**Counting and Sampling** **Fall 2017**

# Lecture 16: Roots of the Matching Polynomial

*Lecturer: Shayan Oveis Gharan* *November 22nd*



Recall that for a graph $G$

$$\mu_G(x) = \sum_{k=0}^{n/2} (-1)^k m_k x^{n-2k}.$$

In this section we prove that the matching polynomial is real rooted and all of its roots are bounded from above by $2\sqrt{\Delta - 1}$ assuming that the maximum degree of $G$ is $\Delta$.

## 16.1 Real Rootedness of Matching Polynomial

The following fact can be proven similar to the facts on $\mu_G^+(x)$ that we proved in lecture 12.

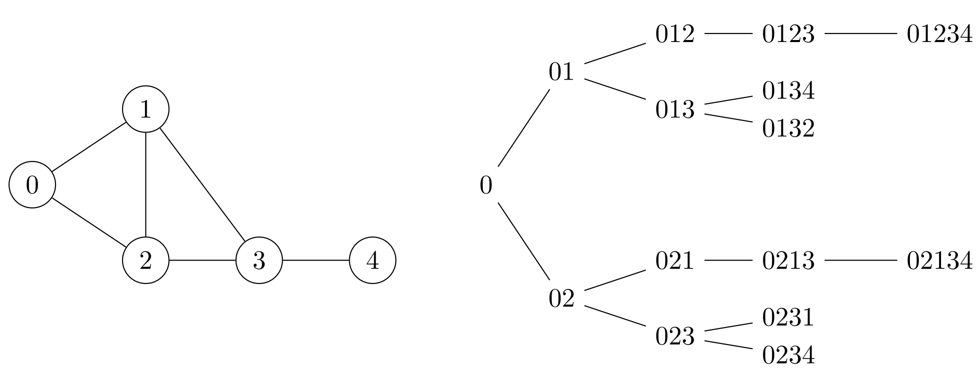
**Fact 16.1.** *For any pair of disjoint graphs $G, H$*

$$\mu_{G\cup H}(x) = \mu_G(x) \cdot \mu_H(x).$$

*For any graph $G$ and any vertex $u$,*

$$\mu_G(x) = x\mu_{G-u}(x) - \sum_{v\sim u} \mu_{G-u-v}(x).$$

For a graph $G$ and a vertex $u$, the *path-tree* of $G$ with respect to $u$, $T = T_G(u)$ is defined as follows: For every path in $G$ that starts at $u$, $T$ has a node and two paths are adjacent if their length differs by 1 and one is a prefix of another. See the following figure for an example:




We prove the following theorem due to Godsil and Gutman.

**Theorem 16.2.** *Let $G$ be a graph and $u$ be a vertex of $G$. Also, let $T = T(G, u)$ be the path tree of $G$ with respect to $u$. Then*

$$\frac{\mu_G(x)}{\mu_{G-u}(x)} = \frac{\mu_T(x)}{\mu_{T-u}(x)}.$$

Recall that this would directly imply that the largest root of $\mu_G(x)$ in absolute value is at most $2\sqrt{\Delta - 1}$. This is because if the maximum degree of $G$ is at most $\Delta$ so is the maximum degree of $T(G, u)$. The above theorem implies that the root of $\mu_G(x)$ are a subset of the roots of $\mu_{T(G,u)}(x)$. And, we proved in the last lecture that the largest root of $\mu_{T(G,u)}(x)$ is at most $2\sqrt{\Delta - 1}$.

*Proof.* Firstly, observe that the theorem obviously holds when $G$ is a tree because the path-tree of a tree is itself. So, suppose (inductively) that the theorem holds for all subgraph so of $G$. Let us write $H = G - u$. Then, using the first part of Fact 16.1 we have

$$\begin{aligned} \frac{\mu_G(x)}{\mu_H(x)} &= \frac{x\mu_{G-u}(x) - \sum_{v\sim u} \mu_{G-u-v}(x)}{\mu_H(x)} \\ &= x - \sum_{v\sim u} \frac{\mu_{H-v}(x)}{\mu_H(x)} = x - \sum_{v\sim u} \frac{\mu_{T(H,v)-v}(x)}{\mu_{T(H,v)}(x)}. \end{aligned} \tag{16.1}$$

The last equality simply follows by the induction hypothesis. Here is the main observation: $T(H, v) = T(G - u, v)$ is isomorphic to the component of $T(G, u) - u$ which contains the point $u, v$. Therefore,

$$\frac{\mu_{T(H,v)-v}(v)}{\mu_{T(H,v)}(x)} = \frac{\mu_{T(G,u)-u-v}(x)}{\mu_{T(G,u)-u}(x)}.$$

Note that the rest of the connected components of $T(G, u)-u$ are also connected components of $T(G, u)-u-v$ so they will be cancelled out in the RHS (see second part of Fact 16.1).

So, we can rewrite the RHS of (16.1) as follows:

$$\begin{aligned} x - \sum_{v\sim u} \frac{\mu_{T(H,v)-v}(x)}{\mu_{T(H,v)}(x)} &= x - \sum_{v\sim u} \frac{\mu_{T(G,u)-u-v}(x)}{\mu_{T(G,u)-u}(x)} \\ &= \frac{x\mu_{T(G,u)-u}(x) - \sum_{v\sim u} \mu_{T(G,u)-u-v}(x)}{\mu_{T(G,u)-u}(x)} = \frac{\mu_{T(G,u)}(x)}{\mu_{T(G,u)-u}(x)}. \end{aligned}$$

This proves the theorem. □

**Corollary 16.3.** *For any graph $G$ the polynomial $\mu_G(x)$ divides $\mu_T(x)$.*

*Proof.* Firstly, by the first part we can write

$$\mu_T(x) = \mu_G(x) \cdot \frac{\mu_{T-u}(x)}{\mu_{G-u}(x)}.$$

To prove the second part we need to show that the ratio $\mu_{T-u}(x)$ is divisible by $\mu_{G-u}(x)$. Firstly, note that $\mu_{T-u}(x) = \mu_{T(G,u)-u}(x)$ is divisible by $\mu_{T(G-u,v)}(x)$. This is because the latter is isomorphic to one of the connected components of the former. Secondly, by induction $\mu_{T(G-u,v)}(x)$ is divisible by $\mu_{G-u}(x)$. Putting these together, we get $\mu_{T-u}(x)$ is divisible by $\mu_{G-u}(x)$. □

## 16.2 Estimating the Coefficients of the Matching Polynomial

Next, we discuss an algorithm to estimate the coefficient of $x^k$ of $\mu_G(x)$ in time $C^{O(k)}$ for some constant $C > 0$. Note that the naiive algorithm takes time $n^{O(k)}$ to count the number of $k$-matchings of $G$. Furthermore, note that in the above application of estimating $\mu_G^+(x)$ for say $x = 1$ we need to know the first $O(\log(n/\epsilon))$ coefficients exactly so we cannot use the FPRAS of Jerrum-Sinclair-Vigoda [JSV04].

The algorithm follows from the above lemma.

**Lemma 16.4.** *For any spanning tree $T$ rooted at $u$, the polynomial*

$$x^{-1}\frac{\mu_{T-u}(x^{-1})}{\mu_T(x^{-1})} = \sum_k A^k_{u,u} x^k$$

*is the generating polynomial of walks where $A$ is the adjacency matrix of $T$, and $A^k_{u,u}$ is the number of closed walks of length $k$ started at $u$.*

*Proof.* First of all in the last lecture we saw that for any tree $T$, $\mu_T(x) = \det(xI - A)$. Therefore, it is enough to prove the claim for the ratio $x^{-1}\frac{\det(x^{-1}I - A_{T-u})}{\det(x^{-1}I-A)}$. We claim that

$$\sum_k A^k_{u,u} x^k = (I - xA)^{-1}_{u,u}.$$

This is just because $(I - xA)^{-1} = \sum_{k\geq 0} x^k A^k$. So, it remains to show that

$$x^{-1}\frac{\det(x^{-1}I - A_{T-u})}{\det(x^{-1}I - A)} = (I - xA)^{-1}_{u,u}.$$

Next, we use the following well-known facts: For any matrix $A$, $\text{adj}(A) = A^{-1}\det(A)$ where $\text{adj}(A)$ is the adjoint of $A$ is the matrix where (up to the sign) the $i, j$ the entry is the determinant of the submatrix of $A$ where the $i$-th row and $j$-th column are removed. Therefore,

$$(I - xA)^{-1} = x^{-1}(x^{-1}I - A)^{-1} = x^{-1}\frac{\text{adj}(x^{-1}I - A)}{\det(x^{-1}I - A)}.$$

So, the $u, u$-th entry of both sides are equal. But the $u, u$-th entry of $\text{adj}(x^{-1}I - A)$ is exactly $\det(x^{-1}I - A_{T-u})$. This completes the proof. □

Now observe that by the above theorem we can compute the coefficient of $x^{-k}$ of $\frac{\mu_{G-u}(x)}{\mu_G(x)}$ in time $O(\Delta)^k$. All we need to do is to construct the path-tree about $u$ of depth $k$ (This needs time $\Delta^k$. Then, we just count the number of closed walks of length $k$ started at $u$ in that tree.

It remains to compute $\mu_G(x)$. We use the following simple observation that we leave as an exercise: For any graph $G$,

$$\sum_u \mu_{G-u}(x) = \mu'_G(x).$$

Therefore, in time $O(n\Delta^k)$ we can compute the coefficients of $x^{-1}, \ldots, x^{-k}$ in $\frac{\mu'_G(x)}{\mu_G(x)}$. Now, we claim this is enough to find the coefficients of $x^1, \ldots, x^k$ of $\mu_G(x)$.

Firstly, say $r_1, \ldots, r_n$ are the roots of $\mu_G(x)$. Then,

$$\frac{\mu'_G(x)}{\mu_G(x)} = \sum_i \frac{1}{x - r_i} = x^{-1}\sum_i\sum_j x^{-j} r_i^j = \sum_j x^{-j}\sum_i r_i^j.$$

Therefore, the coefficient of $x^{-j}$ is the $j$-th power sum of the roots of $\mu_G(x)$. It follows by Newton identities that this is enough to estimate the top $k$ coefficients.

**Lemma 16.5** (Newton Identities)**.** *For any polynomial $\sum_{i=0}^n a_i x^i$ with roots $r_1, \ldots, r_n$ we can determine the coefficients $a_0, \ldots, a_k$ from the power sums of the roots $p_0, p_1, \ldots, p_k$ where for all $k$,*

$$p_k = \sum_i r_i^k.$$

*Proof.* First observe that $a_0, \ldots, a_n$ are elementary symmetric polynomials of the roots:

$$a_0 = 1, a_1 = \sum_i r_i, a_2 = \sum_{i<j} x_i x_j, \ldots$$

It turns out that $a_0, \ldots, a_k$ form a basis for all symmetric polynomials in $r_1, \ldots, r_n$ of degree $k$. Therefore, given $a_0, \ldots, a_k$ we can compute $p_0, \ldots, p_k$ and vice versa. Newton identities give this translation in the case of power sums:

$$\begin{aligned}
p_1 &= e_1 \\
p_2 &= e_1 p_1 - 2e_2, \\
p_3 &= e_1 p_2 - e_2 p_1 + 3e_3, \\
p_4 &= e_1 p_3 - e_2 p_2 + e_3 p_1 - 4e_4, \\
&\vdots
\end{aligned}$$

□

## 16.3 Estimating Low Order Coefficients of the Matching Polynomial

For a graph $G$ and (a small graph) $H$ let $\mathrm{ind}(H, G)$ be the number of subsets $S$ of $G$ such that $H$ is isomorphic to $G[S]$.

In this section we prove the following theorem due to Patel and Regts [PR17].

**Theorem 16.6.** *Let $G$ be a graph of maximum degree. Consider a polynomial $q(z) = x^n + \sum_{i=1}^n e_i x^{n-i}$ where $e_i = \sum_{H \in \mathcal{G}_i} \lambda_H \, \mathrm{ind}(H, G)$ is the coefficient of $x^{n-i}$. Here $\mathcal{G}_i$ is a family of graphs of size at most $i$. Then, we can compute $e_1, \ldots, e_k$ in time $\mathrm{poly}(n)\Delta^{O(k)}$.*

Recall that the above theorem implies that we can estimate the polynomial

**Fact 16.7.** *For any connected graph $H$ and any graph $G$ with maximum degree $\Delta$ we can exactly compute $\mathrm{ind}(H, G)$ in time $O(n\Delta^{|V(H)|})$.*

*Proof.* Let $k = |V(H)|$. Fix a spanning tree subgraph $T$ of $H$ and define an ordering $v_1, v_2, \ldots, v_k$ of vertices of $H$ where for all $i \geq 2$, $v_i$ is adjacent to one of the vertices $v_1, \ldots, v_{i-1}$. First we guess the mapping of $v_1$, say $u_1$ under the isomorphism. There are $n$ possibilities. Next, we find the mapping of $v_2$; since $v_2$ is adjacent to $u_1$ it has to be mapped to one of the neighbors of $u_1$. But, $u_1$ has at most $\Delta$ neighbors. So, there are only $\Delta$ options. Say $u_2$ is the map of $v_2$. Now, $v_3$ is adjacent to $v_1$ or $v_2$. Either way there are at most $\Delta$ options to guess the map of $v_3$, and so on. □

So, the main difficulty in computing $e_i$ is when we need to compute $\mathrm{ind}(H, G)$ for a disconnected graph $H$. For example, if the polynomial $q(.)$ corresponds to the matching polynomial or the independence polynomial $H$ corresponds to matchings or independent sets and it is disconnected.

First of all, using Newton identities instead of computing $e_1, \ldots, e_k$ it is enough to compute $p_1, \ldots, p_k$. Secondly, using the recursive definition of $p_i$'s we can write each $p_i$ as

$$p_i = \sum_{H \in \mathcal{G}_i} a_H \, \mathrm{ind}(H, G).$$

Here we do not discuss how to compute $a_H$ and we refer to [**?**] for details. But in the high-level $a_H$'s are just functions of $\lambda_H$'s and can be computed recursively.

The main observation of [PR17] is that $p_i$'s are *additive properties* in the following sense: Suppose we have a disjoint union of graphs $G_1, G_2$. Then $q_{G_1 \cup G_2}(z) = q_{G_1}(z) \cdot q_{G_2}(z)$. Say $p_i^G$ is the $i$-th power sum of the roots of $q_G(z)$. Then, observe that for all $i$,

$$p_i^{G_1 \cup G_2} = p_i^{G_1} + p_i G_2.$$

On the other hand, observe that if $H$ is connected, then

$$\text{ind}(H, G_1 \cup G_2) = \text{ind}(H, G_1) + \text{ind}(H, G_2),$$

but if it is disconnected say $H = H_1 \cup H_2$, then

$$\text{ind}(H, G_1, \cup G_2) = \text{ind}(H_1, G_1)\,\text{ind}(H_2, G_2) + \text{ind}(H_1, G_2)\,\text{ind}(H_2, G_1).$$

It follows that for any power sum $p_i$ all coefficients $a_H$ corresponding to disconnected graphs $H$ must be zero. For a concrete example recall that in the previous section we showed that in the case of matching polynomial for each $i$, $p_i$ corresponds to closed tree-like-walks which are connected subgraphs. Therefore, by the above fact we can compute each $p_i$ exactly in time polynomial in $n$ and $\Delta^{O(i)}$.

## References

[JSV04] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM*, 51(4):671–697, July 2004. 16-2

[PR17] Viresh Patel and Guus Regts. Deterministic polynomial-time approximation algorithms for partition functions and graph polynomials. *Electronic Notes in Discrete Mathematics*, 61(Supplement C):971 – 977, 2017. The European Conference on Combinatorics, Graph Theory and Applications (EUROCOMB). 16-4, 16-5