MS&E 319: Matching Theory
Instructor: Amin Saberi
Lecture date: 9/26/2019 **Scribe: Matt Tsao**

1 Polynomial Identity Testing

Suppose we want to determine whether two given polynomials (e.g. $(x-1)(x+1)$ and x^2-1) are identical. One way would be to expand each polynomial and compare coefficients. A simpler method is to evaluate both polynomials at several points to see if they agree. If they do not, we now have a *certificate* of their inequality, otherwise with good confidence we can say they are equal. This idea is the basis of a randomized algorithm.

We can formulate the polynomial equality verification of two given univariate polynomials F and G as determining whether their difference $H(x)F(x) - G(x)$ is identically zero. Denote the maximum degree of F and G as n. Assuming that $F(x) \neq G(x)$, $H(x)$ will be a polynomial of degree at most n and therefore it can have at most n roots. Choose an integer x uniformly at random from $\{1, 2, \ldots, nm\}$. $H(x)$ will be zero with probability at most $1/m$, i.e the probability of failing to detect that F and G differ is "small". After just k repetitions, we will be able to determine with probability $1 - 1/m^k$ whether the two polynomials are identical i.e. the probability of success is arbitrarily close to 1.

The following result, known as the *Schwartz-Zippel* lemma, generalizes the above observation to polynomials on more than one variables:

Lemma 1 *Suppose that F is a polynomial in variables $(x_1, x_2, \ldots, x_n)$, and that F is not identically zero. For $1 \leq i \leq n$, let d_i be the degree of $F(\cdot)$ in x_i. Also, for $1 \leq i \leq n$, let I_i be a finite subset of elements in the domain of x_i. Then the number of roots of $F(\cdot)$ in set $I_1 \times \cdots \times I_n$ is at most:*

$$\left(\sum_{i=1}^{n} \frac{d_i}{|I_i|}\right) \prod_{i=1}^{n} |I_i|$$

Proof: The case $n = 1$ is obvious since a nonzero polynomial of degree d can have at most d real roots. We proceed by induction on n. Let F' be the polynomial on $x_2, \ldots, x_n$, a polynomial on at most $n-1$ variables. If $(y_2, \cdots y_n)$ is not a zero of F', $F(x_1, y_2 ..... y_n)$ has at most d_1 zeros in I_1. By inductive hypothesis we have a bound on the number of roots of F' in $I_2 \times \cdots \times I_n$. It follows that the total number of zeros of F in $I_1 \times \cdots \times I_n$ is bounded by

$$d_1(|I_2| \cdots |I_n|) + \left(\left(\sum_{i=2}^{n} \frac{d_i}{|I_i|}\right) \prod_{i=2}^{n} |I_i|\right) |I_1|$$

which gives the desired bound. □

2 Perfect Matchings in Bipartite Graphs

Given a bipartite graph G(U,V,E) where $|U| = |V| = n$, define the matrix A as follows:

$$a_{ij} = \begin{cases} 1 & \text{if } i \sim j,\ i \in U,\ j \in V \\ 0 & \text{otherwise.} \end{cases}$$

where $\sim$ means vertices i and j are adjacent in G. Recall that a *matching* is a set of edges such that no two have a vertex in common. A *perfect matching* is one that covers all vertices of G.

Lemma 2 *If $det(A) \neq 0$ then G has a perfect matching.*

Proof: Recall the definition of determinant:

$$det(A) = \sum_{\pi \in \Pi} sgn(\pi) \prod_{i=1}^{n} a_{i\pi(i)}$$

where Π is the set of all permutations of $\{1, \ldots, n\}$. Recall that a permutation of size n is just a bijection from $\{1, \ldots, n\} \mapsto \{1, \ldots, n\}$. The sign $sgn(\pi)$ of a permutation π is $+1$ if the number of inverted pairs is even and it is -1 if the number of inverted pairs is odd (pair i and j are inverted in π if $i < j$ and $\pi(i) > \pi(j)$). One can also see a permutation π as describing a perfect matching in a bipartite graph: for each vertex $i \in U$, we match it to vertex $\pi(i) \in V$.

For a bipartite graph G with adjacency matrix A, the value of $\prod_{i=1}^{n} a_{i\pi(i)}$ will be non-zero if and only if all terms $a_{i\pi(i)}$ are nonzero, i.e. each $(i, \pi(i))$ is an edge of G, so π describes a perfect matching in G. Since the determinant is the sum of these terms, it follows that if $det(A)$ is nonzero there must exist at least one perfect matching in G. □

Since computing the determinant of a matrix is easy, this gives us a simple test for determining if G has a perfect matching. However, this only gives us a sufficient condition, not a necessary one. It is possible that G has many perfect matchings, but has equal numbers of ones with odd and even permutations, leaving $det(A) = 0$. For example, the complete bipartite graph has a perfect matching but its adjacency matrix is all 1's matrix, which is rank deficient. So how can we modify this so that it is also a necessary condition?

For variables x_{ij} define the matrix B such that

$$b_{ij} = \begin{cases} x_{ij} & \text{if } i \sim j,\ i \in U,\ j \in V \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 3 *$det(B) \neq 0$ if and only if G has a perfect matching.*

Proof: $\Rightarrow$: Choose the permutation π that corresponds to a nonzero term $\prod_{i=1}^{n} b_{i\pi(i)}$ in $det(B)$. Then $\{i, \pi(i)\}$ for $i = 1, \ldots, n$ gives a perfect matching.
$\Leftarrow$: set all x_{ij} corresponding to edges in a perfect matching to 1 and the rest to 0. It follows that $det(B) \neq 0$. □

This suggests the following algorithm to determine whether G has a perfect matching. We just need to see if a multivariate polynomial of degree at most n is equivalent to 0. There can be up to $n!$ terms in the determinant of B, but we can apply the "randomized polynomial identity testing" given in section 1 to design an efficient algorithm.

Algorithm 1. Randomized algorithm to detect a perfect matching:

1. Set x_{ij} to be a number chosen uniformly at random from $\{1, \ldots, n^2 m\}$
2. Compute $det(B)$
3. If $det(B) = 0$ repeat until confidence is above the desired threshold

The polynomial corresponding to $det(B)$ is of degree one in each x_{ij}; there are at most n^2 variables, thus the probability that we choose a root is at most $1/m$ in one trial. As before, k trials yield a probability $1 - 1/m^k$ of failure.

3 Perfect Matching in General Graphs

For a given graph $G(V, E)$ and variables x_{ij} define the *Tutte matrix* T as follows:

$$t_{ij} = \begin{cases} x_{ij} & \text{if } i \sim j,\ i > j \\ -x_{ji} & \text{if } i \sim j,\ i < j \\ 0 & \text{otherwise.} \end{cases}$$

The intuition is that while a bipartite graph has no odd cycles, a general graph G might. For this reason, in a general graph, not every permutation π such that $\{i, \pi(i)\}$ is an edge in G will correspond to a perfect matching. The Tutte matrix addresses this problem by ensuing that all odd cycles cancel each other out. To be more precise, we state the following lemma.

Lemma 4 *$det(T) \neq 0$ iff G has a perfect matching.*

Proof: $\Leftarrow$: Since G has a perfect matching $|V| = n$ is even. Given perfect matching M with each edge and (arbitrarily) ordered pair (i, j). Set $x_{ij} = 1$ for $(i, j) \in M$ and $i > j$, otherwise set $x_{ij} = 0$. Let π be a permutation such that for each $(i, j) \in M$ we have $i = \pi(j)$ and $j = \pi(i)$. Now consider the term $\prod_{i=1}^{n} t_{i\pi(i)}$ of $det(T)$. It is is clearly equal to 1. Moreover, all other terms are zero, therefore $det(T) \neq 0$.

$\Rightarrow$: Note that a permutation π can be decomposed into a collection of cycles where each element exchanges places with the next. For example the permutation $2, 1, 3$ of $1, 2, 3$ is decomposed into $\sigma_1\sigma_2 = (3)(1, 2)$. Any permutation on n elements can be uniquely expressed as a collection of disjoint cycles.

Suppose permutation π has odd cycle $\sigma = (i_1, i_2, \ldots, i_r)$, i.e r is odd, so that $i_{k+1} = \pi(i_k)$. Consider permutation π' which is the same as permutation except that the cycle σ is reversed, i.e. $i_k = \pi'(i_{k+1})$. It is not hard to check that

$$sgn(\pi) \prod_{i=1}^{n} t_{i\pi(i)} = -sgn(\pi') \prod_{i=1}^{n} t_{i\pi'(i)}$$

since we negate an odd number of terms in the product corresponding to π' by definition of T but the sign of the two permutations remains the same. Then when evaluating $det(T)$ we note that permutations with odd cycles cancel out. (note that cycles of length 1 evaluate to zero above since we assume G is simple).

Thus since $det(T) \neq 0$, T must have a permutation whose corresponding cycle decomposition consists only of even cycles. Note that each such cycle corresponds to a perfect matching over the vertices in it. Since the cycles are always disjoint we have a perfect matching in G. □

We can use Algorithm 1 to detect a perfect matching in a general graph by using matrix T instead of B.

Next we present the work of [MVV87] that extends the randomized algorithm for detecting the existence of a perfect matching in a graph to actually finding a perfect matching. The key computational step will be matrix inversion.

4 The Isolating Lemma

First we establish a key (and surprising) property of subsets of random numbers:

A *set system* (S, F) consists of a finite set S of elements, $S = \{x_1, x_2, \ldots, x_n\}$ and a family F of nonempty subsets of S, i.e. $F = \{S_1, \ldots, S_k\}$ such that $S_j \subseteq S$ and $S_j \neq$ for $j = 1, \ldots, k$. For each element of S, assign weight w_i to x_i, where w_i is chosen uniformly at random and independently from $\{1, 2, 3, \ldots, 2n\}$. Denote the weight of set S_j to be $\sum_{x_i \in S_j} w_i$.

Lemma 5 (The Isolating Lemma) *The probability that there is a unique minimum weight set is at least* $1/2$.

Proof: Fix the weight of all elements except x_i. Given F, define the *threshold* for element x_i to be the number α_i such that if $w_i > \alpha_i$ then x_i is in no set with minimum weight, and if $w_i \leq \alpha_i$ then x_i is in some set with minimum weight. Clearly, if $w_i < \alpha_i$, then x_i is in *every* set with minimum weight. The only ambiguity occurs when $w_i = \alpha_i$. In this case we say that x_i is *singular*.

A key observation is that the weight of element x_i is *independent* of the threshold value α_i. Since w_i is chosen uniformly at random from $\{1, \ldots, 2n\}$, the probability that an element is singular is at most $\frac{1}{2n}$. If no element is singular, then the subset with minimum weight is unique. Since S contains n elements, the probability that S contains a singular element is at most $n \cdot \frac{1}{2n} = 1/2$. Thus, with probability at least $1/2$, there exists no singular element, implying the lemma: □

5 Finding a Perfect Matching in a Bipartite Graph

Given a bipartite graph $G(U, V, E)$, assign to each edge $\{i, j\} \in E$ a weight w_{ij} chosen uniformly and independently from $\{1, \ldots, 2m\}$, where $m = |E|$. By the isolating lemma, the minimum weight perfect matching in G will be unique with probability at least $1/2$.

As in the previous lecture, we define the matrix B such that

$$b_{ij} = \begin{cases} x_{ij} & \text{if } i \sim j,\ i \in U,\ j \in V \\ 0 & \text{otherwise.} \end{cases}$$

Set each $x_{ij} = 2^{w_{ij}}$. Let B_{ij} be the matrix obtained from B by removing the i^{th} row and j^{th} column. Now, suppose that there is a perfect matching, call it M, and furthermore, suppose it has a unique minimum weight W. Recall from the previous lecture that $det(B) \neq 0$ if and only if there exist a perfect matching in the graph G. We can now state two useful lemmas regarding the value of $det(B)$.

Lemma 6 *$det(B) \neq 0$ and the highest power of 2 that divides $det(B)$ is 2^W.*

Proof: Since we've assumed the existence of M we must have $det(B) \neq 0$. Also, recall that by definition

$$det(B) = \sum_{\pi \in \Pi} sgn(\pi) \prod_{i=1}^{n} b_{i\pi(i)}$$

Let π_M be the permutation corresponding to M, with $\prod_{i=1}^{n} b_{i\pi_M(i)} = 2^W$. The value of every other permutation is either 0 or $2^{W'}$ with $W' > W$. If we factor out 2^W then all the terms in the sum are even numbers except for the term corresponding to permutation π_M, which is 1. Thus $det(B) = 2^W r$ where r is an odd number. □

Lemma 7 *The edge (i,j) belongs to M if and only if $det(B_{ij})/\,2^{W-w_{ij}}$ is odd.*

Proof: Note that

$$det(B_{ij})2^{w_{ij}} = \sum_{\pi:\pi(i)=j} sgn(\pi)\prod_{i=1}^{n} b_{i,\pi(i)}$$

Let π_M be the permutation corresponding to M, so π_M appears in the sum above. As in the proof of Lemma 6, its weight is 2^W and all other permutations will either have weight 0 or $2^{W'}$ with $W' > W$. Therefore, 2^W will be the highest power of 2 which divides the right hand side. The result of that operation makes the right hand side odd and the forward direction follows.

Now, if $(i,j) \notin M$, then all permutations π in the sum will have weight zero or $2^{W'}$ with $W' > W$. It then follows that dividing both sides by 2^W leaves the right hand side even. This implies the backwards direction. □

Using Lemma 6 we can recover W by calculating the determinant of B, computing the highest power of 2 that divides its value and taking log base 2. Lemma 7 gives us a way of recovering the edges which belong to M. For each edge we have to compute $det(B_{ij})$ and check if $det(B_{ij})/2^{W-w_{ij}}$ is odd. This may seem difficult but we have a nice way of doing this via matrix inversion. Recall that the adjugate of a matrix B, denoted $adj(B)$ is a matrix in which entry ij is given by $(-1)^{i+j}det(B_{ji})$, the transpose of the cofactor matrix of B. Now we use the following useful result from linear algebra.

$$adj(B) = B^{-1}det(B)$$

By inverting B we can recover the values $det(B_{ij})$ for each edge, and test if that edge is in the matching M. The following is an algorithm summarizing the procedure.

Algorithm 2. Randomized algorithm to find a perfect matching:

1. Compute $det(B)$ and obtain W and let $M = \{\}$
2. Compute $adj(B)$ and recover each $det(B_{ij})$
3. For each $\{i,j\} \in E$ if $det(B_{ij})/2^{W-w_{ij}}$ is odd and add $\{i,j\}$ to M.

Since M will be unique with probability at least $1/2$, we only need to run this algorithm a constant number of times to have an arbitrarily high probability of success!

6 Finding a Perfect Matching in a General Graph

Here, we consider any undirected simple graph $G(V,E)$ where $|V| = n$ (note, the number of vertices was $2n$ in the bipartite case). Recall from last lecture the definition of the Tutte matrix T of G:

$$t_{ij} = \begin{cases} x_{ij} & \text{if } i \sim j,\ i > j \\ -x_{ij} & \text{if } i \sim j,\ i < j \\ 0 & \text{otherwise.} \end{cases}$$

As in the bipartite case, set each $x_{ij} = 2^{w_{ij}}$, where w_{ij} is chosen uniformly at random and independently from $\{1,\ldots,2m\}$ where $m = |E|$.

We have shown in the previous lecture that

$$det(T) = \sum_{\pi \in \Pi} sgn(\pi) \prod_{i=1}^{n} t_{i\pi(i)}$$

does not equal zero if and only if G has a perfect matching. Recall that any permutation π on n elements can be uniquely expressed as a collection of disjoint cycles $\pi = \sigma_1 \sigma_2 \dots \sigma_k$. We have shown that permutations with an odd length cycle in their cycle decompositions cancel out and do not contribute to $det(T)$. Thus we only need to consider permutations containing even length cycles. Now, consider a perfect matching M of G. We state two useful lemmas.

Lemma 8 *There exists a permutation π_M corresponding to M which has a cycle decomposition composed only of cycles of length 2.*

Proof: This is easy to see since each cycle of length 2 contains two elements which, since the cycles must be disjoint, correspond to vertices that are endpoints of edges in M. □

Lemma 9 *Any permutation π'_M corresponding to M with a cycle of length greater than 2 increases $det(T)$ more than some π_M from lemma 8.*

Proof: To see this let $\sigma = (i_1, i_2, \dots, i_r)$ be a cycle of even length in permutation π'_M. Consider the corresponding cycles in π_M which are either $(i_1, i_2) \dots (i_{r-1}, i_r)$ or $(i_r, i_1) \dots (i_{r-2}, i_{r-1})$. Say that the first has a product over t_{ij} with value a^2 and the second has value b^2. These values are powers of 2 since each edge is in the product exactly twice i.e. x_{ij} and $-x_{ij}$. Then the product over σ is

$$ab = \prod_{j=1}^{r} t_{i_j i_{j+1}} .$$

Note that $\min\{a^2, b^2\} \leq ab$ so one of the matchings π_M made up of only cycles of length 2 will have have less contribution to $det(T)$. □

Now let M be a prefect matching of minimum weight W. By lemmas 8 and 9, a permutation π_M corresponding to M must have a term $(-1)^{n/2} 2^{2W}$ in the sum in $det(T)$ and all other terms will be of the form $2^{2W'}$ with $W' > W$. Thus we can conclude lemmas analogous to lemmas 6 and 7, stated here without proof.

Lemma 10 *$det(T) \neq 0$ and the highest power of 2 that divides $det(T)$ is 2^{2W}.*

Lemma 11 *The edge $\{i, j\}$ belongs to M if and only if $det(T_{ij}) / \, 2^{2W - w_{ij}}$ is odd.*

The algorithm to find the minimum matching in general graphs is the same as algorithm 2 except we replace B by T and W by $2W$.

7 Finding Maximum Matchings

Algorithm 2 can be generalized to find maximum matchings in graphs that do not have perfect matchings. Suppose we want to determine whether a graph $G(V, E)$ has a matching of size $\lfloor \frac{n}{2} \rfloor - k$ where $|V| = n$. Consider a new graph $G_k(V_k, E_k)$ obtained by introducing new

vertices $U_k\{u_1, u_2, ..., u_k\}$ and connecting them to all members of V so that $V_k = V \cup U_k$ and $E_k = E \cup \{(u, v)\}_{u \in U, v \in V}$. By construction, G has a matching of size $\lfloor \frac{n}{2} \rfloor - k$ if and only if G_k has a perfect matching. To recover a matching M of G from a perfect matching M_k of G_k, simply discard all edges of M_k incident to U_k. Therefore, a perfect matching in G_k for the smallest k for which G_k has a perfect matching gives a maximum matching of G.

8 Finding perfect matchings with parallel computation

The NC complexity class comprises problems that can be solved by a deterministic algorithm using $O(\log^{c_1} n)$ time and $O(n^{c_2})$ processors where n is the size of the input, for some constants c_1, c_2. The RNC complexity class does not require the algorithm to be deterministic, so long as the output is correct with high probability. Since matrix inversion can be parallelized, algorithm 2 shows that finding a perfect matching is in the RNC complexity class. There is no known algorithm for perfect matchings that is in NC, and whether randomization is necessary for efficient parallel algorithms is still an open question. Recently [ST17] made progress on this problem by showing that finding perfect matchings is in *quasi*-NC.

References

[MVV87] Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.

[ST17] Ola Svensson and Jakub Tarnawski. The matching problem in general graphs is in quasi-nc. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 696–707, 2017.