Zero-free regions and approximation algorithms for graph polynomials

Viresh Patel

University of Amsterdam

Queen Mary Algorithms Day
17 July 2018

Joint work with Guus Regts, UvA

Computational Counting

Is there a polynomial-time algorithm to count

1) spanning trees in a graph?
2) independent sets of a graph?
3) proper q-colourings of a graph?

Computational Counting

Is there a polynomial-time algorithm to count

1) spanning trees in a graph?
2) independent sets of a graph?
3) proper q-colourings of a graph?

- 1 Yes - matrix tree theorem
- 2,3 Problems are $\#P$-hard (on bounded degree graphs)

Is there a polynomial-time algorithm to count

1) spanning trees in a graph?
2) independent sets of a graph?
3) proper q-colourings of a graph?

- 1 Yes - matrix tree theorem
- 2,3 Problems are $\#P$-hard (on bounded degree graphs)

Is there a polynomial-time algorithm to approximately count

- independent sets of a graph?
- proper q-colourings of a graph?

Graph polynomials

Let $G = (V, E)$ be an n-vertex graph.

- Independence Polynomial

$$Z_G(\lambda) = \sum_{k=0}^{\alpha(G)} (\text{\# indep sets of size } k) \cdot \lambda^k.$$

$Z_G(1) =$number of independent sets

Graph polynomials

Let $G = (V, E)$ be an n-vertex graph.

- Independence Polynomial

$$Z_G(\lambda) = \sum_{k=0}^{\alpha(G)} (\text{\# indep sets of size } k) \cdot \lambda^k.$$

$Z_G(1) =$number of independent sets

- Chromatic Polynomial

$$\chi_G(q) = \text{\# proper q-colourings of G} .$$

Turns out to be a polynomial in q of degree n.

Outline

Discuss general algorithmic technique for approximate counting.

Applicable to approximately evaluating

- independence polynomial
- chromatic polynomial
- Tutte polynomial

Outline

Discuss general algorithmic technique for approximate counting.

Applicable to approximately evaluating

- independence polynomial
- chromatic polynomial
- Tutte polynomial

for bounded degree graphs

Outline

Discuss general algorithmic technique for approximate counting.

Applicable to approximately evaluating

- independence polynomial
- chromatic polynomial
- Tutte polynomial

for bounded degree graphs

Draw links between

- existence of fast approximate counting algorithms
- locations of (complex) roots of certain polynomials
- existence of certain types of FPT counting algorithms

Fully Polynomial Time Approximation Scheme (FPTAS)

Suppose f is a graph parameter,
(e.g. $f(G) = \#$ independent sets in $G = (V, E)$).

An FPTAS is an algorithm that, for all $0 < \varepsilon < 1$,

- estimates $f(G)$ within a multiplicative factor $1 \pm \varepsilon$
- in time polynomial in $n = |V|$ and ε^{-1}.

Fully Polynomial Time Approximation Scheme (FPTAS)

Suppose f is a graph parameter,
(e.g. $f(G) = \#$ independent sets in $G = (V, E)$).

An FPTAS is an algorithm that, for all $0 < \varepsilon < 1$,

- estimates $f(G)$ within a multiplicative factor $1 \pm \varepsilon$
- in time polynomial in $n = |V|$ and ε^{-1}.

An FPRAS is a randomised algorithm that, for all $0 < \varepsilon < 1$,

- estimates $f(G)$ within a multiplicative factor $1 \pm \varepsilon$
- in time polynomial in $n = |V|$ and ε^{-1}
- with probability $\geq \frac{3}{4}$.

Independence Polynomial

Let $G = (V, E)$ with $|V| = n$ and

$$Z_G(\lambda) = \sum_{k=0}^{\alpha(G)} \#\text{ indep sets of size } k \cdot \lambda^k.$$

Independence Polynomial

Let $G = (V, E)$ with $|V| = n$ and

$$Z_G(\lambda) = \sum_{k=0}^{\alpha(G)} \#\text{ indep sets of size } k \cdot \lambda^k.$$

$\Delta(G) \leq \Delta$

- $0 \leq \lambda < \lambda_c \implies \exists$ FPTAS for $Z_G(\lambda)$ (Weitz)
- $\lambda > \lambda_c \implies \nexists$ FPTAS for $Z_G(\lambda)$ unless RP = NP (Sly-Sun)

where

$$\lambda_c = \lambda_c(\Delta) := \frac{(\Delta - 1)^{\Delta - 1}}{(\Delta - 2)^{\Delta}}.$$

Three methods for approximate counting

- Markov chain Monte Carlo (Jerrum)
- Correlation decay (Weitz)
- Taylor polynomial interpolation (Barvinok)

Taylor Polynomial Interpolation Method (Barvinok)

- Let $p = p_G$ be a (graph) polynomial of degree n.
- Assume $p(z) \neq 0$ for all $|z| \leq C$ for some $C > 0$. ($z \in \mathbb{C}$)
- Let $f(z) = \ln p(z)$ for $|z| < C$ and let

$$T_m(z) = \sum_{k=0}^{m} f^{(k)}(0) \frac{z^k}{k!}.$$

Taylor Polynomial Interpolation Method (Barvinok)

- Let $p = p_G$ be a (graph) polynomial of degree n.
- Assume $p(z) \neq 0$ for all $|z| \leq C$ for some $C > 0$. ($z \in \mathbb{C}$)
- Let $f(z) = \ln p(z)$ for $|z| < C$ and let

$$T_m(z) = \sum_{k=0}^{m} f^{(k)}(0) \frac{z^k}{k!}.$$

Then for $m = O(\ln(n/\varepsilon))$ we have that

$$|f(z) - T_m(z)| \leq \varepsilon.$$

$$\implies \quad \exp T_m(z) = (1 \pm 2\varepsilon e^{i\theta}) p(z) \text{ for some } \theta \in [0, 2\pi].$$

Taylor Polynomial Interpolation Method (Barvinok)

- Let $p = p_G$ be a (graph) polynomial of degree n.
- Assume $p(z) \neq 0$ for all $|z| \leq C$ for some $C > 0$. ($z \in \mathbb{C}$)
- Let $f(z) = \ln p(z)$ for $|z| < C$ and let

$$T_m(z) = \sum_{k=0}^{m} f^{(k)}(0) \frac{z^k}{k!}.$$

Then for $m = O(\ln(n/\varepsilon))$ we have that

$$|f(z) - T_m(z)| \leq \varepsilon.$$

$$\Longrightarrow \quad \exp T_m(z) = (1 \pm 2\varepsilon e^{i\theta}) p(z) \text{ for some } \theta \in [0, 2\pi].$$

Recipe for FPTAS

- Identify zero-free region of p containing z (inc. non-disks).
- Efficiently compute $f^{(k)}(0)$ for $k = 0, \ldots, O(\ln n/\varepsilon)$.

How to compute derivatives

Let p be a graph polynomial and G an n-vertex graph. Suppose

$$p_G(z) = a_0 + a_1 z + \cdots + a_n z^n.$$

Wish to compute $f^{(k)}(0)$ for $k = 1, \ldots, m = \ln(n/\varepsilon)$ where

$$f(z) = \ln p_G(z).$$

How to compute derivatives

Let p be a graph polynomial and G an n-vertex graph. Suppose

$$p_G(z) = a_0 + a_1 z + \cdots + a_n z^n.$$

Wish to compute $f^{(k)}(0)$ for $k = 1, \ldots, m = \ln(n/\varepsilon)$ where

$$f(z) = \ln p_G(z).$$

Observation

If we can compute $a_0, \ldots, a_m$, then we can compute $f^{(0)}(0), f^{(1)}(0), \ldots, f^{(m)}(0)$

How to compute derivatives

Let p be a graph polynomial and G an n-vertex graph. Suppose

$$p_G(z) = a_0 + a_1 z + \cdots + a_n z^n.$$

Wish to compute $f^{(k)}(0)$ for $k = 1, \ldots, m = \ln(n/\varepsilon)$ where

$$f(z) = \ln p_G(z).$$

Observation

If we can compute $a_0, \ldots, a_m$, then we can compute $f^{(0)}(0), f^{(1)}(0), \ldots, f^{(m)}(0)$

$$p'(z) = p(z) f'(z)$$

$$k! a_k = \sum_{j=0}^{k-1} \binom{k-1}{j} a_j f^{(m-j)}(0) \qquad k = 1, \ldots, m$$

Example - independence polynomial

$$Z_G(z) = \sum_{i=0}^{\alpha(G)} a_i z^i$$

where $a_i = a_i(G) = \#$ indep sets of size i in G.

Example - independence polynomial

$$Z_G(z) = \sum_{i=0}^{\alpha(G)} a_i z^i$$

where $a_i = a_i(G) = \#$ indep sets of size i in G.

How do we compute $a_1, \ldots, a_m$ for $m = O(\ln n/\varepsilon)$?

- Check all sets of size $\leq m$: takes $n^{O(m)} = n^{O(\ln n - \ln \varepsilon)}$ time.
- There is a faster way to do this for bounded degree graphs!

Example - independence polynomial

$$Z_G(z) = \sum_{i=0}^{\alpha(G)} a_i z^i$$

where $a_i = a_i(G) = \#$ indep sets of size i in G.

How do we compute $a_1, \ldots, a_m$ for $m = O(\ln n/\varepsilon)$?

- Check all sets of size $\leq m$: takes $n^{O(m)} = n^{O(\ln n - \ln \varepsilon)}$ time.
- There is a faster way to do this for bounded degree graphs!

Lemma (P., Regts)

If $\Delta(G) \leq \Delta$, we can compute

$$a_k = a_k(G) = \#\textit{indep sets of size } k \textit{ in } G$$

in time poly(n)c^k, where $c = c(\Delta)$ is a constant.

Lemma (P., Regts)

If $\Delta(G) \leq \Delta$, we can compute $a_k = a_k(G) = \mathrm{ind}(\circ^k, G)$ in time $c^k n^{O(1)}$, where $c = c(\Delta)$.

Write $\mathrm{ind}(H, G) := \#$ induced copies of H in G

Lemma (P., Regts)

If $\Delta(G) \leq \Delta$, we can compute $a_k = a_k(G) = \mathrm{ind}(\circ^k, G)$ in time $c^k n^{O(1)}$, where $c = c(\Delta)$.

Write $\mathrm{ind}(H, G) := \#$ induced copies of H in G

Can compute $\mathrm{ind}(H, G)$ in time $c^k n^{O(1)}$ where

$|H| = k,\ H$ connected and $|G| = n,\ \Delta(G) \leq \Delta$.

Lemma (P., Regts)

If $\Delta(G) \leq \Delta$, we can compute $a_k = a_k(G) = \mathrm{ind}(\circ^k, G)$ in time $c^k n^{O(1)}$, where $c = c(\Delta)$.

Write $\mathrm{ind}(H, G) := \#$ induced copies of H in G

Can compute $\mathrm{ind}(H, G)$ in time $c^k n^{O(1)}$ where

$|H| = k$, H connected and $|G| = n$, $\Delta(G) \leq \Delta$.

Let $\eta_1, \ldots, \eta_d$ be the roots of $Z_G(\lambda) = \sum a_r \lambda^r$ Let

$p_k = p_k(G) = \eta_1^{-k} + \cdots + \eta_d^{-k}$ (the kth inverse power sum).

Lemma (P., Regts)

If $\Delta(G) \leq \Delta$, we can compute $a_k = a_k(G) = \mathrm{ind}(\circ^k, G)$ in time $c^k n^{O(1)}$, where $c = c(\Delta)$.

Write $\mathrm{ind}(H, G) := \#$ induced copies of H in G

Can compute $\mathrm{ind}(H, G)$ in time $c^k n^{O(1)}$ where

$|H| = k$, H connected and $|G| = n$, $\Delta(G) \leq \Delta$.

Let $\eta_1, \ldots, \eta_d$ be the roots of $Z_G(\lambda) = \sum a_r \lambda^r$ Let

$p_k = p_k(G) = \eta_1^{-k} + \cdots + \eta_d^{-k}$ (the kth inverse power sum).

$$a_0 p_t + a_1 p_{t-1} + \cdots + a_{t-1} p_1 = -t a_t \quad \forall t = 1, 2, \ldots$$

$$p_k(G) = \sum_{\substack{|H| \leq k \\ H \text{ connected}}} c_H \cdot \mathrm{ind}(H, G)$$

Can compute $p_1, \ldots, p_k$ and $a_1, \ldots, a_k$ in time $c^k n^{O(1)}$.

Lemma (P., Regts)

If $\Delta(G) \leq \Delta$, we can compute $a_k = a_k(G) = \mathrm{ind}(\circ^k, G)$ in time $c^k n^{O(1)}$, where $c = c(\Delta)$.

Write $\mathrm{ind}(H, G) := \#$ induced copies of H in G

Can compute $\mathrm{ind}(H, G)$ in time $c^k n^{O(1)}$ where

$|H| = k$, H connected and $|G| = n$, $\Delta(G) \leq \Delta$.

Let $\eta_1, \ldots, \eta_d$ be the roots of $Z_G(\lambda) = \sum a_r \lambda^r$ Let

$p_k = p_k(G) = \eta_1^{-k} + \cdots + \eta_d^{-k}$ (the kth inverse power sum).

$$a_0 p_t + a_1 p_{t-1} + \cdots + a_{t-1} p_1 = -t a_t \quad \forall t = 1, 2, \ldots$$

$$p_k(G) = \sum_{\substack{|H| \leq k \\ H \text{ connected}}} c_H \cdot \mathrm{ind}(H, G)$$

Lemma (P., Regts)

If $\Delta(G) \leq \Delta$, we can compute $a_k = a_k(G) = \mathrm{ind}(\circ^k, G)$ in time $c^k n^{O(1)}$, where $c = c(\Delta)$.

Write $\mathrm{ind}(H, G) := \#$ induced copies of H in G

Can compute $\mathrm{ind}(H, G)$ in time $c^k n^{O(1)}$ where

$|H| = k$, H connected and $|G| = n$, $\Delta(G) \leq \Delta$.

Let $\eta_1, \ldots, \eta_d$ be the roots of $Z_G(\lambda) = \sum a_r \lambda^r$ Let

$p_k = p_k(G) = \eta_1^{-k} + \cdots + \eta_d^{-k}$ (the kth inverse power sum).

$$a_0 p_t + a_1 p_{t-1} + \cdots + a_{t-1} p_1 = -t a_t \quad \forall t = 1, 2, \ldots$$

$$p_k(G) = \sum_{\substack{|H| \leq k \\ H \text{ connected}}} c_H \cdot \mathrm{ind}(H, G)$$

Can compute $p_1, \ldots, p_k$ and $a_1, \ldots, a_k$ in time $c^k n^{O(1)}$.

Theorem (P., Regts)

Suppose $Z_G(z) \neq 0$ *for all* $|z| \leq C$ *and* $\Delta(G) \leq \Delta$.

Then $\exists$ *FPTAS to compute* $Z_G(z)$ *for* $|z| < C$ *and* $\Delta(G) \leq \Delta$.

Theorem (P., Regts)

Suppose $Z_G(z) \neq 0$ for all $|z| \leq C$ and $\Delta(G) \leq \Delta$.

Then $\exists$ FPTAS to compute $Z_G(z)$ for $|z| < C$ and $\Delta(G) \leq \Delta$.

$\lambda^*(\Delta) := \frac{(\Delta-1)^{\Delta-1}}{\Delta^\Delta}$ $\lambda_c(\Delta) = \frac{(\Delta-1)^{\Delta-1}}{(\Delta-2)^\Delta}$ (Note $\lambda^* < \lambda_c$)

Theorem

We have $Z_G(z) \neq 0$ for all $z \in D$ and $\Delta(G) \leq \Delta$ where

(1) $D = \{z \ : \ |z| \leq \lambda^*\}$ *(Dobushin, Shearer)*

(2) $D =$ *open region containing* $[0, \lambda_c)$ *(Peters, Regts)*

Theorem (P., Regts)

Suppose $Z_G(z) \neq 0$ for all $|z| \leq C$ and $\Delta(G) \leq \Delta$.

Then $\exists$ FPTAS to compute $Z_G(z)$ for $|z| < C$ and $\Delta(G) \leq \Delta$.

$\lambda^*(\Delta) := \frac{(\Delta-1)^{\Delta-1}}{\Delta^\Delta}$ $\quad \lambda_c(\Delta) = \frac{(\Delta-1)^{\Delta-1}}{(\Delta-2)^\Delta}$ $\quad$ (Note $\lambda^* < \lambda_c$)

Theorem

We have $Z_G(z) \neq 0$ for all $z \in D$ and $\Delta(G) \leq \Delta$ where

1. $D = \{z \ : \ |z| \leq \lambda^*\}$ *(Dobushin, Shearer)*
2. $D =$ *open region containing* $[0, \lambda_c)$ *(Peters, Regts)*

Implies the following:

There is an FPTAS for computing $Z_G(z)$ if $z \in D$ and $\Delta(G) \leq \Delta$.

Theorem (P., Regts)

Suppose $Z_G(z) \neq 0$ for all $|z| \leq C$ and $\Delta(G) \leq \Delta$.

Then $\exists$ FPTAS to compute $Z_G(z)$ for $|z| < C$ and $\Delta(G) \leq \Delta$.

$\lambda^*(\Delta) := \frac{(\Delta-1)^{\Delta-1}}{\Delta^\Delta}$ $\quad \lambda_c(\Delta) = \frac{(\Delta-1)^{\Delta-1}}{(\Delta-2)^\Delta}$ $\quad$ (Note $\lambda^* < \lambda_c$)

Theorem

We have $Z_G(z) \neq 0$ for all $z \in D$ and $\Delta(G) \leq \Delta$ where

(1) $D = \{z \ : \ |z| \leq \lambda^*\}$ *(Dobushin, Shearer)*

(2) $D =$ *open region containing* $[0, \lambda_c)$ *(Peters, Regts)*

Implies the following:

There is an FPTAS for computing $Z_G(z)$ if $z \in D$ and $\Delta(G) \leq \Delta$.

- Recover result of Weitz and more
- Will see a complete complexity picture in next talk!

Independence polynomial - special graph classes

Theorem (Chudnovsky, Seymour)

$Z_G(z) \neq 0$ *whenever* G *is claw-free and* $z \in \mathbb{C} \setminus (-\infty, 0)$

Independence polynomial - special graph classes

Theorem (Chudnovsky, Seymour)

$Z_G(z) \neq 0$ whenever G is claw-free and $z \in \mathbb{C} \setminus (-\infty, 0)$

So our method implies (after conformal transformation)

There is an FPTAS to evaluate $Z_G(z)$ whenever G is claw-free with $\Delta(G) \leq \Delta$ and $z \in \mathbb{C} \setminus (-\infty, 0)$.

Independence polynomial - special graph classes

Why are claw-free graphs interesting?

- The line graph $L(G)$ of a graph G is claw-free.
- Matchings in $G \leftrightarrow$ independent sets in $L(G)$
- Hence $Z_{L(G)}(\lambda) = M_G(\lambda) := \sum_{\text{matchings } M \text{ of } G} \lambda^{|M|}$

Independence polynomial - special graph classes

Why are claw-free graphs interesting?

- The line graph $L(G)$ of a graph G is claw-free.
- Matchings in $G \leftrightarrow$ independent sets in $L(G)$
- Hence $Z_{L(G)}(\lambda) = M_G(\lambda) := \sum_{\text{matchings } M \text{ of } G} \lambda^{|M|}$

Theorem (Bayati, Gamarnik, Katz, Nair, and Tetali)

$\exists$ *FPTAS to compute* $M_G(\lambda)$ *for* $\Delta(G) \leq \Delta$ *and* $\lambda \in [0, \infty)$.

Independence polynomial - special graph classes

Why are claw-free graphs interesting?

- The line graph $L(G)$ of a graph G is claw-free.
- Matchings in $G \leftrightarrow$ independent sets in $L(G)$
- Hence $Z_{L(G)}(\lambda) = M_G(\lambda) := \sum_{\text{matchings } M \text{ of } G} \lambda^{|M|}$

Theorem (Bayati, Gamarnik, Katz, Nair, and Tetali)

$\exists$ *FPTAS to compute* $M_G(\lambda)$ *for* $\Delta(G) \leq \Delta$ *and* $\lambda \in [0, \infty)$.

We extended this to

- independent sets in claw-free graphs (of bounded degree).
- almost all complex λ.

Independence polynomial - special graph classes

Why are claw-free graphs interesting?

- The line graph $L(G)$ of a graph G is claw-free.
- Matchings in $G \leftrightarrow$ independent sets in $L(G)$
- Hence $Z_{L(G)}(\lambda) = M_G(\lambda) := \sum_{\text{matchings } M \text{ of } G} \lambda^{|M|}$

Theorem (Bayati, Gamarnik, Katz, Nair, and Tetali)

$\exists$ *FPTAS to compute* $M_G(\lambda)$ *for* $\Delta(G) \leq \Delta$ *and* $\lambda \in [0, \infty)$.

We extended this to

- independent sets in claw-free graphs (of bounded degree).
- almost all complex λ.

Theorem (Jerrum and Sinclair)

$\exists$ *FPRAS to compute* $M_G(\lambda)$ *for* $\lambda \in [0, \infty)$ *and* $\forall G$.

General result

Definition

Let $p = p_G$ be a graph polynomial, i.e.

$$p_G(z) = \sum_k a_k(G) z^k.$$

Call p a bounded induced graph counting polynomial (BIGCP) if

- $p_{G_1 \cup G_2} = p_{G_1} \cdot p_{G_2}$
- $a_k(G) = \sum_{|H|=O(k)} s_{H,k} \cdot \mathrm{ind}(H, G)$
- $s_{H,k}$ can be computed in $\exp(O(k))$-time

c.f. independence polynomial

General result

Definition

Let $p = p_G$ be a graph polynomial, i.e.

$$p_G(z) = \sum_k a_k(G) z^k.$$

Call p a bounded induced graph counting polynomial (BIGCP) if

- $p_{G_1 \cup G_2} = p_{G_1} \cdot p_{G_2}$
- $a_k(G) = \sum_{|H|=O(k)} s_{H,k} \cdot \mathrm{ind}(H, G)$
- $s_{H,k}$ can be computed in $\exp(O(k))$-time

c.f. independence polynomial

Theorem (P., Regts)

Let p be a BIGCP with $p_G(z) \neq 0$ for $|z| \leq K = K(\Delta)$ and $\Delta(G) \leq \Delta$.

$\exists$ FPTAS to compute $p_G(z)$ for $|z| \leq K$ and $\Delta(G) \leq \Delta$.

Chromatic polynomial

For a graph $G = (V, E)$

$$\chi_G(q) = \text{\# proper q-colourings of G};$$

hence $\chi_{G_1 \cup G_2}(q) = \chi_{G_1}(q) \cdot \chi_{G_2}(q)$

Chromatic polynomial

For a graph $G = (V, E)$

$$\chi_G(q) = \#\text{ proper q-colourings of G;}$$

hence $\chi_{G_1 \cup G_2}(q) = \chi_{G_1}(q) \cdot \chi_{G_2}(q)$

Random cluster model formulation

$$\chi_G(q) = \sum_{A \subseteq E} (-1)^{|A|} q^{k(A)} =: \sum_i a_i(G) q^i,$$

where

- $a_n = 1$
- $a_{n-1} = (-1)\mathrm{ind}(e, G)$
- $a_{n-2} = \mathrm{ind}(P_3, G) - \mathrm{ind}(K_3, G) + \mathrm{ind}(2K_2, G)$ etc

Chromatic polynomial

For a graph $G = (V, E)$

$$\chi_G(q) = \text{\# proper q-colourings of G;}$$

hence $\chi_{G_1 \cup G_2}(q) = \chi_{G_1}(q) \cdot \chi_{G_2}(q)$

Random cluster model formulation

$$\chi_G(q) = \sum_{A \subseteq E} (-1)^{|A|} q^{k(A)} =: \sum_i a_i(G) q^i,$$

where

- $a_n = 1$
- $a_{n-1} = (-1)\mathrm{ind}(e, G)$
- $a_{n-2} = \mathrm{ind}(P_3, G) - \mathrm{ind}(K_3, G) + \mathrm{ind}(2K_2, G)$ etc

Hence $z^n \chi_G(z^{-1})$ is a BIGCP

Chromatic polynomial

The polynomial $z^n \chi_G(z^{-1})$ is a BIGCP

Theorem (Jackson, Procacci and Sokal)

$\chi_G(z) \neq 0$ *whenever* $\Delta(G) \leq \Delta$ *and* $|z| \geq K(\Delta) = 6.91\Delta$.

Chromatic polynomial

The polynomial $z^n \chi_G(z^{-1})$ is a BIGCP

Theorem (Jackson, Procacci and Sokal)

$\chi_G(z) \neq 0$ whenever $\Delta(G) \leq \Delta$ and $|z| \geq K(\Delta) = 6.91\Delta$.

Using our method, this implies

$\exists$ FPTAS to compute $\chi_G(z)$ for $|z| \geq 6.91\Delta$ and $\Delta(G) \leq \Delta$.

Chromatic polynomial

The polynomial $z^n \chi_G(z^{-1})$ is a BIGCP

Theorem (Jackson, Procacci and Sokal)

$\chi_G(z) \neq 0$ *whenever* $\Delta(G) \leq \Delta$ *and* $|z| \geq K(\Delta) = 6.91\Delta$.

Using our method, this implies

$\exists$ FPTAS to compute $\chi_G(z)$ for $|z| \geq 6.91\Delta$ and $\Delta(G) \leq \Delta$.

- FPTAS for $q \geq 2.58\Delta$ (Lu and Yin)
- FPRAS for $q \geq \frac{11}{6}\Delta$ (Vigoda)
- FPRAS for $q > (\frac{11}{6} - \varepsilon)\Delta$ (Delcourt-Perarnau-Postle and Chen-Moitra)
- No FPTAS for $k < \Delta$ unless $P = NP$
- **FP(RT)AS conjectured for** $q > \Delta$

Chromatic polynomial

The polynomial $z^n \chi_G(z^{-1})$ is a BIGCP

Conjecture (Sokal)

$\chi_G(z) \neq 0$ *if* $\Re(z) > \Delta(G)$.

Chromatic polynomial

The polynomial $z^n \chi_G(z^{-1})$ is a BIGCP

Conjecture (Sokal)

$\chi_G(z) \neq 0$ *if* $\Re(z) > \Delta(G)$.

Our method shows Sokal's conjecture implies

Conjecture (Folklore)

There is an FPTAS for $\chi_G(q)$ *whenever* $q > \Delta(G)$.

Tutte polynomial

For a graph $G = (V, E)$

$$T_G(q, w) = \sum_{A \subseteq E} w^{|A|} q^{k(A)}$$

So $T_G(q, -1) = \chi_G(q)$

Tutte polynomial

For a graph $G = (V, E)$

$$T_G(q, w) = \sum_{A \subseteq E} w^{|A|} q^{k(A)}$$

So $T_G(q, -1) = \chi_G(q)$

Tutte polynomial captures number of

- forests, spanning trees, spanning subgraphs, acyclic orientations, nowhere zero flows, q state Potts model

Tutte polynomial

For a graph $G = (V, E)$

$$T_G(q, w) = \sum_{A \subseteq E} w^{|A|} q^{k(A)}$$

So $T_G(q, -1) = \chi_G(q)$

Tutte polynomial captures number of

- forests, spanning trees, spanning subgraphs, acyclic orientations, nowhere zero flows, q state Potts model

As before $T_{G_1 \cup G_2}(q, w) = T_{G_1}(q, w) \cdot T_{G_2}(q, w)$

For each fixed $w \in \mathbb{C}$, $z^n T_G(z^{-1}, w)$ is a BIGCP

Tutte polynomial

For each fixed $w \in \mathbb{C}$, $z^n T_G(z^{-1}, w)$ is a BIGCP

Theorem (Jackson, Procacci and Sokal)

$\exists K := K(\Delta, z_2)$ *s.t.* $T_G(z_1, z_2) \neq 0$ *whenever* $\Delta(G) \leq \Delta$ *and* $|z_1| \geq K(\Delta, z_2)$.

Tutte polynomial

For each fixed $w \in \mathbb{C}$, $z^n T_G(z^{-1}, w)$ is a BIGCP

Theorem (Jackson, Procacci and Sokal)

$\exists K := K(\Delta, z_2)$ *s.t.* $T_G(z_1, z_2) \neq 0$ *whenever* $\Delta(G) \leq \Delta$ *and* $|z_1| \geq K(\Delta, z_2)$.

So our method implies

There is an FPTAS to evaluate $T_G(z, w)$ for graphs of maximum degree Δ whenever $|z| \geq K(\Delta, w)$.

Summary

Shown how Taylor Polynomial Interpolation Method can give FPTAS for approximating graph polynomials on bounded degree graphs.

Summary

Shown how Taylor Polynomial Interpolation Method can give FPTAS for approximating graph polynomials on bounded degree graphs.

- Generally applicable to several graph polynomials e.g.
  - Independence polynomial
  - Chromatic / Tutte polynomial
  - Partition functions of spin models
  - Partition functions of edge-colouring models

Summary

Shown how Taylor Polynomial Interpolation Method can give FPTAS for approximating graph polynomials on bounded degree graphs.

- Generally applicable to several graph polynomials e.g.
  - Independence polynomial
  - Chromatic / Tutte polynomial
  - Partition functions of spin models
  - Partition functions of edge-colouring models
- Deterministic algorithm

Summary

Shown how Taylor Polynomial Interpolation Method can give FPTAS for approximating graph polynomials on bounded degree graphs.

- Generally applicable to several graph polynomials e.g.
  - Independence polynomial
  - Chromatic / Tutte polynomial
  - Partition functions of spin models
  - Partition functions of edge-colouring models
- Deterministic algorithm
- Works for complex evaluations

Summary

Shown how Taylor Polynomial Interpolation Method can give FPTAS for approximating graph polynomials on bounded degree graphs.

- Generally applicable to several graph polynomials e.g.
  - Independence polynomial
  - Chromatic / Tutte polynomial
  - Partition functions of spin models
  - Partition functions of edge-colouring models
- Deterministic algorithm
- Works for complex evaluations
- Links FPTAS for $P_G(t)$ and locations of its roots
  - $\chi_G(z) \neq 0$ for $\Re(z) > \Delta(G) \implies$ FPTAS for $\chi_G(q)$ if $q \geq \Delta(G)$