CSE599s: Extremal Combinatorics October 19, 2011

Lecture 7 Sunflowers, Cliques and a Brief Tour of Circuit Complexity

Lecturer: Anup Rao

1 Avoiding Cliques

Here we prove a couple of results that initiated the study of extremal graph theory. Recall that a clique of size k is a set of vertices that are pairwise connected by edges.

Question: How many edges can a graph on n vertices have without having a clique of size k?

To prove this kind of theorem, it is a good idea to keep in mind the extremal object that we are shooting for. We start with Mantel's theorem, which has to do with avoiding triangles:

Theorem 1 (Mantel)**.** *If a graph on $2n$ vertices has n^2+1 edges, then it must have a triangle.*

The above theorem shows that n^2 is the right answer: the complete bipartite graph with bipartitions of equal size will have n^2 edges and no triangle.

Proof We prove the result by induction on n. When $n = 1$, it is at most 1 edge is possible, so the theorem is vacuously true.

Now suppose the graph has $2(n+1)$ vertices and $(n+1)^2+1 = n^2+2n+2$ edges. Let x, y be adjacent vertices in the graph, and let H denote the subgraph induced on the remaining $2n$ vertices. If H contains n^2+1 edges, then H must contain a triangle by induction, so H must have at most n^2 edges. Thus, there are at least $2n+2$ edges that are not contained in H. One of those edges is between x, y, but the rest ($\geq 2n+1$) are from $\{x, y\}$ to H. By the pigeonhole principle, two of these edges must originate from the same vertex v. Then we see that v, x, y form a triangle. ■

Next we discuss the case of larger cliques. We shall prove the following theorem due to Turán.

Theorem 2 (Turán)**.** *If a graph on n vertices has no $k+1$ sized cliques, then it has at most $\binom{k}{2}(n/k)^2$ edges.*

Again, we see that the theorem points to a particular type of extremal graph: Partition the vertices into k equal sized sets. Consider the graph that has all edges except those that stay in the same part of the partition. This graph has no $k+1$ cliques: for any set of $k+1$ vertices, two of the vertices will be in the same part by the pigeonhole principle, so there are no cliques of size $k+1$. On the other hand, the number of edges is exactly $\binom{k}{2}(n/k)^2$.

Next we prove the theorem. The proof shall closely follow Mantel's proof for the case $k = 2$.

Proof We shall prove the statement by induction on n. For $n \leq k$, the graph cannot have a $k+1$ sized clique, so the statement is vacuously true.

Suppose we are given a graph on n vertices with no $k+1$-cliques. If the graph does not have a k-clique, then adding any edge cannot give it a $k+1$-clique, so keep adding edges until a k-clique is created. Let A denote a k-clique. Let H denote all the remaining vertices.

The number of edges in A is exactly

$$\binom{k}{2}. \tag{1}$$

By induction, the number of edges in H is at most

$$\binom{k}{2} \cdot \left(\frac{n-k}{k}\right)^2. \tag{2}$$

We claim that the number of edges going from H to A is at most

$$(k-1)(n-k) = \binom{k}{2} \cdot 2\left(\frac{n-k}{k}\right), \tag{3}$$

otherwise, some vertex of H will be connected to all the vertices of A, forming a $k+1$ sized clique.

Summing the bounds (1), (2), (3), we get that the number of edges in the graph is at most

$$\binom{k}{2}\left(1 + 2\left(\frac{n-k}{k}\right) + \left(\frac{n-k}{k}\right)^2\right) = \binom{k}{2}\left(1 + \frac{n-k}{k}\right)^2 = \binom{k}{2}(n/k)^2.$$

■

2 The Sunflower Lemma

A *sunflower* is a collection of sets $S_1, \ldots, S_p$ that have exactly the same pairwise intersection. p will be called the number of petals.

Lemma 3. *Let $\mathcal{F}$ be a family of sets of cardinality ℓ. If $|\mathcal{F}| > \ell!(p-1)^\ell$, then $\mathcal{F}$ contains a sunflower with p petals.*

Proof We use induction on ℓ. For $\ell = 1$, there are more than $p-1$ disjoint sets that form a sunflower.

For larger ℓ, let $\mathcal{D}$ be a maximal collection of pairwise disjoint sets from $\mathcal{F}$. If $|\mathcal{D}| \geq p$, we are done, since $\mathcal{D}$ is a sunflower. Otherwise, let A be the union of the sets of $\mathcal{D}$. Then $|A| \leq (p-1)\ell$, and every set in $\mathcal{F}$ must intersect some set of A by the maximality of A. Thus, there must be some element of $x \in A$ that is in $\mathcal{F}/(p-1)\ell = (\ell-1)!(p-1)^{\ell-1}$ sets. Construct a family $\mathcal{F}'$ by taking these sets out, and removing x from them. By induction, $\mathcal{F}'$ has a sunflower with p petals, from which we obtain a sunflower in $\mathcal{F}$ by adding back the element x. ■

It is not known whether this is the right bound or not. Let $A_1, \ldots, A_\ell$ be disjoint sets of size $p-1$. Consider the family of sets $\{S : \forall i, |S \cap A_i| = 1\}$. This family has $(p-1)^\ell$ sets, yet no sunflower with p petals.

So here is a conjecture that is open:

Conjecture 4. *For every p, there is a constant C such that any family with C^ℓ sets of size ℓ must have a sunflower with p petals.*

3 Circuits

A boolean circuit is a directed acyclic graph where every vertex has fan-in 0 or 2. The vertices are sometimes called *gates*, and the edges are sometimes called *wires*. Each gate with fan-in 0 is labeled either by an input variable (x_i). Every other gate is labeled by a function mapping 2 bits to 1 bit. For any fixed input, the value of a gate is the value of the corresponding input variable, or the value of the corresponding function when evaluated on the gates that fan-in to the gate.

The *size* of such a circuit is the number of edges in the underlying graph, and we shall write $\mathsf{size}(f)$ to denote the size of the smallest circuit computing a function f. This model of computation is very robust under small changes.

First, every function can be computed by a circuit of exponential size:

Claim 5. *For every $f : \{0,1\}^n \to \{0,1\}$, $\mathsf{size}(f) \leq O(2^n/n)$.*

If we allow gates with fan-in C for any constant C, this does not change the size of any function by more than a constant factor, since it can be simulated by a circuit with fan-in 2 of constant size.

Every algorithm that runs in time $t(n)$ can be simulated on n-bit inputs by a circuit of size $t(n)^2$. Thus, it we want to prove that $P = NP$, it is enough to prove that some NP-complete problem cannot be computed by polynomial sized circuits.

Indeed, it is easy to prove that there are many functions that require large circuits:

Lemma 6. *Circuits of size s can compute at most $2^{O(s \log s)}$ functions.*

Proof Without loss of generality, assume that there are n input gates which take on each of the variables, and that there is a single designated output gate. Then each other gate must be labeled with one of $2^4 = 16$ functions. Each gate can take in two inputs, for which there are at most s^2 choices. Thus, the total number of configurations is at most $(16s^2)^s = 2^{O(s \log s)}$. ■

On the other hand, the number of functions on n bits is 2^{2^n}. Thus, if $s \ll 2^n/n$, almost all functions cannot be computed by circuits of size s. Still, it is open to give an example of any explicit family of functions $\{f_1, f_2, \ldots, \}$, where f_n is a boolean function on n bit inputs, such that $\lim_{n\to\infty} \frac{\mathsf{size}(f_n)}{n} = \infty$.

Another interesting quantity is the *depth* of a circuit. The depth is the length of the longest path in the circuit. Again, it is easy to check that every function can be computed in depth n, most functions require depth n, yet we know of no examples that require depth more than $O(\log n)$ (which is necessary just to touch all the inputs). It would be very interesting to find an example of such a function that cannot be computed by a logarithmic depth, linear sized circuit.

In light of this, research has focussed on special cases.

3.1 Linear Circuits

Let us restrict our attention to linear functions over $GF(2)$. Given any parity of k bits (i.e. a sum of k bits over $GF(2)$), one can compute it using a circuit of size $2k - 1$, just by taking a tree of parity gates. This is optimal, since any circuit that computes the parity must touch every input gate that is relevant, and it is easy to check that any graph with fewer edges will not be able to do it.

The situation becomes more interesting when we consider computing the matrix product Mx, where x is the n bit input written as an $n \times 1$ matrix, and M is a $n \times n$ boolean matrix. Then again,

we see that there are 2^{n^2} such linear functions, so all of them cannot be computed by circuits of size $n^2/\log n$ (here we require that every coordinate of Mx appear as some gate in the circuit). We can restrict the model further by restricting every internal gate of the circuit to only computing the sum (over $GF(2)$) of its inputs. Even with all of these restriction, we do not know of a way to show that a function cannot be computed by a logarithmic depth, linear sized circuit.

So let us restrict the model further. We allow each gate to compute the parity of an unbounded number of other gates, but restrict the depth of the circuit to be constant. In some sense, this is like forcing most of the circuit to look like a tree.

Given any two strings $a, b \in \{0,1\}^{\log n}$, write $a \cdot b = \sum_{i=1}^{\log n} a_i b_i$ to denote their inner product (over $GF(2)$). We define the matrix:

$$H_{a,b} = a \cdot b,$$

where the entries are indexed by $\log n$ bit strings.

Since $H = AB$, where A is the $n \times \log n$ matrix where every row is a distinct $\log n$ bit string, and B is the $\log n \times n$ matrix where every column is a distinct $\log n$ bit string, we see that H can be computed by a depth 2 circuit with unbounded fan-in parity gates, using $n \log n$ edges. We shall prove the following theorem, due to Alon, Karchmer and Wigderson:

Theorem 7. *Every depth 2 circuit computing H must have $\Omega(n \log n/\log\log n)$ wires.*

Proof For a constant c, we shall set $m = c \log n/\log\log n$. Now suppose the number of wires going from the inputs to the middle layer is at most $mn/2$. Then, there must be at least $n/2$ input gates that have at most m wires coming out of them. Consider the family of sets where the universe is a gate in the middle layer, and the set S_i is the set of gates that have a wire from x_i into them. By the sunflower lemma, as long as $n/2 \geq m^{2m} \geq m!(m-1)^m$, we can find a sunflower in this family.

Suppose S is the core of this sunflower, and $Z_1, \ldots, Z_m$ are the petals, where by relabeling the variables, we assume Z_i corresponds to the variable x_i. Then consider any output bit whose linear form includes exactly one of x_i, x_j. Such an output bit *must* use a gate from either Z_i or Z_j. There are $n/2$ such output bits, since for any two variables that correspond to vectors b_1, b_2, this happens exactly when $a \cdot (b_1 + b_2) = 1$. Thus we find $n/2$ wires into Z_i, Z_j. By pairing up all the petals and repeating the argument, we can find $mn/4$ wires into the petals. ■

3.2 Valiant's Rigidity Argument (not covered in class)

However, Valiant discovered a path to proving such a lower bound. First, he proved the following lemma about graphs:

Lemma 8. *Suppose every path in an undirected acyclic graph with e edges is of length at most 2^ℓ. Then for every k, there is a set of ek/ℓ edges that intersects every path of length $2^{\ell-k}$.*

Proof Label every vertex v of the graph by the length of the longest path starting at v, $p(v)$. Observe that if (u, v) is an edge, then $p(u) > p(v)$. Color an edge (u, v) with the color i if the i'th bit is the most significant bit where $p(u), p(v)$ disagree in their binary representation. Let S be the

set of edges that are colored with the k most infrequent colors. Then by averaging, we see that $|S| \leq ek/\ell$.

Now let $p'(u)$ be the number obtained from $p(u)$ by dropping the bit locations in the binary representation that correspond to the infrequent colors. Then if (u, v) is an edge colored by a frequent color, $p'(u) > p'(v)$, since the values of the infrequent bits are irrelevant to this inequality. Thus, we obtain a labeling of the vertices of the graph with $2^{\ell-k}$ values, such that the longest path that starts at u and avoids S is of length at most $p'(u)$. ■

Thus, (ignoring issues of rounding) given any circuit depth $c \cdot \log^n$ and size $c \cdot n$ with parity gates computing Mx, we can find a small subset of $cnk/\log\log n$ gates, such that all paths of length $c \log n/2^k$ must pass through this set of gates. In other words,

$$Mx = B(x, Ax),$$

where A is an $cnk/\log\log n \times n$ matrix that computes the values computed at the special gates, and B is a linear function that can be computed in depth $c \log n/2^k$. Since B is computable in such small depth, every row of B has at most $n^{c/2^k}$ non-zero values. In total, this shows that

$$M = S + L,$$

where $S = B(x, 0)$ is sparse in the sense that every row has at most $n^{c/2^k}$ entries, and $L = B(0, Ax)$ has rank at most $2^{cnk/\log\log n}$. Valiant called a matrix that cannot be expressed this way a *rigid* matrix. It is easy to show that most matrices are rigid. Thus, it would be enough to find a rigid matrix to find a matrix that cannot be computed in small depth and small size.