CS 263: Counting and Sampling

Nima Anari

Stanford University

slides for

Zeros of Polynomials

Review

- Linear tilts:

$$\nu(x) = \underbrace{(1 + \langle w, x - \mathrm{mean}(\mu)\rangle)}_{\text{linear tilt}} \mu(x)$$

Review

- Linear tilts:

$$\nu(x) = \underbrace{(1 + \langle w, x - \mathrm{mean}(\mu)\rangle)}_{\text{linear tilt}}\,\mu(x)$$

- Covariance evolution: $\mathrm{cov}(\mu) =$

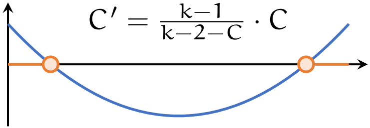
$$\mathbb{E}[\mathrm{cov}(\nu)] + \mathrm{cov}(\mu)\,\mathrm{cov}(w)\,\mathrm{cov}(\mu)$$

Review

- Linear tilts:

$$\nu(x) = \underbrace{(1 + \langle w, x - \mathsf{mean}(\mu)\rangle)}_{\text{linear tilt}} \mu(x)$$

- Covariance evolution: $\mathsf{cov}(\mu) = \mathbb{E}[\mathsf{cov}(\nu)] + \mathsf{cov}(\mu)\,\mathsf{cov}(w)\,\mathsf{cov}(\mu)$
- Trickle down: C-SI for links means either μ is disconnected or C'-SI:



Review

- Linear tilts:

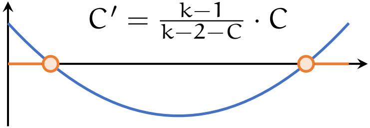
$$\nu(x) = \underbrace{(1 + \langle w, x - \mathrm{mean}(\mu)\rangle)}_{\text{linear tilt}}\,\mu(x)$$

- Covariance evolution: $\mathrm{cov}(\mu) =$

$$\mathbb{E}[\mathrm{cov}(\nu)] + \mathrm{cov}(\mu)\,\mathrm{cov}(w)\,\mathrm{cov}(\mu)$$

- Trickle down: C-SI for links means either μ is disconnected or C'-SI:

$$C' = \frac{k-1}{k-2-C} \cdot C$$



Review

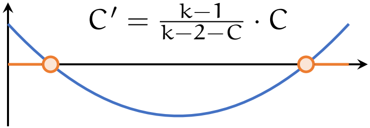
- Linear tilts:

$$\nu(x) = \underbrace{(1 + \langle w, x - \text{mean}(\mu)\rangle)}_{\text{linear tilt}}\, \mu(x)$$

- Covariance evolution: $\text{cov}(\mu) =$

$$\mathbb{E}[\text{cov}(\nu)] + \text{cov}(\mu)\,\text{cov}(w)\,\text{cov}(\mu)$$

- Trickle down: C-SI for links means either μ is disconnected or C'-SI:



- Generating polynomial:

$$g_\mu(z_1, \ldots, z_n) = \sum_S \mu(S) \prod_{i \in S} z_i.$$

Review

- Linear tilts:

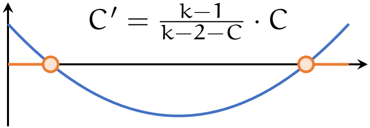
$$\nu(x) = \underbrace{(1 + \langle w, x - \text{mean}(\mu)\rangle)}_{\text{linear tilt}} \mu(x)$$

- Covariance evolution: $\text{cov}(\mu) =$

$$\mathbb{E}[\text{cov}(\nu)] + \text{cov}(\mu)\,\text{cov}(w)\,\text{cov}(\mu)$$

- Trickle down: C-SI for links means either μ is disconnected or C'-SI:

$$C' = \frac{k-1}{k-2-C} \cdot C$$



- Generating polynomial:

$$g_\mu(z_1, \ldots, z_n) = \sum_S \mu(S) \prod_{i \in S} z_i.$$

- 1 spectral independence same as

$$\lambda_2(\nabla^2 g(\mathbb{1})) \leqslant 0 \leftrightarrow \nabla^2 \log g(\mathbb{1}) \preceq 0$$

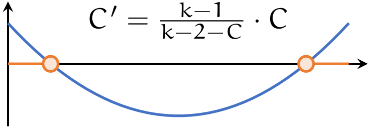
Review

- Linear tilts:

$$\nu(x) = \underbrace{(1 + \langle w, x - \mathrm{mean}(\mu)\rangle)}_{\text{linear tilt}} \mu(x)$$

- Covariance evolution: $\mathrm{cov}(\mu) = \mathbb{E}[\mathrm{cov}(\nu)] + \mathrm{cov}(\mu)\,\mathrm{cov}(w)\,\mathrm{cov}(\mu)$
- Trickle down: C-SI for links means either μ is disconnected or C'-SI:

$$C' = \frac{k-1}{k-2-C} \cdot C$$



- Generating polynomial:

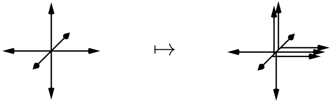
$$g_\mu(z_1, \ldots, z_n) = \sum_S \mu(S) \prod_{i \in S} z_i.$$

- 1 spectral independence same as

$$\lambda_2(\nabla^2 g(\mathbb{1})) \leqslant 0 \leftrightarrow \nabla^2 \log g(\mathbb{1}) \preceq 0$$

- Exponential tilts/external fields:

$$\nu(x) \propto \underbrace{\exp(\langle w, x\rangle)}_{\text{exponential tilt}} \mu(x)$$



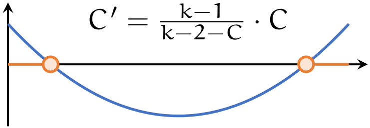
Review

- Linear tilts:

$$\nu(x) = \underbrace{(1 + \langle w, x - \mathrm{mean}(\mu)\rangle)}_{\text{linear tilt}}\,\mu(x)$$

- Covariance evolution: $\mathrm{cov}(\mu) = \mathbb{E}[\mathrm{cov}(\nu)] + \mathrm{cov}(\mu)\,\mathrm{cov}(w)\,\mathrm{cov}(\mu)$
- Trickle down: C-SI for links means either μ is disconnected or C'-SI:

$$C' = \frac{k-1}{k-2-C} \cdot C$$



- Generating polynomial:

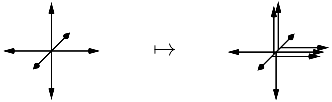
$$g_\mu(z_1, \ldots, z_n) = \sum_S \mu(S) \prod_{i \in S} z_i.$$

- 1 spectral independence same as

$$\lambda_2(\nabla^2 g(\mathbb{1})) \leqslant 0 \leftrightarrow \nabla^2 \log g(\mathbb{1}) \preceq 0$$

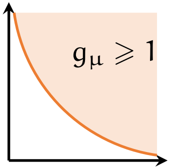
- Exponential tilts/external fields:

$$\nu(x) \propto \underbrace{\exp(\langle w, x\rangle)}_{\text{exponential tilt}}\,\mu(x)$$



- Exp tilts of matroids are 1-SI:

$$\nabla^2 \log g \preceq 0 \text{ on } \mathbb{R}^n_{>0}$$

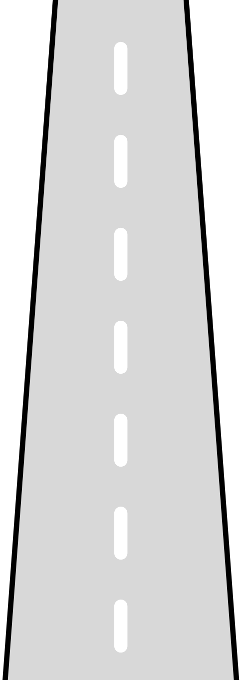


Entropic Independence

- Fractional log-concavity

Polynomial Interpolation

- Matching polynomial
- Taylor approximation
- Riemann mappings

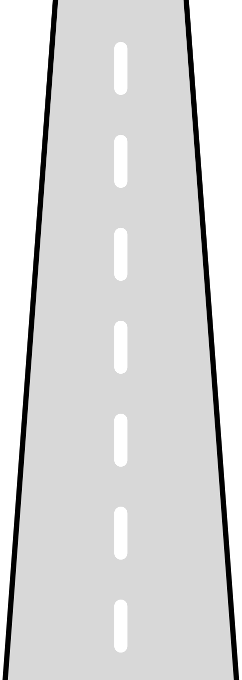


Entropic Independence

- Fractional log-concavity

Polynomial Interpolation

- Matching polynomial
- Taylor approximation
- Riemann mappings



Entropic Independence

For all distributions ν,

$$\mathcal{D}_{\mathsf{KL}}(\nu D_{k \to 1} \parallel \mu D_{k \to 1}) \leqslant \frac{C}{k} \cdot \mathcal{D}_{\mathsf{KL}}(\nu \parallel \mu).$$

Entropic Independence

For all distributions ν,

$$\mathcal{D}_{KL}(\nu D_{k\to 1} \parallel \mu D_{k\to 1}) \leqslant \frac{C}{k} \cdot \mathcal{D}_{KL}(\nu \parallel \mu).$$

- The greatest aspect of entropic independence: no need to consider all ν. Enough to look at

$$\nu(x) \propto \exp(\langle w, x\rangle)\mu(x)$$

for $w \in \mathbb{R}^n$. This is an external field applied to μ.

Entropic Independence

For all distributions ν,

$$\mathcal{D}_{\mathrm{KL}}(\nu D_{k\to 1} \parallel \mu D_{k\to 1}) \leqslant \frac{C}{k} \cdot \mathcal{D}_{\mathrm{KL}}(\nu \parallel \mu).$$

- The greatest aspect of entropic independence: no need to consider all ν. Enough to look at
$$\nu(x) \propto \exp(\langle w, x\rangle)\mu(x)$$
for $w \in \mathbb{R}^n$. This is an external field applied to μ.
- If $q = (q_1, \dots, q_n)$ is some distribution on $[n] = \binom{[n]}{1}$, then
$$\inf\{\mathcal{D}_{\mathrm{KL}}(\nu \parallel \mu) \mid \nu D_{k\to 1} = q\} = -\log\left(\inf_{z_1,\dots,z_n>0} \frac{\sum_S \mu(S)\prod_{i\in S} z_i}{z_1^{kq_1}\cdots z_n^{kq_n}}\right).$$

- Let $\nu/\mu = f$, and $q/(\mu D_{k\to 1}) = g$.
  Then convex program is

$$\inf\{\mathrm{Ent}_\mu[f] \mid U_{1\to k} f = g\}$$

- Let $\nu/\mu = f$, and $q/(\mu D_{k\to 1}) = g$.
  Then convex program is
  $$\inf\{\mathrm{Ent}_\mu[f] \mid U_{1\to k} f = g\}$$
- The Lagrangian is
  $$\mathrm{Ent}_\mu[f] - \langle \lambda, U_{1\to k} f - g\rangle$$

- Let $\nu/\mu = f$, and $q/(\mu D_{k\to 1}) = g$. Then convex program is

$$\inf\{\mathrm{Ent}_\mu[f] \mid U_{1\to k} f = g\}$$

- The Lagrangian is

$$\mathrm{Ent}_\mu[f] - \langle \lambda, U_{1\to k} f - g\rangle$$

- Fixing λ, optimality cond for f is

$$\nabla_f \mathrm{Ent}_\mu[f] = \lambda^\intercal U_{1\to k}$$

- Let $\nu/\mu = f$, and $q/(\mu D_{k\to 1}) = g$.
  Then convex program is
  $$\inf\{\mathrm{Ent}_\mu[f] \mid U_{1\to k} f = g\}$$
- The Lagrangian is
  $$\mathrm{Ent}_\mu[f] - \langle \lambda, U_{1\to k} f - g\rangle$$
- Fixing λ, optimality cond for f is
  $$\nabla_f \mathrm{Ent}_\mu[f] = \lambda^\intercal U_{1\to k}$$
- But we have
  $$\frac{d}{df(S)} \mathrm{Ent}_\mu[f] = \mu(S) \log \frac{f(S)}{\mathbb{E}_\mu[f]}$$

- Let $\nu/\mu = f$, and $q/(\mu D_{k\to 1}) = g$. Then convex program is

$$\inf\{\mathrm{Ent}_\mu[f] \mid U_{1\to k} f = g\}$$

- The Lagrangian is

$$\mathrm{Ent}_\mu[f] - \langle \lambda, U_{1\to k} f - g\rangle$$

- Fixing λ, optimality cond for f is

$$\nabla_f \mathrm{Ent}_\mu[f] = \lambda^\intercal U_{1\to k}$$

- But we have

$$\frac{d}{df(S)} \mathrm{Ent}_\mu[f] = \mu(S) \log \frac{f(S)}{\mathbb{E}_\mu[f]}$$

- If we let $w_i = \lambda_i / \mathbb{P}_{S\sim\mu}[i \in S]$, then

$$(\lambda^\intercal U_{1\to k})_S = \mu(S) \sum_{i\in S} w_i$$

- Let $\nu/\mu = f$, and $q/(\mu D_{k\to 1}) = g$. Then convex program is

$$\inf\{\mathrm{Ent}_\mu[f] \mid U_{1\to k} f = g\}$$

- The Lagrangian is

$$\mathrm{Ent}_\mu[f] - \langle \lambda, U_{1\to k} f - g\rangle$$

- Fixing λ, optimality cond for f is

$$\nabla_f \mathrm{Ent}_\mu[f] = \lambda^\intercal U_{1\to k}$$

- But we have

$$\frac{d}{df(S)} \mathrm{Ent}_\mu[f] = \mu(S) \log \frac{f(S)}{\mathbb{E}_\mu[f]}$$

- If we let $w_i = \lambda_i / \mathbb{P}_{S\sim\mu}[i \in S]$, then

$$(\lambda^\intercal U_{1\to k})_S = \mu(S) \sum_{i\in S} w_i$$

- Thus, the optimality condition is

$$f(S) \propto \exp(\langle w, \mathbb{1}_S\rangle)$$

- Let $\nu/\mu = f$, and $q/(\mu D_{k\to 1}) = g$. Then convex program is
$$\inf\{\mathrm{Ent}_\mu[f] \mid U_{1\to k} f = g\}$$
- The Lagrangian is
$$\mathrm{Ent}_\mu[f] - \langle \lambda, U_{1\to k} f - g\rangle$$
- Fixing λ, optimality cond for f is
$$\nabla_f \mathrm{Ent}_\mu[f] = \lambda^\intercal U_{1\to k}$$
- But we have
$$\frac{d}{df(S)} \mathrm{Ent}_\mu[f] = \mu(S) \log \frac{f(S)}{\mathbb{E}_\mu[f]}$$
- If we let $w_i = \lambda_i / \mathbb{P}_{S\sim\mu}[i \in S]$, then
$$(\lambda^\intercal U_{1\to k})_S = \mu(S) \sum_{i\in S} w_i$$
- Thus, the optimality condition is
$$f(S) \propto \exp(\langle w, \mathbb{1}_S\rangle)$$
- If we let $Z = \mathbb{E}_\mu[\exp(\langle w, \mathbb{1}_S\rangle)]$, then
$$f(S) = \exp(\langle w, \mathbb{1}_S\rangle)/Z$$

- Let $\nu/\mu = f$, and $q/(\mu D_{k\to 1}) = g$. Then convex program is
$$\inf\{\text{Ent}_\mu[f] \mid U_{1\to k} f = g\}$$
- The Lagrangian is
$$\text{Ent}_\mu[f] - \langle \lambda, U_{1\to k} f - g\rangle$$
- Fixing λ, optimality cond for f is
$$\nabla_f \text{Ent}_\mu[f] = \lambda^\intercal U_{1\to k}$$
- But we have
$$\frac{d}{df(S)} \text{Ent}_\mu[f] = \mu(S) \log \frac{f(S)}{\mathbb{E}_\mu[f]}$$
- If we let $w_i = \lambda_i / \mathbb{P}_{S\sim\mu}[i \in S]$, then
$$(\lambda^\intercal U_{1\to k})_S = \mu(S) \sum_{i\in S} w_i$$
- Thus, the optimality condition is
$$f(S) \propto \exp(\langle w, \mathbb{1}_S\rangle)$$
- If we let $Z = \mathbb{E}_\mu[\exp(\langle w, \mathbb{1}_S\rangle)]$, then
$$f(S) = \exp(\langle w, \mathbb{1}_S\rangle)/Z$$
- Note Ent is 1-homogeneous, so
$$\text{Ent}_\mu[f] = \text{Ent}_\mu[\exp(\langle w, \mathbb{1}_S\rangle)]/Z$$

- Let $\nu/\mu = f$, and $q/(\mu D_{k\to 1}) = g$. Then convex program is
$$\inf\{\mathrm{Ent}_\mu[f] \mid U_{1\to k} f = g\}$$
- The Lagrangian is
$$\mathrm{Ent}_\mu[f] - \langle \lambda, U_{1\to k} f - g\rangle$$
- Fixing λ, optimality cond for f is
$$\nabla_f \mathrm{Ent}_\mu[f] = \lambda^\intercal U_{1\to k}$$
- But we have
$$\frac{d}{df(S)} \mathrm{Ent}_\mu[f] = \mu(S) \log \frac{f(S)}{\mathbb{E}_\mu[f]}$$
- If we let $w_i = \lambda_i / \mathbb{P}_{S\sim\mu}[i \in S]$, then
$$(\lambda^\intercal U_{1\to k})_S = \mu(S) \sum_{i\in S} w_i$$
- Thus, the optimality condition is
$$f(S) \propto \exp(\langle w, \mathbb{1}_S\rangle)$$
- If we let $Z = \mathbb{E}_\mu[\exp(\langle w, \mathbb{1}_S\rangle)]$, then
$$f(S) = \exp(\langle w, \mathbb{1}_S\rangle)/Z$$
- Note Ent is 1-homogeneous, so
$$\mathrm{Ent}_\mu[f] = \mathrm{Ent}_\mu[\exp(\langle w, \mathbb{1}_S\rangle)]/Z$$
- We get that
$$\langle \lambda, U_{1\to k} f\rangle = \sum_S \mu(s) f(s) \langle w, \mathbb{1}_S\rangle$$

- Let $\nu/\mu = f$, and $q/(\mu D_{k\to 1}) = g$. Then convex program is
$$\inf\{\mathrm{Ent}_\mu[f] \mid U_{1\to k} f = g\}$$
- The Lagrangian is
$$\mathrm{Ent}_\mu[f] - \langle \lambda, U_{1\to k} f - g\rangle$$
- Fixing λ, optimality cond for f is
$$\nabla_f \mathrm{Ent}_\mu[f] = \lambda^\intercal U_{1\to k}$$
- But we have
$$\frac{d}{df(S)} \mathrm{Ent}_\mu[f] = \mu(S) \log \frac{f(S)}{\mathbb{E}_\mu[f]}$$
- If we let $w_i = \lambda_i / \mathbb{P}_{S\sim\mu}[i \in S]$, then
$$(\lambda^\intercal U_{1\to k})_S = \mu(S) \sum_{i\in S} w_i$$
- Thus, the optimality condition is
$$f(S) \propto \exp(\langle w, \mathbb{1}_S\rangle)$$
- If we let $Z = \mathbb{E}_\mu[\exp(\langle w, \mathbb{1}_S\rangle)]$, then
$$f(S) = \exp(\langle w, \mathbb{1}_S\rangle)/Z$$
- Note Ent is 1-homogeneous, so
$$\mathrm{Ent}_\mu[f] = \mathrm{Ent}_\mu[\exp(\langle w, \mathbb{1}_S\rangle)]/Z$$
- We get that
$$\langle \lambda, U_{1\to k} f\rangle = \sum_S \mu(s) f(s) \langle w, \mathbb{1}_S\rangle$$
- The Lagrangian simplifies to
$$-\log(Z) + \langle \lambda, g\rangle = k\langle w, q\rangle - \log(Z)$$

- Let $\nu/\mu = f$, and $q/(\mu D_{k\to 1}) = g$. Then convex program is
$$\inf\{\mathrm{Ent}_\mu[f] \mid U_{1\to k} f = g\}$$
- The Lagrangian is
$$\mathrm{Ent}_\mu[f] - \langle \lambda, U_{1\to k} f - g\rangle$$
- Fixing λ, optimality cond for f is
$$\nabla_f \mathrm{Ent}_\mu[f] = \lambda^\intercal U_{1\to k}$$
- But we have
$$\frac{d}{df(S)} \mathrm{Ent}_\mu[f] = \mu(S) \log \frac{f(S)}{\mathbb{E}_\mu[f]}$$
- If we let $w_i = \lambda_i / \mathbb{P}_{S\sim\mu}[i \in S]$, then
$$(\lambda^\intercal U_{1\to k})_S = \mu(S) \sum_{i\in S} w_i$$
- Thus, the optimality condition is
$$f(S) \propto \exp(\langle w, \mathbb{1}_S\rangle)$$
- If we let $Z = \mathbb{E}_\mu[\exp(\langle w, \mathbb{1}_S\rangle)]$, then
$$f(S) = \exp(\langle w, \mathbb{1}_S\rangle)/Z$$
- Note Ent is 1-homogeneous, so
$$\mathrm{Ent}_\mu[f] = \mathrm{Ent}_\mu[\exp(\langle w, \mathbb{1}_S\rangle)]/Z$$
- We get that
$$\langle \lambda, U_{1\to k} f\rangle = \sum_S \mu(s) f(s) \langle w, \mathbb{1}_S\rangle$$
- The Lagrangian simplifies to
$$-\log(Z) + \langle \lambda, g\rangle = k\langle w, q\rangle - \log(Z)$$
- If we reparameterize $z_i = e^{w_i}$, then this is
$$\log\left(\frac{\prod_{i\in[n]} z_i^{kq_i}}{g_\mu(z)}\right)$$

- Let $\nu/\mu = f$, and $q/(\mu D_{k\to 1}) = g$. Then convex program is

$$\inf\{\text{Ent}_\mu[f] \mid U_{1\to k} f = g\}$$

- The Lagrangian is

$$\text{Ent}_\mu[f] - \langle \lambda, U_{1\to k} f - g\rangle$$

- Fixing λ, optimality cond for f is

$$\nabla_f \text{Ent}_\mu[f] = \lambda^\intercal U_{1\to k}$$

- But we have

$$\frac{d}{df(S)} \text{Ent}_\mu[f] = \mu(S) \log \frac{f(S)}{\mathbb{E}_\mu[f]}$$

- If we let $w_i = \lambda_i / \mathbb{P}_{S\sim\mu}[i \in S]$, then

$$(\lambda^\intercal U_{1\to k})_S = \mu(S) \sum_{i\in S} w_i$$

- Thus, the optimality condition is

$$f(S) \propto \exp(\langle w, \mathbb{1}_S\rangle)$$

- If we let $Z = \mathbb{E}_\mu[\exp(\langle w, \mathbb{1}_S\rangle)]$, then

$$f(S) = \exp(\langle w, \mathbb{1}_S\rangle)/Z$$

- Note Ent is 1-homogeneous, so

$$\text{Ent}_\mu[f] = \text{Ent}_\mu[\exp(\langle w, \mathbb{1}_S\rangle)]/Z$$

- We get that

$$\langle \lambda, U_{1\to k} f\rangle = \sum_S \mu(s) f(s) \langle w, \mathbb{1}_S\rangle$$

- The Lagrangian simplifies to

$$-\log(Z) + \langle \lambda, g\rangle = k\langle w, q\rangle - \log(Z)$$

- If we reparameterize $z_i = e^{w_i}$, then this is

$$\log\left(\frac{\prod_{i\in[n]} z_i^{kq_i}}{g_\mu(z)}\right)$$

- This finishes the proof. 🙂

- For entropic independence we want $\frac{k}{C}\mathcal{D}_{\mathsf{KL}}(q \parallel \mu D_{k\to 1}) \leqslant$
  $$\sup\left\{ \log\left(\frac{\prod_{i\in[n]} z_i^{kq_i}}{g_\mu(z)} \right) \;\middle|\; z \in \mathbb{R}^n_{>0} \right\}$$

- For entropic independence we want $\frac{k}{C}\mathcal{D}_{\mathsf{KL}}(q \parallel \mu D_{k\to 1}) \leqslant$
  $$\sup\left\{\log\left(\frac{\prod_{i\in[n]} z_i^{kq_i}}{g_\mu(z)}\right) \;\middle|\; z \in \mathbb{R}^n_{>0}\right\}$$
- Let $h(z) = \mathbb{E}_{i\sim\mu D_{k\to 1}}\left[z_i^C\right]^{k/C}$. Then
  $$\sup\left\{\log\left(\frac{\prod_{i\in[n]} z_i^{kq_i}}{h(z)}\right) \;\middle|\; z \in \mathbb{R}^n_{>0}\right\}$$
  is achieved at $z_i = \sqrt[C]{q_i/(\mu D_{k\to 1})_i}$ and has value $\frac{k}{C}\mathcal{D}_{\mathsf{KL}}(q \parallel \mu D_{k\to 1})$.

- For entropic independence we want $\frac{k}{C}\mathcal{D}_{\mathsf{KL}}(q \parallel \mu D_{k\to 1}) \leqslant$
$$\sup\left\{\log\left(\frac{\prod_{i\in[n]} z_i^{kq_i}}{g_\mu(z)}\right) \;\middle|\; z \in \mathbb{R}^n_{>0}\right\}$$
- Let $h(z) = \mathbb{E}_{i\sim \mu D_{k\to 1}}\left[z_i^C\right]^{k/C}$. Then
$$\sup\left\{\log\left(\frac{\prod_{i\in[n]} z_i^{kq_i}}{h(z)}\right) \;\middle|\; z \in \mathbb{R}^n_{>0}\right\}$$
is achieved at $z_i = \sqrt[C]{q_i/(\mu D_{k\to 1})_i}$ and has value $\frac{k}{C}\mathcal{D}_{\mathsf{KL}}(q \parallel \mu D_{k\to 1})$.
- Thus C-EI is same as
$$g_\mu(z) \leqslant h(z)$$

- For entropic independence we want $\frac{k}{C}\mathcal{D}_{\mathsf{KL}}(q \parallel \mu D_{k\to 1}) \leqslant$
$$\sup\left\{\log\left(\frac{\prod_{i\in[n]} z_i^{kq_i}}{g_\mu(z)}\right) \,\middle|\, z\in\mathbb{R}^n_{>0}\right\}$$
- Let $h(z) = \mathbb{E}_{i\sim\mu D_{k\to 1}}\left[z_i^C\right]^{k/C}$. Then
$$\sup\left\{\log\left(\frac{\prod_{i\in[n]} z_i^{kq_i}}{h(z)}\right) \,\middle|\, z\in\mathbb{R}^n_{>0}\right\}$$
is achieved at $z_i = \sqrt[C]{q_i/(\mu D_{k\to 1})_i}$ and has value $\frac{k}{C}\mathcal{D}_{\mathsf{KL}}(q \parallel \mu D_{k\to 1})$.
- Thus C-EI is same as
$$g_\mu(z) \leqslant h(z)$$
- If $y_i = z_i^C$, this is the same as
$$g_\mu(\sqrt[C]{y_i})^{C/k} \leqslant \underbrace{\langle \mu D_{k\to 1}, y\rangle}_{\text{linear tangent at } \mathbb{1}}$$

- For entropic independence we want $\frac{k}{C}\mathcal{D}_{\mathrm{KL}}(q \parallel \mu D_{k\to 1}) \leqslant$
$$\sup\left\{\log\left(\frac{\prod_{i\in[n]} z_i^{kq_i}}{g_\mu(z)}\right) \,\middle|\, z \in \mathbb{R}^n_{>0}\right\}$$
- Let $h(z) = \mathbb{E}_{i\sim\mu D_{k\to 1}}\left[z_i^C\right]^{k/C}$. Then
$$\sup\left\{\log\left(\frac{\prod_{i\in[n]} z_i^{kq_i}}{h(z)}\right) \,\middle|\, z \in \mathbb{R}^n_{>0}\right\}$$
is achieved at $z_i = \sqrt[C]{q_i/(\mu D_{k\to 1})_i}$ and has value $\frac{k}{C}\mathcal{D}_{\mathrm{KL}}(q \parallel \mu D_{k\to 1})$.
- Thus C-EI is same as
$$g_\mu(z) \leqslant h(z)$$
- If $y_i = z_i^C$, this is the same as
$$g_\mu(\sqrt[C]{y_i})^{C/k} \leqslant \underbrace{\langle \mu D_{k\to 1}, y\rangle}_{\text{linear tangent at } \mathbb{1}}$$
- For $f(z) = g(\sqrt[C]{z})^{C/k}$, we have $\nabla f(\mathbb{1}) = \mu D_{k\to 1}$ and $\nabla^2 f(\mathbb{1}) \propto$
$$\mathrm{cov} - C\cdot\left(\mathrm{diag}(\mathrm{mean}) - \frac{\mathrm{mean}\,\mathrm{mean}^\intercal}{k}\right)$$

Folklore lemma

For a d-homogeneous function f, tfae:

1. $\{z \mid f(z) \geqslant 1\}$ convex
2. $\sqrt[d]{f}$ is concave
3. $\log f$ is concave

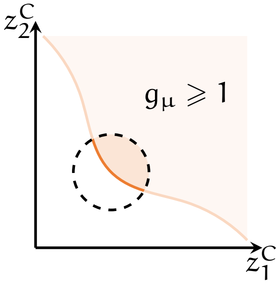
Similarly tfae:

1. $\{z \mid f(z) \geqslant 1\} \subseteq \{z \mid \langle \frac{\nabla f(\mathbb{1})}{d f(\mathbb{1})}, z\rangle \geqslant 1\}$
2. $\sqrt[d]{f}$ bounded by tangent at $\mathbb{1}$
3. $\log f$ bounded by tangent at $\mathbb{1}$

Polynomial view of HDX

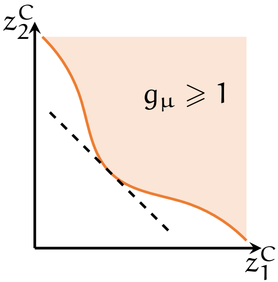
Spectral Independence

Level sets of $g_\mu(\sqrt[C]{z})$ locally convex at $\mathbb{1}$.



Entropic Independence

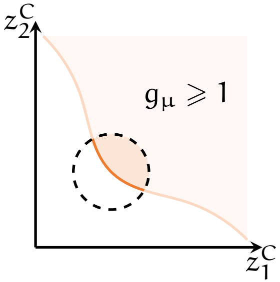
Level sets of $g_\mu(\sqrt[C]{z})$ bounded by tangent at $\mathbb{1}$.



Polynomial view of HDX

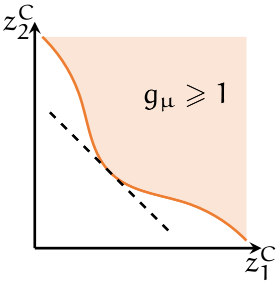
Spectral Independence

Level sets of $g_\mu(\sqrt[C]{z})$ locally convex at $\mathbb{1}$.



Entropic Independence

Level sets of $g_\mu(\sqrt[C]{z})$ bounded by tangent at $\mathbb{1}$.



Theorem [A-Jain-Koehler-Pham-Vuong'21]

C-spectral independence for all exp tilts $\Longrightarrow$ C-entropic independence.

- When $g_\mu(\sqrt[C]{z})$ is log-concave, we call it C fractionally log-concave.

- When $g_\mu(\sqrt[C]{z})$ is log-concave, we call it C fractionally log-concave.
- Matroids are $C = 1$ log-concave.

- When $g_\mu(\sqrt[C]{z})$ is log-concave, we call it C fractionally log-concave.
- Matroids are $C = 1$ log-concave.
- For $D_{k \to k-C} U_{k-C \to k}$ we have

$$t_{\text{mix}} = O\left(\binom{k}{C} \cdot \log \mathcal{D}_{\text{KL}}(\nu_0 \parallel \mu)\right)$$

- When $g_\mu(\sqrt[C]{z})$ is log-concave, we call it C fractionally log-concave.
- Matroids are $C = 1$ log-concave.
- For $D_{k\to k-C}U_{k-C\to k}$ we have

$$t_{\mathsf{mix}} = O\left(\binom{k}{C} \cdot \log \mathcal{D}_{\mathsf{KL}}(\nu_0 \parallel \mu)\right)$$

- For matroids, this was proved before EI by [Cryan-Guo-Mousa].

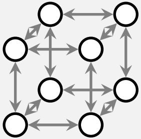
- When $g_\mu(\sqrt[C]{z})$ is log-concave, we call it C fractionally log-concave.
- Matroids are $C = 1$ log-concave.
- For $D_{k\to k-C} U_{k-C\to k}$ we have

$$t_{\mathsf{mix}} = O\left(\binom{k}{C} \cdot \log \mathcal{D}_{\mathsf{KL}}(\nu_0 \parallel \mu)\right)$$

- For matroids, this was proved before EI by [Cryan-Guo-Mousa].

Example: hypercube

- $\{0,1\}^n \hookrightarrow \binom{[2n]}{n}$
- Glauber becomes $D_{n\to n-1}$



- $t_{\mathsf{mix}} = O(n \log n)$

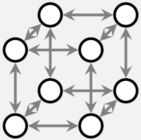
- When $g_\mu(\sqrt[C]{z})$ is log-concave, we call it C fractionally log-concave.
- Matroids are $C = 1$ log-concave.
- For $D_{k \to k-C} U_{k-C \to k}$ we have

$$t_{\text{mix}} = O\left(\binom{k}{C} \cdot \log \mathcal{D}_{\text{KL}}(\nu_0 \parallel \mu)\right)$$

- For matroids, this was proved before EI by [Cryan-Guo-Mousa].

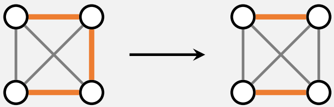
Example: hypercube

- $\{0,1\}^n \hookrightarrow \binom{[2n]}{n}$
- Glauber becomes $D_{n \to n-1}$



- $t_{\text{mix}} = O(n \log n)$

Example: spanning trees (I)



- P: drop edge u.a.r., then add
- $t_{\text{mix}} = O(n \log n)$

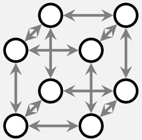
- When $g_\mu(\sqrt[C]{z})$ is log-concave, we call it C fractionally log-concave.
- Matroids are $C = 1$ log-concave.
- For $D_{k\to k-C} U_{k-C\to k}$ we have

$$t_{mix} = O\left(\binom{k}{C} \cdot \log \mathcal{D}_{KL}(\nu_0 \parallel \mu)\right)$$

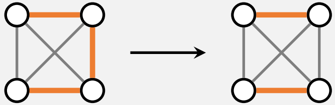
- For matroids, this was proved before EI by [Cryan-Guo-Mousa].

Example: hypercube

- $\{0,1\}^n \hookrightarrow \binom{[2n]}{n}$
- Glauber becomes $D_{n\to n-1}$
- $t_{mix} = O(n \log n)$

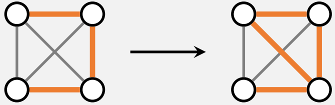


Example: spanning trees (I)



- P: drop edge u.a.r., then add
- $t_{mix} = O(n \log n)$

Example: spanning trees (II)



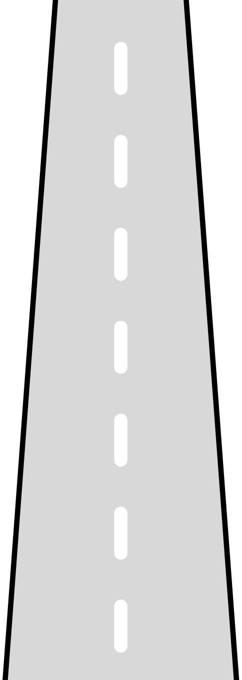
- P: add edge u.a.r., then drop
- $t_{mix} = O(m \log n)$

Entropic Independence

- Fractional log-concavity

Polynomial Interpolation

- Matching polynomial
- Taylor approximation
- Riemann mappings

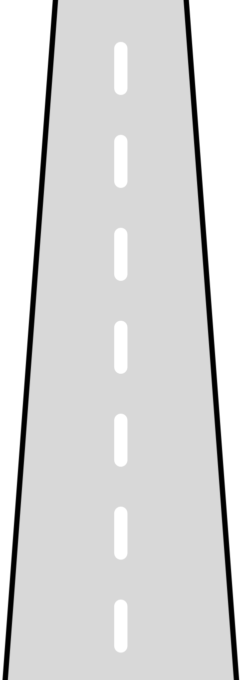


Entropic Independence

- Fractional log-concavity

Polynomial Interpolation

- Matching polynomial
- Taylor approximation
- Riemann mappings



Polynomial interpolation

- We will now focus on a deterministic counting method invented by [Barvinok].

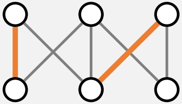
Polynomial interpolation

- We will now focus on a deterministic counting method invented by [Barvinok].

Running example: matchings

Goal: matchings.

↑ not necessarily perfect



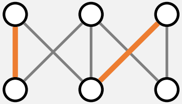
Polynomial interpolation

- We will now focus on a deterministic counting method invented by [Barvinok].

Running example: matchings

Goal: matchings.

↑ not necessarily perfect



- There is FPRAS [Jerrum-Sinclair].

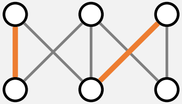
Polynomial interpolation

- We will now focus on a deterministic counting method invented by [Barvinok].

Running example: matchings

Goal: matchings.

↑ not necessarily perfect



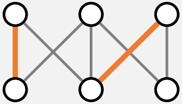
- There is FPRAS [Jerrum-Sinclair].
- Open: design FPTAS

Polynomial interpolation

- We will now focus on a deterministic counting method invented by [Barvinok].

Running example: matchings

Goal: matchings.
↑
not necessarily perfect



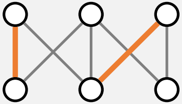
- There is FPRAS [Jerrum-Sinclair].
- Open: design FPTAS
- For $\Delta = O(1)$-bounded degree graphs, we know FPTAS [Bayati-Gamarnik-Katz-Nair-Tetali]

Polynomial interpolation

- We will now focus on a deterministic counting method invented by [Barvinok].

Running example: matchings

Goal: matchings.
↑
not necessarily perfect



- There is FPRAS [Jerrum-Sinclair].
- Open: design FPTAS
- For $\Delta = O(1)$-bounded degree graphs, we know FPTAS [Bayati-Gamarnik-Katz-Nair-Tetali]

- With $m_k = \#(k\text{-matchings in } G)$, matching polynomial is:

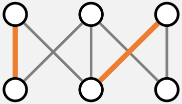
$$p_G(z) = m_0 + m_1 z + m_2 z^2 + \ldots$$

Polynomial interpolation

- We will now focus on a deterministic counting method invented by [Barvinok].

Running example: matchings

Goal: matchings.
↑
not necessarily perfect



- There is FPRAS [Jerrum-Sinclair].
- Open: design FPTAS
- For $\Delta = O(1)$-bounded degree graphs, we know FPTAS [Bayati-Gamarnik-Katz-Nair-Tetali]

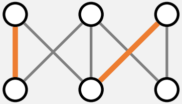
- With $m_k = \#(k\text{-matchings in } G)$, matching polynomial is:
$$p_G(z) = m_0 + m_1 z + m_2 z^2 + \dots$$
- Goal: approximate $p_G(1)$

Polynomial interpolation

- We will now focus on a deterministic counting method invented by [Barvinok].

Running example: matchings

Goal: matchings.
↑
not necessarily perfect



- There is FPRAS [Jerrum-Sinclair].
- Open: design FPTAS
- For $\Delta = O(1)$-bounded degree graphs, we know FPTAS [Bayati-Gamarnik-Katz-Nair-Tetali]

- With $m_k = \#(k\text{-matchings in } G)$, matching polynomial is:

$$p_G(z) = m_0 + m_1 z + m_2 z^2 + \ldots$$

- Goal: approximate $p_G(1)$
- Note that m_k, and thus $p_G^{(k)}(0)$ can be computed in $n^{O(k)}$ time.

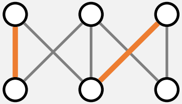
kth derivative

Polynomial interpolation

- We will now focus on a deterministic counting method invented by [Barvinok].

Running example: matchings

Goal: matchings.
↑
not necessarily perfect



- There is FPRAS [Jerrum-Sinclair].
- Open: design FPTAS
- For $\Delta = O(1)$-bounded degree graphs, we know FPTAS [Bayati-Gamarnik-Katz-Nair-Tetali]

- With $m_k = \#(k\text{-matchings in } G)$, matching polynomial is:
$$p_G(z) = m_0 + m_1 z + m_2 z^2 + \ldots$$
- Goal: approximate $p_G(1)$
- Note that m_k, and thus $p_G^{(k)}(0)$ can be computed in $n^{O(k)}$ time.

kth derivative

Polynomial interpolation

For $\Delta = O(1)$, we can multiplicatively approximate $p_G(1)$ using
$$p_G^{(0)}(0), \ldots, p_G^{O(\log n)}(0)$$

- Naïvely this gives $n^{O(\log n)}$.

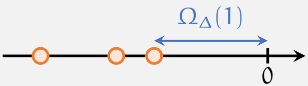
- Naïvely this gives $n^{O(\log n)}$.
- Trick of [Patel-Regts] computes $m_0, \ldots, m_k$ in time $\Delta^{O(k)}\text{poly}(n)$.

- Naïvely this gives $n^{O(\log n)}$.
- Trick of [Patel-Regts] computes $m_0, \ldots, m_k$ in time $\Delta^{O(k)}\text{poly}(n)$.
- What's special about matchings?

- Naïvely this gives $n^{O(\log n)}$.
- Trick of [Patel-Regts] computes $m_0, \ldots, m_k$ in time $\Delta^{O(k)}\mathrm{poly}(n)$.
- What's special about matchings?

Theorem [Heilmann-Lieb'72]

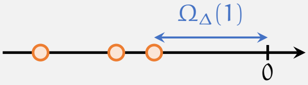
Zeros/roots of p_G are real. In fact they lie in $(-\infty, -\Omega_\Delta(1)]$.



- Naïvely this gives $n^{O(\log n)}$.
- Trick of [Patel-Regts] computes $m_0, \ldots, m_k$ in time $\Delta^{O(k)} \mathrm{poly}(n)$.
- What's special about matchings?

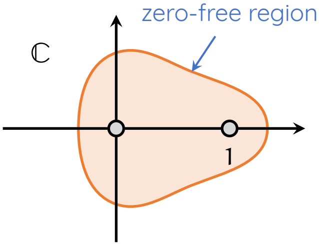
Theorem [Heilmann-Lieb'72]

Zeros/roots of p_G are real. In fact they lie in $(-\infty, -\Omega_\Delta(1)]$.



- Roots are singularities of $\log p_G$. Derivatives of $\log p_G$ are macroscopic observables. Physics: phase transitions happen at roots.

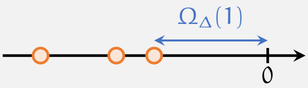
- General setting for [Barvinok]:



- Naïvely this gives $n^{O(\log n)}$.
- Trick of [Patel-Regts] computes $m_0, \ldots, m_k$ in time $\Delta^{O(k)}\mathrm{poly}(n)$.
- What's special about matchings?

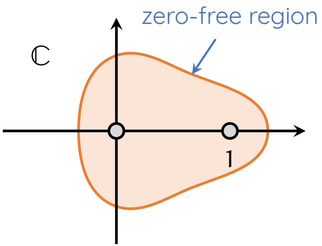
Theorem [Heilmann-Lieb'72]

Zeros/roots of p_G are real. In fact they lie in $(-\infty, -\Omega_\Delta(1)]$.



- Roots are singularities of $\log p_G$. Derivatives of $\log p_G$ are macroscopic observables. Physics: phase transitions happen at roots.

- General setting for [Barvinok]:



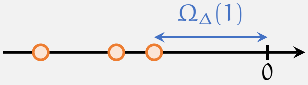
- "Fat" simply connected zero-free region around $0, 1$.

can be any two points

- Naïvely this gives $n^{O(\log n)}$.
- Trick of [Patel-Regts] computes $m_0, \ldots, m_k$ in time $\Delta^{O(k)}\text{poly}(n)$.
- What's special about matchings?

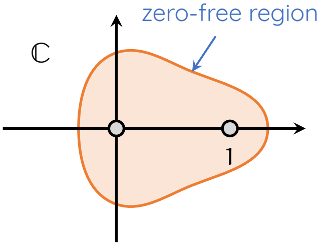
Theorem [Heilmann-Lieb'72]

Zeros/roots of p_G are real. In fact they lie in $(-\infty, -\Omega_\Delta(1)]$.



- Roots are singularities of $\log p_G$. Derivatives of $\log p_G$ are macroscopic observables. Physics: phase transitions happen at roots.

- General setting for [Barvinok]:



- "Fat" simply connected zero-free region around $0, 1$.

  can be any two points

- Approximate $p(1)$ using low-order derivatives of p at 0. 🙂

- Idea: truncate Taylor of $\log p$:

$$\log p(z) = a_0 + a_1 z + \dots$$

where $k! \cdot a_k = \frac{d^k}{dz^k} \log p(0)$ is a function of $p^{(0)}(0), \dots, p^{(k)}(0)$.

↑ by calculus rules

- Idea: truncate Taylor of $\log p$:

$$\log p(z) = a_0 + a_1 z + \dots$$

where $k! \cdot a_k = \frac{d^k}{dz^k} \log p(0)$ is a function of $p^{(0)}(0), \dots, p^{(k)}(0)$.

by calculus rules

- Complex analysis fact: Taylor series convergence radius is distance to nearest singularity.

zero of p

- Idea: truncate Taylor of $\log p$:
$$\log p(z) = a_0 + a_1 z + \dots$$
where $k! \cdot a_k = \frac{d^k}{dz^k} \log p(0)$ is a function of $p^{(0)}(0), \dots, p^{(k)}(0)$.

by calculus rules

- Complex analysis fact: Taylor series convergence radius is distance to nearest singularity.

zero of p

- When converging, there is hope truncation has low error.

- Idea: truncate Taylor of $\log p$:

$$\log p(z) = a_0 + a_1 z + \ldots$$

where $k! \cdot a_k = \frac{d^k}{dz^k} \log p(0)$ is a function of $p^{(0)}(0), \ldots, p^{(k)}(0)$.

by calculus rules

- Complex analysis fact: Taylor series convergence radius is distance to nearest singularity.

zero of p

- When converging, there is hope truncation has low error.
- This can only work for disks. Will generalize to other regions later.

- Idea: truncate Taylor of $\log p$:

$$\log p(z) = a_0 + a_1 z + \dots$$

where $k! \cdot a_k = \frac{d^k}{dz^k} \log p(0)$ is a function of $p^{(0)}(0), \dots, p^{(k)}(0)$.

by calculus rules

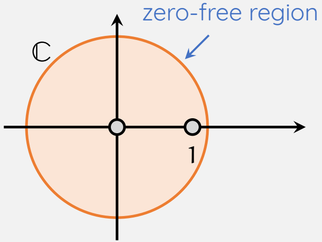
- Complex analysis fact: Taylor series convergence radius is distance to nearest singularity.

zero of p

- When converging, there is hope truncation has low error.
- This can only work for disks. Will generalize to other regions later.

Polynomial interpolation for disks

Suppose $p(z) \neq 0$ whenever $|z| \leqslant 1 + \delta$:



then k-trunc of Taylor for $\log p(1)$ has additive error $\leqslant \left(2e^{-\delta k}/k\delta\right) \cdot \deg(p)$.

- Idea: truncate Taylor of $\log p$:

$$\log p(z) = a_0 + a_1 z + \dots$$

where $k! \cdot a_k = \frac{d^k}{dz^k} \log p(0)$ is a function of $p^{(0)}(0), \dots, p^{(k)}(0)$.

by calculus rules

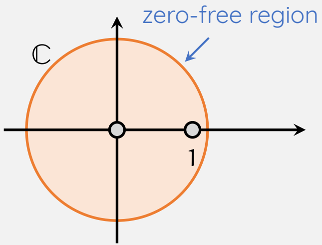
- Complex analysis fact: Taylor series convergence radius is distance to nearest singularity.

zero of p

- When converging, there is hope truncation has low error.
- This can only work for disks. Will generalize to other regions later.

Polynomial interpolation for disks

Suppose $p(z) \neq 0$ whenever $|z| \leqslant 1+\delta$:



then k-trunc of Taylor for $\log p(1)$ has additive error $\leqslant \left(2e^{-\delta k}/k\delta\right) \cdot \deg(p)$.

For $1+\epsilon$ approx of $p(1)$, set

$$k = O\left(\frac{\log(\deg(p)/\epsilon)}{\delta}\right)$$

Proof:

- Since p is polynomial we can write
$$p(z) = c\left(1 - \frac{z}{\lambda_1}\right)\cdots\left(1 - \frac{z}{\lambda_n}\right)$$

Proof:

- Since p is polynomial we can write

$$p(z) = c\left(1 - \frac{z}{\lambda_1}\right) \cdots \left(1 - \frac{z}{\lambda_n}\right)$$

- Or in logarithms

$$\log p(z) = \log(c) + \sum_i \log\left(1 - \frac{z}{\lambda_i}\right)$$

Proof:

- Since p is polynomial we can write
$$p(z) = c\left(1 - \frac{z}{\lambda_1}\right)\cdots\left(1 - \frac{z}{\lambda_n}\right)$$
- Or in logarithms
$$\log p(z) = \log(c) + \sum_i \log\left(1 - \frac{z}{\lambda_i}\right)$$
- Which means $\mathrm{Taylor}_k(\log p) =$
$$\log(c) + \sum_i \mathrm{Taylor}_k\left(1 - \frac{z}{\lambda_i}\right)$$

Proof:

- Since p is polynomial we can write
$$p(z) = c\left(1 - \frac{z}{\lambda_1}\right) \cdots \left(1 - \frac{z}{\lambda_n}\right)$$
- Or in logarithms
$$\log p(z) = \log(c) + \sum_i \log\left(1 - \frac{z}{\lambda_i}\right)$$
- Which means $\mathrm{Taylor}_k(\log p) =$
$$\log(c) + \sum_i \mathrm{Taylor}_k\left(1 - \frac{z}{\lambda_i}\right)$$
- Enough to bound error of each $\mathrm{Taylor}_k\left(1 - \frac{z}{\lambda_i}\right)$ and multiply by n.

Proof:

- Since p is polynomial we can write
$$p(z) = c\left(1 - \frac{z}{\lambda_1}\right) \cdots \left(1 - \frac{z}{\lambda_n}\right)$$
- Or in logarithms
$$\log p(z) = \log(c) + \sum_i \log\left(1 - \frac{z}{\lambda_i}\right)$$
- Which means $\mathsf{Taylor}_k(\log p) =$
$$\log(c) + \sum_i \mathsf{Taylor}_k\left(1 - \frac{z}{\lambda_i}\right)$$
- Enough to bound error of each $\mathsf{Taylor}_k\left(1 - \frac{z}{\lambda_i}\right)$ and multiply by n.
- Taylor series of $\log(1 - x)$ is
$$-x - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} - \ldots$$

Proof:

- Since p is polynomial we can write
$$p(z) = c\left(1 - \frac{z}{\lambda_1}\right) \cdots \left(1 - \frac{z}{\lambda_n}\right)$$
- Or in logarithms
$$\log p(z) = \log(c) + \sum_i \log\left(1 - \frac{z}{\lambda_i}\right)$$
- Which means $\text{Taylor}_k(\log p) =$
$$\log(c) + \sum_i \text{Taylor}_k\left(1 - \frac{z}{\lambda_i}\right)$$
- Enough to bound error of each $\text{Taylor}_k\left(1 - \frac{z}{\lambda_i}\right)$ and multiply by n.
- Taylor series of $\log(1 - x)$ is
$$-x - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} - \dots$$
- Error of Taylor_k is
$$\leqslant \sum_{i>k} \frac{|x|^i}{i} \leqslant \sum_{i>k} \frac{|x|^i}{k} = \frac{|x|^k}{k(1-|x|)}$$

Proof:

- Since p is polynomial we can write
$$p(z) = c\left(1 - \frac{z}{\lambda_1}\right) \cdots \left(1 - \frac{z}{\lambda_n}\right)$$
- Or in logarithms
$$\log p(z) = \log(c) + \sum_i \log\left(1 - \frac{z}{\lambda_i}\right)$$
- Which means $\text{Taylor}_k(\log p) =$
$$\log(c) + \sum_i \text{Taylor}_k\left(1 - \frac{z}{\lambda_i}\right)$$
- Enough to bound error of each $\text{Taylor}_k\left(1 - \frac{z}{\lambda_i}\right)$ and multiply by n.
- Taylor series of $\log(1 - x)$ is
$$-x - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} - \dots$$
- Error of Taylor_k is
$$\leqslant \sum_{i>k} \frac{|x|^i}{i} \leqslant \sum_{i>k} \frac{|x|^i}{k} = \frac{|x|^k}{k(1-|x|)}$$
- For $x = 1/\lambda_i$, we have $|x| \leqslant e^{-\delta}$, so error of each term is
$$\leqslant \frac{e^{-\delta k}}{k(1-e^{-\delta})} \leqslant \frac{2e^{-\delta k}}{k\delta}$$

Proof:

- Since p is polynomial we can write
  $$p(z) = c\left(1 - \frac{z}{\lambda_1}\right) \cdots \left(1 - \frac{z}{\lambda_n}\right)$$
- Or in logarithms
  $$\log p(z) = \log(c) + \sum_i \log\left(1 - \frac{z}{\lambda_i}\right)$$
- Which means $\text{Taylor}_k(\log p) =$
  $$\log(c) + \sum_i \text{Taylor}_k\left(1 - \frac{z}{\lambda_i}\right)$$
- Enough to bound error of each $\text{Taylor}_k\left(1 - \frac{z}{\lambda_i}\right)$ and multiply by n.
- Taylor series of $\log(1 - x)$ is
  $$-x - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} - \dots$$
- Error of Taylor_k is
  $$\leqslant \sum_{i>k} \frac{|x|^i}{i} \leqslant \sum_{i>k} \frac{|x|^i}{k} = \frac{|x|^k}{k(1-|x|)}$$
- For $x = 1/\lambda_i$, we have $|x| \leqslant e^{-\delta}$, so error of each term is
  $$\leqslant \frac{e^{-\delta k}}{k(1-e^{-\delta})} \leqslant \frac{2e^{-\delta k}}{k\delta}$$
- Since there are $n = \deg(p)$ terms, overall error is $\leqslant n \cdot \frac{2e^{-\delta k}}{k\delta}$.

Proof:

- Since p is polynomial we can write
$$p(z) = c\left(1 - \frac{z}{\lambda_1}\right) \cdots \left(1 - \frac{z}{\lambda_n}\right)$$
- Or in logarithms
$$\log p(z) = \log(c) + \sum_i \log\left(1 - \frac{z}{\lambda_i}\right)$$
- Which means $\text{Taylor}_k(\log p) =$
$$\log(c) + \sum_i \text{Taylor}_k\left(1 - \frac{z}{\lambda_i}\right)$$
- Enough to bound error of each $\text{Taylor}_k\left(1 - \frac{z}{\lambda_i}\right)$ and multiply by n.
- Taylor series of $\log(1 - x)$ is
$$-x - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} - \dots$$
- Error of Taylor_k is
$$\leqslant \sum_{i>k} \frac{|x|^i}{i} \leqslant \sum_{i>k} \frac{|x|^i}{k} = \frac{|x|^k}{k(1-|x|)}$$
- For $x = 1/\lambda_i$, we have $|x| \leqslant e^{-\delta}$, so error of each term is
$$\leqslant \frac{e^{-\delta k}}{k(1-e^{-\delta})} \leqslant \frac{2e^{-\delta k}}{k\delta}$$
- Since there are $n = \deg(p)$ terms, overall error is $\leqslant n \cdot \frac{2e^{-\delta k}}{k\delta}$.
- Note $k = O(\log(n/\epsilon)/\delta)$ makes overall error $\simeq \epsilon$, which means a $1 + O(\epsilon)$ mult approx of $p(1)$.

Proof:

- Since p is polynomial we can write
$$p(z) = c\left(1 - \frac{z}{\lambda_1}\right) \cdots \left(1 - \frac{z}{\lambda_n}\right)$$
- Or in logarithms
$$\log p(z) = \log(c) + \sum_i \log\left(1 - \frac{z}{\lambda_i}\right)$$
- Which means $\text{Taylor}_k(\log p) =$
$$\log(c) + \sum_i \text{Taylor}_k\left(1 - \frac{z}{\lambda_i}\right)$$
- Enough to bound error of each $\text{Taylor}_k\left(1 - \frac{z}{\lambda_i}\right)$ and multiply by n.
- Taylor series of $\log(1-x)$ is
$$-x - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} - \ldots$$

- Error of Taylor_k is
$$\leqslant \sum_{i>k} \frac{|x|^i}{i} \leqslant \sum_{i>k} \frac{|x|^i}{k} = \frac{|x|^k}{k(1-|x|)}$$
- For $x = 1/\lambda_i$, we have $|x| \leqslant e^{-\delta}$, so error of each term is
$$\leqslant \frac{e^{-\delta k}}{k(1-e^{-\delta})} \leqslant \frac{2e^{-\delta k}}{k\delta}$$
- Since there are $n = \deg(p)$ terms, overall error is $\leqslant n \cdot \frac{2e^{-\delta k}}{k\delta}$.
- Note $k = O(\log(n/\epsilon)/\delta)$ makes overall error $\simeq \epsilon$, which means a $1 + O(\epsilon)$ mult approx of $p(1)$.
- If Taylor_k takes time $n^{O(k)}$, overall runtime is $n^{O(\log(n/\epsilon))}$.

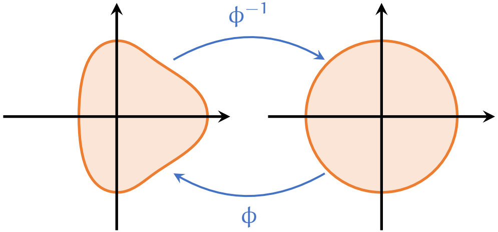
Proof:

- Since p is polynomial we can write
  $$p(z) = c\left(1 - \frac{z}{\lambda_1}\right) \cdots \left(1 - \frac{z}{\lambda_n}\right)$$
- Or in logarithms
  $$\log p(z) = \log(c) + \sum_i \log\left(1 - \frac{z}{\lambda_i}\right)$$
- Which means $\text{Taylor}_k(\log p) =$
  $$\log(c) + \sum_i \text{Taylor}_k\left(1 - \frac{z}{\lambda_i}\right)$$
- Enough to bound error of each $\text{Taylor}_k\left(1 - \frac{z}{\lambda_i}\right)$ and multiply by n.
- Taylor series of $\log(1-x)$ is
  $$-x - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} - \cdots$$
- Error of Taylor_k is
  $$\leqslant \sum_{i>k} \frac{|x|^i}{i} \leqslant \sum_{i>k} \frac{|x|^i}{k} = \frac{|x|^k}{k(1-|x|)}$$
- For $x = 1/\lambda_i$, we have $|x| \leqslant e^{-\delta}$, so error of each term is
  $$\leqslant \frac{e^{-\delta k}}{k(1-e^{-\delta})} \leqslant \frac{2e^{-\delta k}}{k\delta}$$
- Since there are $n = \deg(p)$ terms, overall error is $\leqslant n \cdot \frac{2e^{-\delta k}}{k\delta}$.
- Note $k = O(\log(n/\epsilon)/\delta)$ makes overall error $\simeq \epsilon$, which means a $1 + O(\epsilon)$ mult approx of $p(1)$.
- If Taylor_k takes time $n^{O(k)}$, overall runtime is $n^{O(\log(n/\epsilon))}$.
- With [Patel-Regts], runtime is $O(1)^{O(\log(n/\epsilon))} = \text{poly}(n, 1/\epsilon)$. 🙂

How to extend beyond disks?

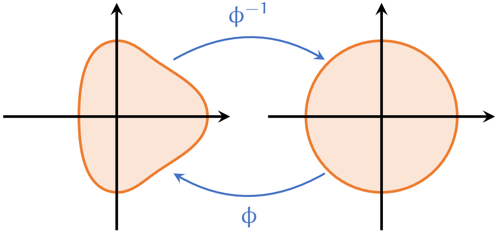
Riemann mapping

There is a biholomorphic map between any two simply connected regions in $\mathbb{C}$:



Riemann mapping

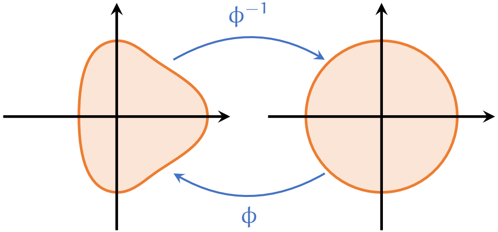
There is a biholomorphic map between any two simply connected regions in $\mathbb{C}$:



- We can also map one interior point to one interior point.

Riemann mapping

There is a biholomorphic map between any two simply connected regions in $\mathbb{C}$:



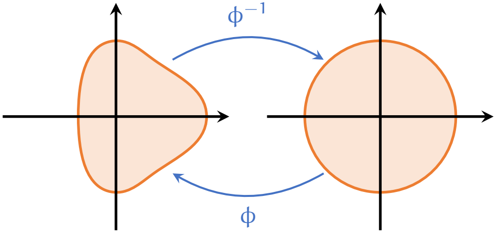
- We can also map one interior point to one interior point.
- Approximating ϕ, we can construct polynomial ψ such that

$$\psi(0) = 0, \psi(1) = 1$$

$$\psi(\text{disk}) \subseteq \text{region}$$

Riemann mapping

There is a biholomorphic map between any two simply connected regions in $\mathbb{C}$:



- We can also map one interior point to one interior point.
- Approximating ϕ, we can construct polynomial ψ such that

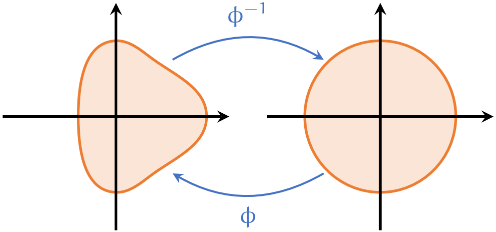
$$\psi(0) = 0, \psi(1) = 1$$

$$\psi(\text{disk}) \subseteq \text{region}$$

- Apply disk [Barvinok] to $p \circ \psi$. ☺

Riemann mapping

There is a biholomorphic map between any two simply connected regions in $\mathbb{C}$:



- We can also map one interior point to one interior point.
- Approximating ϕ, we can construct polynomial ψ such that

$$\psi(0) = 0, \psi(1) = 1$$

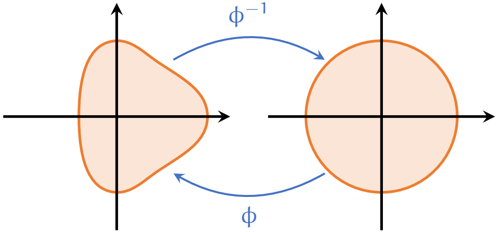
$$\psi(\text{disk}) \subseteq \text{region}$$

- Apply disk [Barvinok] to $p \circ \psi$. ☺
- Read first k derivatives from

$$p^{(0)}(0), \ldots, p^{(k)}(0).$$

Riemann mapping

There is a biholomorphic map between any two simply connected regions in $\mathbb{C}$:



- We can also map one interior point to one interior point.
- Approximating ϕ, we can construct polynomial ψ such that

$$\psi(0) = 0, \psi(1) = 1$$
$$\psi(\text{disk}) \subseteq \text{region}$$

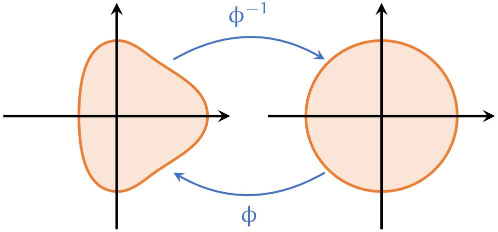
- Apply disk [Barvinok] to $p \circ \psi$. 🙂
- Read first k derivatives from

$$p^{(0)}(0), \ldots, p^{(k)}(0).$$

- Fine when $\deg(\psi)$ reasonable.

Riemann mapping

There is a biholomorphic map between any two simply connected regions in $\mathbb{C}$:



- We can also map one interior point to one interior point.
- Approximating ϕ, we can construct polynomial ψ such that

$$\psi(0) = 0, \psi(1) = 1$$
$$\psi(\text{disk}) \subseteq \text{region}$$

- Apply disk [Barvinok] to $p \circ \psi$. 🙂
- Read first k derivatives from

$$p^{(0)}(0), \ldots, p^{(k)}(0).$$

- Fine when $\deg(\psi)$ reasonable.

Example: matching polynomial

- Region is $\mathbb{C} - \mathbb{R}_{\leqslant -r}$ for some r.
- Start with Möbius map

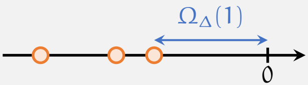
$$\phi(z) = (az + b)/(cz + d)$$

- Set a, b, c, d to ensure $\phi(0) = 0, \phi(1) = 1, \phi(\text{disk}) \cap \mathbb{R}_{\leqslant -r/2} = \emptyset$.
- Exercise: Taylor approx ϕ and compose with linear fn to get ψ.

to ensure $\psi(0) = 0, \psi(1) = 1$

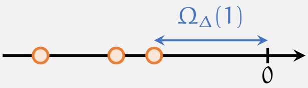
Theorem [Heilmann-Lieb'72]

Zeros/roots of p_G are real. In fact they lie in $(-\infty, -\Omega_\Delta(1)]$.



Theorem [Heilmann-Lieb'72]

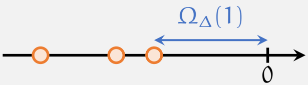
Zeros/roots of p_G are real. In fact they lie in $(-\infty, -\Omega_\Delta(1)]$.



Proof:

Theorem [Heilmann-Lieb'72]

Zeros/roots of p_G are real. In fact they lie in $(-\infty, -\Omega_\Delta(1)]$.



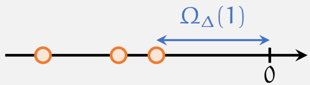
Proof:

- Idea: induction. Let u be vertex:

$$p_G = p_{G-u} + z \sum_{v \sim u} p_{G-u-v}$$

Theorem [Heilmann-Lieb'72]

Zeros/roots of p_G are real. In fact they lie in $(-\infty, -\Omega_\Delta(1)]$.

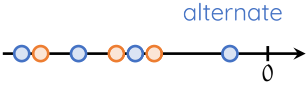


Proof:

- Idea: induction. Let u be vertex:

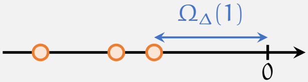
$$p_G = p_{G-u} + z\sum_{v\sim u} p_{G-u-v}$$

- Inductive claim: roots of p_G and p_{G-u} are real and interlace:



Theorem [Heilmann-Lieb'72]

Zeros/roots of p_G are real. In fact they lie in $(-\infty, -\Omega_\Delta(1)]$.



Proof:

- Idea: induction. Let u be vertex:

$$p_G = p_{G-u} + z \sum_{v \sim u} p_{G-u-v}$$

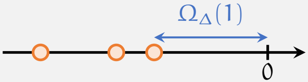
- Inductive claim: roots of p_G and p_{G-u} are real and interlace:

alternate

- Apply induction to p_{G-u} and p_{G-u-v}. Signs at roots of p_{G-u}:

Theorem [Heilmann-Lieb'72]

Zeros/roots of p_G are real. In fact they lie in $(-\infty, -\Omega_\Delta(1)]$.



Proof:

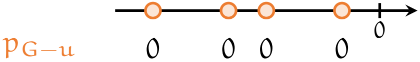
- Idea: induction. Let u be vertex:

$$p_G = p_{G-u} + z \sum_{v \sim u} p_{G-u-v}$$

- Inductive claim: roots of p_G and p_{G-u} are real and interlace:

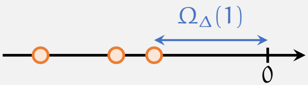
alternate

- Apply induction to p_{G-u} and p_{G-u-v}. Signs at roots of p_{G-u}:



Theorem [Heilmann-Lieb'72]

Zeros/roots of p_G are real. In fact they lie in $(-\infty, -\Omega_\Delta(1)]$.



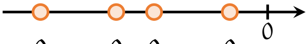
Proof:

- Idea: induction. Let u be vertex:

$$p_G = p_{G-u} + z\sum_{v\sim u} p_{G-u-v}$$

- Inductive claim: roots of p_G and p_{G-u} are real and interlace:
alternate

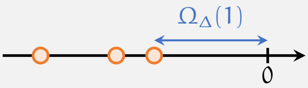
- Apply induction to p_{G-u} and p_{G-u-v}. Signs at roots of p_{G-u}:



p_{G-u}	0	0	0	0
p_{G-u-v_1}	$-$	$+$	$-$	$+$

Theorem [Heilmann-Lieb'72]

Zeros/roots of p_G are real. In fact they lie in $(-\infty, -\Omega_\Delta(1)]$.



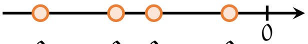
Proof:

- Idea: induction. Let u be vertex:

$$p_G = p_{G-u} + z \sum_{v\sim u} p_{G-u-v}$$

- Inductive claim: roots of p_G and p_{G-u} are real and interlace:
  alternate

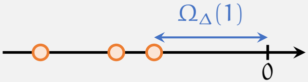
- Apply induction to p_{G-u} and p_{G-u-v}. Signs at roots of p_{G-u}:



p_{G-u}	0	0	0	0
p_{G-u-v_1}	$-$	$+$	$-$	$+$
p_{G-u-v_2}	$-$	$+$	$-$	$+$

Theorem [Heilmann-Lieb'72]

Zeros/roots of p_G are real. In fact they lie in $(-\infty, -\Omega_\Delta(1)]$.



Proof:

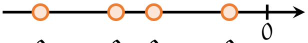
- Idea: induction. Let u be vertex:

$$p_G = p_{G-u} + z \sum_{v \sim u} p_{G-u-v}$$

- Inductive claim: roots of p_G and p_{G-u} are real and interlace:

alternate

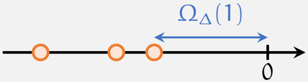
- Apply induction to p_{G-u} and p_{G-u-v}. Signs at roots of p_{G-u}:



p_{G-u}	0	0	0	0
p_{G-u-v_1}	$-$	$+$	$-$	$+$
p_{G-u-v_2}	$-$	$+$	$-$	$+$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
p_G	$+$	$-$	$+$	$-$

Theorem [Heilmann-Lieb'72]

Zeros/roots of p_G are real. In fact they lie in $(-\infty, -\Omega_\Delta(1)]$.



Proof:

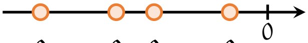
- Idea: induction. Let u be vertex:

$$p_G = p_{G-u} + z\sum_{v\sim u} p_{G-u-v}$$

- Inductive claim: roots of p_G and p_{G-u} are real and interlace:

alternate

- Apply induction to p_{G-u} and p_{G-u-v}. Signs at roots of p_{G-u}:

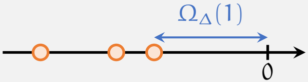


p_{G-u}	0	0	0	0
p_{G-u-v_1}	$-$	$+$	$-$	$+$
p_{G-u-v_2}	$-$	$+$	$-$	$+$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
p_G	$+$	$-$	$+$	$-$

- By sign alts, we get interlacing of roots for p_G and p_{G-u}. ☺

Theorem [Heilmann-Lieb'72]

Zeros/roots of p_G are real. In fact they lie in $(-\infty, -\Omega_\Delta(1)]$.



Proof:

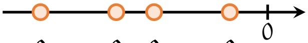
- Idea: induction. Let u be vertex:

$$p_G = p_{G-u} + z \sum_{v \sim u} p_{G-u-v}$$

- Inductive claim: roots of p_G and p_{G-u} are real and interlace:

alternate

- Apply induction to p_{G-u} and p_{G-u-v}. Signs at roots of p_{G-u}:



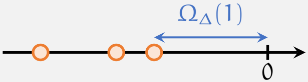
p_{G-u}	0	0	0	0
p_{G-u-v_1}	$-$	$+$	$-$	$+$
p_{G-u-v_2}	$-$	$+$	$-$	$+$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
p_G	$+$	$-$	$+$	$-$

- By sign alts, we get interlacing of roots for p_G and p_{G-u}. ☺
- Next prove for $z \in (-\frac{1}{4\Delta}, 0]$:

$$2p_G(z) > p_{G-u}(z) > 0.$$

Theorem [Heilmann-Lieb'72]

Zeros/roots of p_G are real. In fact they lie in $(-\infty, -\Omega_\Delta(1)]$.



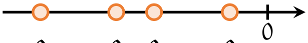
Proof:

- Idea: induction. Let u be vertex:

$$p_G = p_{G-u} + z\sum_{v\sim u} p_{G-u-v}$$

- Inductive claim: roots of p_G and p_{G-u} are real and interlace:

- Apply induction to p_{G-u} and p_{G-u-v}. Signs at roots of p_{G-u}:



p_{G-u}	0	0	0	0
p_{G-u-v_1}	$-$	$+$	$-$	$+$
p_{G-u-v_2}	$-$	$+$	$-$	$+$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
p_G	$+$	$-$	$+$	$-$

- By sign alts, we get interlacing of roots for p_G and p_{G-u}. ☺
- Next prove for $z \in (-\frac{1}{4\Delta}, 0]$:

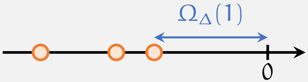
$$2p_G(z) > p_{G-u}(z) > 0.$$

- Induction step:

$$p_G(z) \geqslant (1+2\Delta z)p_{G-u} \geqslant \tfrac{1}{2}p_{G-u}(z)$$

Theorem [Heilmann-Lieb'72]

Zeros/roots of p_G are real. In fact they lie in $(-\infty, -\Omega_\Delta(1)]$.



Proof:

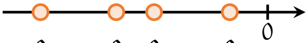
- Idea: induction. Let u be vertex:

$$p_G = p_{G-u} + z \sum_{v \sim u} p_{G-u-v}$$

- Inductive claim: roots of p_G and p_{G-u} are real and interlace:

alternate

- Apply induction to p_{G-u} and p_{G-u-v}. Signs at roots of p_{G-u}:



p_{G-u}	0	0	0	0
p_{G-u-v_1}	$-$	$+$	$-$	$+$
p_{G-u-v_2}	$-$	$+$	$-$	$+$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
p_G	$+$	$-$	$+$	$-$

- By sign alts, we get interlacing of roots for p_G and p_{G-u}. ☺
- Next prove for $z \in (-\frac{1}{4\Delta}, 0]$:

$$2p_G(z) > p_{G-u}(z) > 0.$$

- Induction step:

$$p_G(z) \geqslant (1+2\Delta z) p_{G-u} \geqslant \tfrac{1}{2} p_{G-u}(z)$$

- No roots $\geqslant -1/4\Delta$ ☺