Partition Functions: Zeros and efficient approximation III

Viresh Patel and Guus Regts

Summerschool on Algorithms, Dynamics, and Information Flow in Networks, Dortmund

June 27–July 1, 2022

Introduction

In the previous two lectures:
Absence of zeros for partition functions $\Rightarrow$ efficient approximation algorithms.

Introduction

In the previous two lectures:
Absence of zeros for partition functions $\Rightarrow$ efficient approximation algorithms.

This and the next lecture:
What about presence of zeros?

The matching polynomial

The *matching polynomial* of a graph $G = (V, E)$ is defined as

$$M_G(z) = \sum_{k \geq 0} m_k z^k$$

m_k denotes the number of matchings with k edges.

The matching polynomial

The *matching polynomial* of a graph $G = (V, E)$ is defined as

$$M_G(z) = \sum_{k \geq 0} m_k z^k$$

m_k denotes the number of matchings with k edges.

Theorem (Heilmann and Lieb, 1972)

Let $\Delta \geq 3$. Then for any graph $G \in \mathcal{G}_\Delta$, and any $z \notin (-\infty, -\frac{1}{4(\Delta-1)})$, $M_G(z) \neq 0$ and this is tight.

The matching polynomial

The *matching polynomial* of a graph $G = (V, E)$ is defined as

$$M_G(z) = \sum_{k \geq 0} m_k z^k$$

m_k denotes the number of matchings with k edges.

Theorem (Heilmann and Lieb, 1972)

Let $\Delta \geq 3$. Then for any graph $G \in \mathcal{G}_\Delta$, and any $z \notin (-\infty, -\frac{1}{4(\Delta-1)})$, $M_G(z) \neq 0$ and this is tight.

Theorem (Bezáková, Galanis, Goldberg Štefankovič, 2021)

Let $\Delta \geq 3$ and let $z < -\frac{1}{4(\Delta-1)}$ be rational. Approximating the absolute value of $M_G(z)$ for $G \in \mathcal{G}_\Delta$ is #P-hard.

The partition function of the ferromagnetic Ising model

The *partition function of the ferromagnetic Ising model* of a graph $G = (V, E)$ is defined as

$$Z_G(\lambda, b) = \sum_{S \subseteq V} \lambda^{|S|} b^{e(S, V \setminus S)}$$

$e(S, V \setminus S)$ denotes the number of edges across the cut defined by S.

The partition function of the ferromagnetic Ising model

The *partition function of the ferromagnetic Ising model* of a graph $G = (V, E)$ is defined as

$$Z_G(\lambda, b) = \sum_{S \subseteq V} \lambda^{|S|} b^{e(S, V \setminus S)}$$

$e(S, V \setminus S)$ denotes the number of edges across the cut defined by S.

Theorem (Peters, R., 2020)

Let $b \in (0, 1)$. Then there exists $\Delta_b > 0$ such that the roots $Z_G(\lambda, b)$ as a polynomial in λ as G ranges over $\mathcal{G}_{\Delta_b}$ are dense in the unit circle.

The partition function of the ferromagnetic Ising model

The *partition function of the ferromagnetic Ising model* of a graph $G = (V, E)$ is defined as

$$Z_G(\lambda, b) = \sum_{S \subseteq V} \lambda^{|S|} b^{e(S, V \setminus S)}$$

$e(S, V \setminus S)$ denotes the number of edges across the cut defined by S.

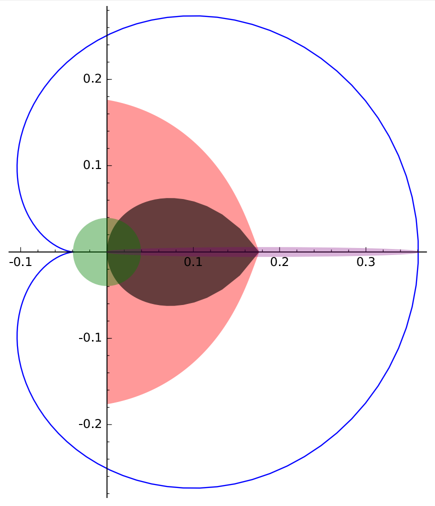
Theorem (Peters, R., 2020)

Let $b \in (0, 1)$. Then there exists $\Delta_b > 0$ such that the roots $Z_G(\lambda, b)$ as a polynomial in λ as G ranges over $\mathcal{G}_{\Delta_b}$ are dense in the unit circle.

Theorem (Buys, Galanis, Patel, R., 2022)

Let $b \in (0, 1) \cap \mathbb{Q}$. Let $\lambda \in \mathbb{Q}[i] \setminus \mathbb{R}$ such that $|\lambda| = 1$. Approximating the absolute value of $Z_G(\lambda, b)$ for $G \in \mathcal{G}_{\Delta_b}$ is #P-hard.

The independence polynomial on $\mathbb{C}$



The independence polynomial on $\mathbb{C}$

$$\mathcal{Z}_\Delta := \{\lambda \in \mathbb{C} \mid Z(G;\lambda) = 0 \text{ for some } G \in \mathcal{G}_\Delta\}$$
$$\mathcal{P}_\Delta := \{\lambda \in \mathbb{Q}[i] \mid \text{ approximating } |Z(G;\lambda)| \text{ is \#P-hard on } \mathcal{G}_\Delta\}$$

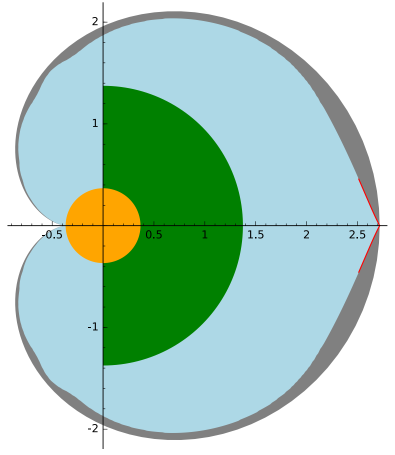
The independence polynomial on $\mathbb{C}$

$$\mathcal{Z}_\Delta := \{\lambda \in \mathbb{C} \mid Z(G;\lambda) = 0 \text{ for some } G \in \mathcal{G}_\Delta\}$$
$$\mathcal{P}_\Delta := \{\lambda \in \mathbb{Q}[i] \mid \text{ approximating } |Z(G;\lambda)| \text{ is \#P-hard on } \mathcal{G}_\Delta\}$$

Theorem (de Boer, Buys, Guerini, Peters, R. 2021+)

Let $\Delta \geq 3$. The closure of $\mathcal{Z}_\Delta$ is contained in the closure of $\mathcal{P}_\Delta$.

State of the art for the independence polynomial as $\Delta \to \infty$



Overview of the rest of the lectures

Ingredients for 'zeros implies hardness' for the independence polynomial on $\mathbb{C}$.

- Why is approximating as hard as exact computing?
- What do the complex zeros have to do with this?

Precise formulation of the problem

Let $\lambda \in \mathbb{Q}[i]$ and $\Delta \geq 4$.

Name #Hardcorenorm(λ, Δ).

Instance A graph $G \in \mathcal{G}_\Delta$.

Output If $Z_G(\lambda) = 0$ the algorithm may output any rational number; otherwise the algorithm must output a rational number N such that

$$\frac{1}{1.001} \leq |Z_G(\lambda)|/N \leq 1.001.$$

Precise formulation of the problem

Let $\lambda \in \mathbb{Q}[i]$ and $\Delta \geq 4$.

Name $\#\text{Hardcorenorm}(\lambda, \Delta)$.

Instance A graph $G \in \mathcal{G}_\Delta$.

Output If $Z_G(\lambda) = 0$ the algorithm may output any rational number; otherwise the algorithm must output a rational number N such that

$$\frac{1}{1.001} \leq |Z_G(\lambda)|/N \leq 1.001.$$

We will show that for certain λ's, having acces to an algorithm that solves $\#\text{Hardcorenorm}(\lambda, \Delta)$ in polynomial time, we can compute $Z_G(\lambda)$ exactly in time polynomial in $|V(G)|$ for graphs $G \in \mathcal{G}_3$.

Reducing approximate counting to exact counting: ratios I

Let $\mu \in \mathbb{Q}[i]$ and $H \in \mathcal{G}_3$. We want to compute $Z_H(\mu)$ exactly.

Reducing approximate counting to exact counting: ratios I

Let $\mu \in \mathbb{Q}[i]$ and $H \in \mathcal{G}_3$. We want to compute $Z_H(\mu)$ exactly.

Lemma

Suppose we have acces to an algorithm that on input of a graph $G \in \mathcal{G}_3$ outputs numbers $r \in \mathbb{Q}[i]$ and $b \in \{0,1\}$ in time polynomial in $|V(G)|$ such that

Reducing approximate counting to exact counting: ratios I

Let $\mu \in \mathbb{Q}[i]$ and $H \in \mathcal{G}_3$. We want to compute $Z_H(\mu)$ exactly.

Lemma

Suppose we have acces to an algorithm that on input of a graph $G \in \mathcal{G}_3$ outputs numbers $r \in \mathbb{Q}[i]$ and $b \in \{0,1\}$ in time polynomial in $|V(G)|$ such that

- *if $Z_{G \setminus N[v]}(\mu) \neq 0$, then $b = 1$ and $r = \frac{Z_G(\mu)}{Z_{G\setminus N[v]}(\mu)}$*

Reducing approximate counting to exact counting: ratios I

Let $\mu \in \mathbb{Q}[i]$ and $H \in \mathcal{G}_3$. We want to compute $Z_H(\mu)$ exactly.

Lemma

Suppose we have acces to an algorithm that on input of a graph $G \in \mathcal{G}_3$ outputs numbers $r \in \mathbb{Q}[i]$ and $b \in \{0,1\}$ in time polynomial in $|V(G)|$ such that

- *if $Z_{G\setminus N[v]}(\mu) \neq 0$, then $b = 1$ and $r = \frac{Z_G(\mu)}{Z_{G\setminus N[v]}(\mu)}$*
- *if $Z_{G\setminus N[v]}(\mu) = 0$ and $Z_{G-v}(\mu) \neq 0$, then $b = 0$ and $r = \frac{Z_G(\mu)}{Z_{G-v}(\mu)}$.*

Then $Z_G(\mu)$ can be computed in time polynomial in $|V(G)|$.

Reducing approximate counting to exact counting: ratios I

Reducing approximate counting to exact counting: ratios I

Reducing approximate counting to exact counting: ratios II

Let (G, v) be a rooted graph. The ratio $R_{G,v}$ is the rational function

$$\lambda \mapsto \frac{Z_G^{v\ \mathrm{in}}(\lambda)}{Z_G^{v\ \mathrm{out}}(\lambda)} = \frac{\lambda Z_{G \setminus N[v]}(\lambda)}{Z_{G-v}(\lambda)}.$$

Reducing approximate counting to exact counting: ratios II

Let (G, v) be a rooted graph. The ratio $R_{G,v}$ is the rational function

$$\lambda \mapsto \frac{Z_G^{v \text{ in}}(\lambda)}{Z_G^{v \text{ out}}(\lambda)} = \frac{\lambda Z_{G \setminus N[v]}(\lambda)}{Z_{G-v}(\lambda)}.$$

We have

$$\frac{Z_G(\lambda)}{Z_{G \setminus N[v]}(\lambda)}$$

Reducing approximate counting to exact counting: ratios II

Let (G, v) be a rooted graph. The ratio $R_{G,v}$ is the rational function

$$\lambda \mapsto \frac{Z_G^{v \text{ in}}(\lambda)}{Z_G^{v \text{ out}}(\lambda)} = \frac{\lambda Z_{G \setminus N[v]}(\lambda)}{Z_{G-v}(\lambda)}.$$

We have

$$\frac{Z_G(\lambda)}{Z_{G \setminus N[v]}(\lambda)} = \frac{\lambda Z_{G \setminus N[v]}(\lambda) + Z_{G-v}(\lambda)}{Z_{G \setminus N[v]}(\lambda)}$$

Reducing approximate counting to exact counting: ratios II

Let (G, v) be a rooted graph. The ratio $R_{G,v}$ is the rational function

$$\lambda \mapsto \frac{Z_G^{v \text{ in}}(\lambda)}{Z_G^{v \text{ out}}(\lambda)} = \frac{\lambda Z_{G \setminus N[v]}(\lambda)}{Z_{G-v}(\lambda)}.$$

We have

$$\frac{Z_G(\lambda)}{Z_{G \setminus N[v]}(\lambda)} = \frac{\lambda Z_{G \setminus N[v]}(\lambda) + Z_{G-v}(\lambda)}{Z_{G \setminus N[v]}(\lambda)} = \lambda \left(1 + \frac{1}{R_{G,v}}\right)$$

Reducing approximate counting to exact counting: ratios II

Let (G, v) be a rooted graph. The ratio $R_{G,v}$ is the rational function

$$\lambda \mapsto \frac{Z_G^{v \text{ in}}(\lambda)}{Z_G^{v \text{ out}}(\lambda)} = \frac{\lambda Z_{G \setminus N[v]}(\lambda)}{Z_{G-v}(\lambda)}.$$

We have

$$\frac{Z_G(\lambda)}{Z_{G \setminus N[v]}(\lambda)} = \frac{\lambda Z_{G \setminus N[v]}(\lambda) + Z_{G-v}(\lambda)}{Z_{G \setminus N[v]}(\lambda)} = \lambda \left(1 + \frac{1}{R_{G,v}}\right)$$

$$\frac{Z_G(\lambda)}{Z_{G-v}(\lambda)} = \frac{\lambda Z_{G \setminus N[v]}(\lambda) + Z_{G-v}(\lambda)}{Z_{G-v}(\lambda)} = R_{G,v} + 1$$

Reducing approximate counting to exact counting: ratios II

Let (G, v) be a rooted graph. The ratio $R_{G,v}$ is the rational function

$$\lambda \mapsto \frac{Z_G^{v \text{ in}}(\lambda)}{Z_G^{v \text{ out}}(\lambda)} = \frac{\lambda Z_{G \setminus N[v]}(\lambda)}{Z_{G-v}(\lambda)}.$$

We have

$$\frac{Z_G(\lambda)}{Z_{G \setminus N[v]}(\lambda)} = \frac{\lambda Z_{G \setminus N[v]}(\lambda) + Z_{G-v}(\lambda)}{Z_{G \setminus N[v]}(\lambda)} = \lambda \left(1 + \frac{1}{R_{G,v}}\right)$$

$$\frac{Z_G(\lambda)}{Z_{G-v}(\lambda)} = \frac{\lambda Z_{G \setminus N[v]}(\lambda) + Z_{G-v}(\lambda)}{Z_{G-v}(\lambda)} = R_{G,v} + 1$$

It thus suffices to be able to compute $R_{G,v}$ in the previous lemma.

An assumption on μ

(Assumption)

Let $\Delta \geq 4$. Assume that μ is such that on input of any $y \in \mathbb{Q}[i]$ and $\varepsilon \in (0,1)$ we can compute in time $\text{poly}(\log(1/\varepsilon) + \text{size}(y))$ a rooted tree $(T, u) \in \mathcal{G}_\Delta$ such that $\deg_T(u) = 1$ and

An assumption on μ

(Assumption)

Let $\Delta \geq 4$. Assume that μ is such that on input of any $y \in \mathbb{Q}[i]$ and $\varepsilon \in (0, 1)$ we can compute in time $\text{poly}(\log(1/\varepsilon) + \text{size}(y))$ a rooted tree $(T, u) \in \mathcal{G}_\Delta$ such that $\deg_T(u) = 1$ and

- $|R_{T,u}(\mu) - y| \leq \varepsilon$,

An assumption on μ

(Assumption)

Let $\Delta \geq 4$. Assume that μ is such that on input of any $y \in \mathbb{Q}[i]$ and $\varepsilon \in (0,1)$ we can compute in time $\text{poly}(\log(1/\varepsilon) + \text{size}(y))$ a rooted tree $(T, u) \in \mathcal{G}_\Delta$ such that $\deg_T(u) = 1$ and

- $|R_{T,u}(\mu) - y| \leq \varepsilon$,
- $|T| = \text{poly}(\log(1/\varepsilon) + \text{size}(y))$,

An assumption on μ

(Assumption)

Let $\Delta \geq 4$. Assume that μ is such that on input of any $y \in \mathbb{Q}[i]$ and $\varepsilon \in (0,1)$ we can compute in time $\text{poly}(\log(1/\varepsilon) + \text{size}(y))$ a rooted tree $(T, u) \in \mathcal{G}_\Delta$ such that $\deg_T(u) = 1$ and

- $|R_{T,u}(\mu) - y| \leq \varepsilon$,
- $|T| = \text{poly}(\log(1/\varepsilon) + \text{size}(y))$,
- $Z_{T-u}(\mu) \neq 0$.

An assumption on μ

(Assumption)

Let $\Delta \geq 4$. Assume that μ is such that on input of any $y \in \mathbb{Q}[i]$ and $\varepsilon \in (0,1)$ we can compute in time poly$(\log(1/\varepsilon) + \text{size}(y))$ a rooted tree $(T,u) \in \mathcal{G}_\Delta$ such that $\deg_T(u) = 1$ and

- $|R_{T,u}(\mu) - y| \leq \varepsilon$,
- $|T| = \text{poly}(\log(1/\varepsilon) + \text{size}(y))$,
- $Z_{T-u}(\mu) \neq 0$.

Given $(G,v) \in \mathcal{G}_3$ and (T,u) as above make a new graph $G(T) \in \mathcal{G}_\Delta$ by identifying u with v.

(Assumption)

Let $\Delta \geq 4$. Assume that μ is such that on input of any $y \in \mathbb{Q}[i]$ and $\varepsilon \in (0,1)$ we can compute in time $\text{poly}(\log(1/\varepsilon) + \text{size}(y))$ a rooted tree $(T,u) \in \mathcal{G}_\Delta$ such that $\deg_T(u) = 1$ and

- $|R_{T,u}(\mu) - y| \leq \varepsilon$,
- $|T| = \text{poly}(\log(1/\varepsilon) + \text{size}(y))$,
- $Z_{T-u}(\mu) \neq 0$.

Given $(G,v) \in \mathcal{G}_3$ and (T,u) as above make a new graph $G(T) \in \mathcal{G}_\Delta$ by identifying u with v. Then

$$Z_{G(T)}(\mu) \cong Z_{T-u}(\mu)\left(Z_{G-v}(\mu) + yZ_{G\setminus N[v]}(\mu)\right).$$

Reducing exact counting approximate counting: binary search I

$$Z_{G(T)}(\mu) \cong Z_{T-u}(\mu)\left(Z_{G-v}(\mu) + y Z_{G\setminus N[v]}(\mu)\right).$$

Reducing exact counting approximate counting: binary search I

$$Z_{G(T)}(\mu) \cong Z_{T-u}(\mu)\left(Z_{G-v}(\mu) + y Z_{G\setminus N[v]}(\mu)\right).$$

Write $A = Z_{G\setminus N[v]}(\mu)$ and $B = Z_{G-v}(\mu)$. Define $f(y) = Ay + B$.

Reducing exact counting approximate counting: binary search I

$$Z_{G(T)}(\mu) \cong Z_{T-u}(\mu)\left(Z_{G-v}(\mu) + y Z_{G\setminus N[v]}(\mu)\right).$$

Write $A = Z_{G\setminus N[v]}(\mu)$ and $B = Z_{G-v}(\mu)$. Define $f(y) = Ay + B$. Note that provided $A \neq 0$,

$$f(y) = 0 \text{ iff } y = -B/A.$$

Reducing exact counting approximate counting: binary search I

$$Z_{G(T)}(\mu) \cong Z_{T-u}(\mu)\left(Z_{G-v}(\mu) + y Z_{G\setminus N[v]}(\mu)\right).$$

Write $A = Z_{G\setminus N[v]}(\mu)$ and $B = Z_{G-v}(\mu)$. Define $f(y) = Ay + B$. Note that provided $A \neq 0$,

$$f(y) = 0 \text{ iff } y = -B/A.$$

Using an algorithm for #Hardcorenorm(μ, Δ) we can compute an η-approximation $\hat{f}(y)$ to $|f(y)|$ (recall $\eta = 1.001$)

Intermezzo on rational numbers

Let $\lambda = x + iy \in \mathbb{Q}[i]$ be fixed. Let $G = (V, E)$ be an n-vertex graph. Then

- $|Z_G(\lambda)| \leq 2^{O(n)}$,
- the bit size of the number $Z_G(\lambda)$ is $O(n)$,
- if $Z_G(\lambda) \neq 0$, then $|Z_G(\lambda)| > 2^{-O(n)}$.

Reducing exact counting approximate counting: binary search II

Assume $A \neq 0$. Denote $y^* = -B/A$. Note that since $\mu \in \mathbb{Q}[i]$ we must have that y^* is contained in some square box S of diameter $D = 2^{O(|V(G)|)}$ with center m.

Reducing exact counting approximate counting: binary search II

Assume $A \neq 0$. Denote $y^* = -B/A$. Note that since $\mu \in \mathbb{Q}[i]$ we must have that y^* is contained in some square box S of diameter $D = 2^{O(|V(G)|)}$ with center m.

Let $y_1 = m - 4\eta D \quad y_2 = m + 4\eta D$. Then

$$\hat{f}(y_1) - \hat{f}(y_2) \geq |A| \left(|y_1 - y^*| - |y_2 - y^*| - 2\eta D\right)$$

Reducing exact counting approximate counting: binary search II

Assume $A \neq 0$. Denote $y^* = -B/A$. Note that since $\mu \in \mathbb{Q}[i]$ we must have that y^* is contained in some square box S of diameter $D = 2^{O(|V(G)|)}$ with center m.

Let $y_1 = m - 4\eta D \quad y_2 = m + 4\eta D$. Then

$$\hat{f}(y_1) - \hat{f}(y_2) \geq |A| \left(|y_1 - y^*| - |y_2 - y^*| - 2\eta D \right)$$

Suppose now that $\hat{f}(y_1) \leq \hat{f}(y_2)$. Then

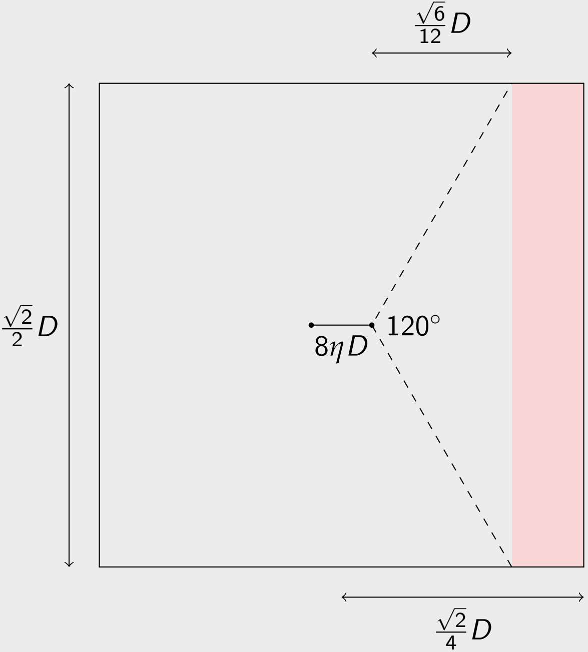
Reducing exact counting approximate counting: binary search II

Assume $A \neq 0$. Denote $y^* = -B/A$. Note that since $\mu \in \mathbb{Q}[i]$ we must have that y^* is contained in some square box S of diameter $D = 2^{O(|V(G)|)}$ with center m.

Let $y_1 = m - 4\eta D \quad y_2 = m + 4\eta D$. Then

$$\hat{f}(y_1) - \hat{f}(y_2) \geq |A| \left(|y_1 - y^*| - |y_2 - y^*| - 2\eta D\right)$$

Suppose now that $\hat{f}(y_1) \leq \hat{f}(y_2)$. Then y^* cannot be contained in the cone of 120° centered at y_2 pointing towards y_1.


$\frac{\sqrt{6}}{12}D$
$\frac{\sqrt{2}}{2}D$
$8\eta D$
120°
$\frac{\sqrt{2}}{4}D$

Box shrinking

After $O(|V(G)|)$ many steps the initial box S shrinks down to a box S' of diameter

$$D' = (7/8)^{O(|V(G)|)} D = 2^{-O(|V(G)|)}.$$

Box shrinking

After $O(|V(G)|)$ many steps the initial box S shrinks down to a box S' of diameter

$$D' = (7/8)^{O(|V(G)|)} D = 2^{-O(|V(G)|)}.$$

- If $A \neq 0$, then for any $y \in S' \setminus \{y^*\}$, $\hat{f}(y) \leq (1+\eta)|A|D'$.

Box shrinking

After $O(|V(G)|)$ many steps the initial box S shrinks down to a box S' of diameter

$$D' = (7/8)^{O(|V(G)|)} D = 2^{-O(|V(G)|)}.$$

- If $A \neq 0$, then for any $y \in S' \setminus \{y^*\}$, $\hat{f}(y) \leq (1+\eta)|A|D'$.
- If $A = 0$ and $B \neq 0$, then for any $y \in S'$, $\hat{f}(y) \geq (1+\eta)|B|$.

Box shrinking

After $O(|V(G)|)$ many steps the initial box S shrinks down to a box S' of diameter

$$D' = (7/8)^{O(|V(G)|)} D = 2^{-O(|V(G)|)}.$$

- If $A \neq 0$, then for any $y \in S' \setminus \{y^*\}$, $\hat{f}(y) \leq (1+\eta)|A|D'$.
- If $A = 0$ and $B \neq 0$, then for any $y \in S'$, $\hat{f}(y) \geq (1+\eta)|B|$.

Since $|B| = \Omega(2^{-|V(G)|})$ and $|A| = O(2^{|V(G)|})$, we can thus determine if $A \neq 0$ or not by computing $\hat{f}(y)$ for two values $y \in S'$.

Box shrinking

After $O(|V(G)|)$ many steps the initial box S shrinks down to a box S' of diameter

$$D' = (7/8)^{O(|V(G)|)} D = 2^{-O(|V(G)|)}.$$

- If $A \neq 0$, then for any $y \in S' \setminus \{y^*\}$, $\hat{f}(y) \leq (1+\eta)|A|D'$.
- If $A = 0$ and $B \neq 0$, then for any $y \in S'$, $\hat{f}(y) \geq (1+\eta)|B|$.

Since $|B| = \Omega(2^{-|V(G)|})$ and $|A| = O(2^{|V(G)|})$, we can thus determine if $A \neq 0$ or not by computing $\hat{f}(y)$ for two values $y \in S'$.

- If $A \neq 0$ (and thus y^* is contained in the initial box), we can determine y^* exactly (being the unique complex number with rational coordinates whose denominators are bounded by $2^{O(|V(G)|)}$.)

Conclusion

Since $\varepsilon = 2^{-O(|V(G)|)}$, having an poly-time algorithm for $\#$Hardcorenorm(μ, Δ), the box shrinking procedure gives us a polynomial time algorithm to detect whether $A = 0$ or not and compute $1/y^* = -A/B$ provided not both A and B are equal to 0.

Conclusion

Since $\varepsilon = 2^{-O(|V(G)|)}$, having an poly-time algorithm for #Hardcorenorm(μ, Δ), the box shrinking procedure gives us a polynomial time algorithm to detect whether $A = 0$ or not and compute $1/y^* = -A/B$ provided not both A and B are equal to 0.

(Summary)

- Use 'box shrinking' to compute ratios $R_{G,v}$ exactly.
- Use ratios to compute $Z_H(\mu)$ exactly with telescoping product.
- Be careful when and where to 'trust' the algorithm.

Thank you for your attention!