begin{align*}
\text{maximize} & \quad f(x) = \sum_{i,j \leq n} x_{i,j} \ln \left( \frac{a_{i,j}}{x_{i,j}} \right) \\
\text{subject to} & \quad \sum_{j=1}^n x_{i,j} = 1, \ i = 1, \ldots, n \\
& \quad \sum_{i=1}^n x_{i,j} = 1, \ j = 1, \ldots, n \\
& \quad x_{i,j} \geq 0.
\end{align*}
$$

Now, by taking partial derivatives of $f(x)$ we see that $f$ is a strictly concave function. Hence, the maximum is unique and inside the polytope. By writing out the KKT conditions, if $\alpha_i$ and $\beta_j$ are dual variables of constraints $\sum_{j=1}^n x_{i,j} = 1$ and $\sum_{i=1}^n x_{i,j} = 1$ respectively, we have that:

$$
\ln \left( \frac{a_{i,j}}{x_{i,j}} \right) - \alpha_i - \beta_j = 1
$$

which results in $x_{i,j} = e^{-1-\alpha_i a_{i,j} e^{-\beta_j}}$. Since, $x_{i,j}$ forms a doubly stochastic matrix by primal constraints, we have the result. \[ \square \]

We can generalize the above convex program to the case where $a_{i,j}$ can be zero. Let $f(z) = \prod_{i=1}^n (\sum_{j=1}^n a_{i,j} z_i)$. Note that since $\text{per}(A)$ is simply the square free terms in $f(z)$, we have that:

$$
\inf_z \frac{f(z)}{z_1 \ldots z_n} \geq \text{per}(A).
$$
References
