Semidefinite programming and vectors

Péter Hajnal

2024. Fall

Reminder

Reminder

We remind you of two descriptions of positive semidefinite (thus symmetric) matrices:

(1) Their eigenvalues are non-negative,

(2) they are Gram matrices of a set of vectors.

Reminder

We remind you of two descriptions of positive semidefinite (thus symmetric) matrices:

(1) Their eigenvalues are non-negative,

(2) they are Gram matrices of a set of vectors.

• Last time, using the eigenvalue interpretation, we formulated several problems related to multiple eigenvalues as SDP problems.

Reminder

We remind you of two descriptions of positive semidefinite (thus symmetric) matrices:

(1) Their eigenvalues are non-negative,

(2) they are Gram matrices of a set of vectors.

• Last time, using the eigenvalue interpretation, we formulated several problems related to multiple eigenvalues as SDP problems.

• Now, we use the Gram matrix description to answer combinatorial optimization questions.

A fundamental question

A fundamental question

Claude Shannon (1916—2001), one of the founding figures of information theory, asked the following question:

A fundamental question

Claude Shannon (1916—2001), one of the founding figures of information theory, asked the following question:

- Given an alphabet in which certain letters are confusable.

A fundamental question

Claude Shannon (1916—2001), one of the founding figures of information theory, asked the following question:

• Given an alphabet in which certain letters are confusable.

• This relation extends to ℓ-length words: Two (same-length) words are confusable if the same letter or confusable letter pair appears in every position.

A fundamental question

Claude Shannon (1916—2001), one of the founding figures of information theory, asked the following question:

• Given an alphabet in which certain letters are confusable.

• This relation extends to ℓ-length words: Two (same-length) words are confusable if the same letter or confusable letter pair appears in every position.

• Equivalently, two words are not confusable if in some position, two different, non-confusable letter pairs appear. These concepts can also be formulated in the language of graph theory.

A fundamental question

Claude Shannon (1916—2001), one of the founding figures of information theory, asked the following question:

• Given an alphabet in which certain letters are confusable.

• This relation extends to ℓ-length words: Two (same-length) words are confusable if the same letter or confusable letter pair appears in every position.

• Equivalently, two words are not confusable if in some position, two different, non-confusable letter pairs appear. These concepts can also be formulated in the language of graph theory.

• The characters of the alphabet form the set V. This confusability relation is described by a graph G.

Graph Products

Graph Products

Definition

The product of graphs G and H, denoted $G \boxtimes H$, is the graph whose vertex set is $V(G) \times V(H)$, and (v, w) is connected to (v', w') if and only if one of the following holds:

(i) $v = v'$ and $ww' \in E(H)$,

(ii) $vv' \in E(H)$ and $w = w'$,

(iii) $vv' \in E(G)$ and $ww' \in E(H)$.

Graph Products

Definition

The product of graphs G and H, denoted $G \boxtimes H$, is the graph whose vertex set is $V(G) \times V(H)$, and (v, w) is connected to (v', w') if and only if one of the following holds:

(i) $v = v'$ and $ww' \in E(H)$,

(ii) $vv' \in E(H)$ and $w = w'$,

(iii) $vv' \in E(G)$ and $ww' \in E(H)$.

• The product of two edges results in a complete graph with four vertices. Hence the notation.

Graph Theoretical Reformulation

Observation

It's easy to see that if G is the confusability graph of an alphabet, then $G^{\boxtimes k} := G \boxtimes G \boxtimes \ldots \boxtimes G$ where k times product has vertices as k-length words and adjacency describes the confusability relation.

Graph Theoretical Reformulation

Observation

It's easy to see that if G is the confusability graph of an alphabet, then $G^{\boxtimes k} := G \boxtimes G \boxtimes \ldots \boxtimes G$ where k times product has vertices as k-length words and adjacency describes the confusability relation.

• Shannon's question was: How many pairwise non-confusable words can we select from ℓ-length words?

Graph Theoretical Reformulation

Observation

It's easy to see that if G is the confusability graph of an alphabet, then $G^{\boxtimes k} := G \boxtimes G \boxtimes \ldots \boxtimes G$ where k times product has vertices as k-length words and adjacency describes the confusability relation.

- Shannon's question was: How many pairwise non-confusable words can we select from ℓ-length words?
- For $\ell = 1$, this is obviously $\alpha(G)$.

Graph Theoretical Reformulation

Observation

It's easy to see that if G is the confusability graph of an alphabet, then $G^{\boxtimes k} := G \boxtimes G \boxtimes \ldots \boxtimes G$ where k times product has vertices as k-length words and adjacency describes the confusability relation.

- Shannon's question was: How many pairwise non-confusable words can we select from ℓ-length words?
- For $\ell = 1$, this is obviously $\alpha(G)$.
- Generally, the answer is $\alpha(G^{\boxtimes \ell})$.

Graph Shannon Capacity

Graph Shannon Capacity

It's easy to see that the order of magnitude of the answer to the counting question (as ℓ increases) is exponential. Without proof, we state the following mathematical assertion.

Graph Shannon Capacity

It's easy to see that the order of magnitude of the answer to the counting question (as ℓ increases) is exponential. Without proof, we state the following mathematical assertion.

Fekete/Subadditivity Lemma

Let G be a simple graph. Then $\left(\sqrt[\ell]{\alpha(G^{\boxtimes \ell})}\right)_{\ell=1}^{\infty}$ is a convergent sequence.

Graph Shannon Capacity

It's easy to see that the order of magnitude of the answer to the counting question (as ℓ increases) is exponential. Without proof, we state the following mathematical assertion.

Fekete/Subadditivity Lemma

Let G be a simple graph. Then $\left(\sqrt[\ell]{\alpha(G^{\boxtimes \ell})}\right)_{\ell=1}^{\infty}$ is a convergent sequence.

Definition

$$\mathrm{Sh}(G) = \lim_{\ell \to \infty} \sqrt[\ell]{\alpha(G^{\boxtimes \ell})},$$

the Shannon theta function or Shannon capacity of G.

Graph Shannon Capacity

It's easy to see that the order of magnitude of the answer to the counting question (as ℓ increases) is exponential. Without proof, we state the following mathematical assertion.

Fekete/Subadditivity Lemma

Let G be a simple graph. Then $\left(\sqrt[\ell]{\alpha(G^{\boxtimes \ell})}\right)_{\ell=1}^{\infty}$ is a convergent sequence.

Definition

$$\mathrm{Sh}(G) = \lim_{\ell\to\infty} \sqrt[\ell]{\alpha(G^{\boxtimes \ell})},$$

the Shannon theta function or Shannon capacity of G.

The relatively simple concept hides a very difficult mathematical problem.

The Basic Sandwich

The Basic Sandwich

Observation

$$\alpha(G) \leq \mathrm{Sh}(G) \leq \overline{\chi}(G).$$

The Basic Sandwich

Observation

$$\alpha(G) \le \mathrm{Sh}(G) \le \overline{\chi}(G).$$

What is $\overline{\chi}(G)$?

The Basic Sandwich

Observation

$$\alpha(G) \leq \mathrm{Sh}(G) \leq \overline{\chi}(G).$$

What is $\overline{\chi}(G)$?

Definitions

$\overline{\chi}(G) = \chi(\overline{G})$

The Basic Sandwich

Observation

$$\alpha(G) \leq \mathrm{Sh}(G) \leq \overline{\chi}(G).$$

What is $\overline{\chi}(G)$?

Definitions

$\overline{\chi}(G) = \chi(\overline{G})$

The *clique covering number* of G: How few cliques can cover $V(G)$?

The Basic Sandwich

Observation

$$\alpha(G) \leq \mathrm{Sh}(G) \leq \overline{\chi}(G).$$

What is $\overline{\chi}(G)$?

Definitions

$\overline{\chi}(G) = \chi(\overline{G})$

The *clique covering number* of G: How few cliques can cover $V(G)$?

$\alpha(G) \leq \overline{\chi}(G)$ is obvious.

Making the Basic Sandwich

Making the Basic Sandwich

- Let F be an $\alpha(G)$-sized independent set in G. Then F^ℓ is an independent set in $G^{\boxtimes \ell}$.

Making the Basic Sandwich

- Let F be an $\alpha(G)$-sized independent set in G. Then F^ℓ is an independent set in $G^{\boxtimes \ell}$.

- Take a clique covering of G with $\overline{\chi}(G)$ classes.

Making the Basic Sandwich

• Let F be an $\alpha(G)$-sized independent set in G. Then F^ℓ is an independent set in $G^{\boxtimes\ell}$.

• Take a clique covering of G with $\overline{\chi}(G)$ classes. It's easy to see that this classification yields a $\overline{\chi}^\ell(G)$ classification of $G^{\boxtimes\ell}$ where each class is a clique.

Making the Basic Sandwich

- Let F be an $\alpha(G)$-sized independent set in G. Then F^ℓ is an independent set in $G^{\boxtimes \ell}$.

- Take a clique covering of G with $\overline{\chi}(G)$ classes. It's easy to see that this classification yields a $\overline{\chi}^\ell(G)$ classification of $G^{\boxtimes \ell}$ where each class is a clique.

- Hence

$$\alpha^\ell(G) \leq \alpha(G^{\boxtimes \ell}) \leq \overline{\chi}(G^{\boxtimes \ell}) \leq \overline{\chi}^\ell(G).$$

Making the Basic Sandwich

- Let F be an $\alpha(G)$-sized independent set in G. Then F^ℓ is an independent set in $G^{\boxtimes\ell}$.

- Take a clique covering of G with $\overline{\chi}(G)$ classes. It's easy to see that this classification yields a $\overline{\chi}^\ell(G)$ classification of $G^{\boxtimes\ell}$ where each class is a clique.

- Hence

$$\alpha^\ell(G) \le \alpha(G^{\boxtimes\ell}) \le \overline{\chi}(G^{\boxtimes\ell}) \le \overline{\chi}^\ell(G).$$

- Hence

$$\alpha(G) \le \mathrm{Sh}(G) \le \overline{\chi}(G).$$

Graphs We Already Know Everything About

Consequence

If G is such that $\alpha(G) = \overline{\chi}(G)$, then

$$\mathrm{Sh}(G) = \alpha(G).$$

Graphs We Already Know Everything About

Consequence

If G is such that $\alpha(G) = \overline{\chi}(G)$, then

$$\mathrm{Sh}(G) = \alpha(G).$$

- The condition stated in the theorem is not so rare.

Graphs We Already Know Everything About

Consequence

If G is such that $\alpha(G) = \overline{\chi}(G)$, then

$$\mathrm{Sh}(G) = \alpha(G).$$

- The condition stated in the theorem is not so rare.
- For example, every perfect (e.g., bipartite) graph satisfies it.

Graphs We Already Know Everything About

Consequence

If G is such that $\alpha(G) = \overline{\chi}(G)$, then

$$\mathrm{Sh}(G) = \alpha(G).$$

- The condition stated in the theorem is not so rare.
- For example, every perfect (e.g., bipartite) graph satisfies it.
- Actually, we get equality for complements of nice graphs.

Graphs We Already Know Everything About

Consequence

If G is such that $\alpha(G) = \overline{\chi}(G)$, then

$$\mathrm{Sh}(G) = \alpha(G).$$

- The condition stated in the theorem is not so rare.
- For example, every perfect (e.g., bipartite) graph satisfies it.
- Actually, we get equality for complements of nice graphs.
- The smallest graph for which the five-cycle (C_5) fails to satisfy it: $\alpha(C_5) = 2 < 3 = \overline{\chi}(G)$.

The first problem

The first problem

Determining $\mathrm{Sh}(C_5)$ is a serious mathematical problem.

The first problem

Determining $\mathrm{Sh}(C_5)$ is a serious mathematical problem.

• After posing the question, it took more than a decade to solve.
László Lovász's proof became a central tool in optimization.

The first problem

Determining $\mathrm{Sh}(C_5)$ is a serious mathematical problem.

• After posing the question, it took more than a decade to solve. László Lovász's proof became a central tool in optimization.

• Based on initial thoughts, $2 \leq \mathrm{Sh}(C_5) \leq 3$.

The first problem

Determining $\mathrm{Sh}(C_5)$ is a serious mathematical problem.

• After posing the question, it took more than a decade to solve.
László Lovász's proof became a central tool in optimization.

• Based on initial thoughts, $2 \leq \mathrm{Sh}(C_5) \leq 3$.

• Improving the lower bound is simpler.

The first problem

Determining $\mathrm{Sh}(C_5)$ is a serious mathematical problem.

- After posing the question, it took more than a decade to solve.

László Lovász's proof became a central tool in optimization.

- Based on initial thoughts, $2 \leq \mathrm{Sh}(C_5) \leq 3$.
- Improving the lower bound is simpler.

Lemma

$$\sqrt{5} \leq \mathrm{Sh}(C_5).$$

The first problem

Determining $\mathrm{Sh}(C_5)$ is a serious mathematical problem.

• After posing the question, it took more than a decade to solve. László Lovász's proof became a central tool in optimization.

• Based on initial thoughts, $2 \leq \mathrm{Sh}(C_5) \leq 3$.

• Improving the lower bound is simpler.

Lemma

$$\sqrt{5} \leq \mathrm{Sh}(C_5).$$

The lemma is easily verifiable.

Lower bound for $\mathrm{Sh}(C_5)$

Lower bound for $\mathrm{Sh}(C_5)$

For even k

$$\alpha\left(C_5^{\boxtimes k}\right) = \alpha\left((C_5 \boxtimes C_5)^{\boxtimes k/2}\right) \geq 5^{k/2} = \sqrt{5}^k,$$

Consequence

$$\sqrt{5} \leq \mathrm{Sh}(C_5).$$

Lower bound for $\mathrm{Sh}(C_5)$

For even k

$$\alpha\left(C_5^{\boxtimes k}\right) = \alpha\left(\left(C_5 \boxtimes C_5\right)^{\boxtimes k/2}\right) \geq 5^{k/2} = \sqrt{5}^k,$$

Consequence

$$\sqrt{5} \leq \mathrm{Sh}(C_5).$$

Strengthening the upper bound is the essence of Lovász's solution. It revolutionizes the concept of clique covers.

Revisiting clique covers

Revisiting clique covers

First, let's define the clique cover of a graph G with k cliques.

Revisiting clique covers

First, let's define the clique cover of a graph G with k cliques.

Clique cover

A function $c : V(G) \to \{e_1, e_2, \ldots, e_k\}$ is a clique cover if for every $uv \notin E(G)$ edge, $c(u) = e_i$, $c(v) = e_j$ implies $i \neq j$.

Revisiting clique covers

First, let's define the clique cover of a graph G with k cliques.

Clique cover

A function $c : V(G) \to \{e_1, e_2, \ldots, e_k\}$ is a clique cover if for every $uv \notin E(G)$ edge, $c(u) = e_i$, $c(v) = e_j$ implies $i \neq j$.

- We think of the e_i's as colors.

Revisiting clique covers

First, let's define the clique cover of a graph G with k cliques.

Clique cover

A function $c : V(G) \to \{e_1, e_2, \ldots, e_k\}$ is a clique cover if for every $uv \notin E(G)$ edge, $c(u) = e_i$, $c(v) = e_j$ implies $i \neq j$.

- We think of the e_i's as colors.

- In a clique cover, images/colors of non-adjacent vertices are distinct.

Orthonormal representation of graphs

Orthonormal representation of graphs

László Lovász replaces colors with vectors, and distinctness with orthogonality.

Orthonormal representation of graphs

László Lovász replaces colors with vectors, and distinctness with orthogonality.

Definition

Let G be a simple graph.

$$\rho : V(G) \to \mathbb{R}^d \qquad \text{i.e.,} \qquad (\rho_v)_{v \in V} \in \mathbb{R}^{V(G)}$$

is an orthonormal representation (ONR) of G if the ρ_v vectors $(v \in V)$ are unit vectors ($\rho : V(G) \to \mathbb{S}^{d-1} \subset \mathbb{R}^d$) and $\rho_u \perp \rho_v$ for every $uv \notin E$.

Vector clique cover

Vector clique cover

The ONR can be viewed as a vector clique cover.

Vector clique cover

The ONR can be viewed as a vector clique cover.

• The usual clique cover becomes a vector clique cover if we think of the e_i's as pairwise orthogonal unit vectors (instead of the original color interpretation).

Vector clique cover

The ONR can be viewed as a vector clique cover.

• The usual clique cover becomes a vector clique cover if we think of the e_i's as pairwise orthogonal unit vectors (instead of the original color interpretation). That is, $\{e_i\}$ can be the standard basis of $\mathbb{R}^k$ (where k is the number of cliques).

Vector clique cover

The ONR can be viewed as a vector clique cover.

• The usual clique cover becomes a vector clique cover if we think of the e_i's as pairwise orthogonal unit vectors (instead of the original color interpretation). That is, $\{e_i\}$ can be the standard basis of $\mathbb{R}^k$ (where k is the number of cliques).

• What will be the new concept, the color demand of a vector clique cover? To answer this, let's take a detour.

Detour: Pythagoras' theorem

Detour: Pythagoras' theorem

- Let $e_1, e_2, \ldots, e_k$ be pairwise orthogonal unit vectors and h be any unit vector.

Detour: Pythagoras' theorem

- Let $e_1, e_2, \ldots, e_k$ be pairwise orthogonal unit vectors and h be any unit vector.
- If (e_i) were a basis of our space,

$$1 = |h|^2 = h^\mathsf{T} h = \sum_{i=1}^{k} (e_i^\mathsf{T} h)^2.$$

This is a higher-dimensional form of Pythagoras' theorem. In general, we can state the following lemma.

Detour: Pythagoras' theorem

- Let $e_1, e_2, \ldots, e_k$ be pairwise orthogonal unit vectors and h be any unit vector.
- If (e_i) were a basis of our space,

$$1 = |h|^2 = h^{\mathsf{T}} h = \sum_{i=1}^{k} (e_i^{\mathsf{T}} h)^2.$$

This is a higher-dimensional form of Pythagoras' theorem. In general, we can state the following lemma.

Lemma

Let $e_1, e_2, \ldots, e_k$ be pairwise orthogonal unit vectors and h be any unit vector. Then

$$1 = |h|^2 = h^{\mathsf{T}} h \geq \sum_{i=1}^{k} (e_i^{\mathsf{T}} h)^2.$$

The classic clique cover as vector clique cover

The classic clique cover as vector clique cover

- From the lemma, it follows that

$$\min_{i=1,2,\ldots,k}(e_i^{\mathsf{T}} h)^2 \leq \frac{1}{k}.$$

The classic clique cover as vector clique cover

- From the lemma, it follows that

$$\min_{i=1,2,\dots,k} (e_i^{\mathsf{T}} h)^2 \leq \frac{1}{k}.$$

- Alternatively,

$$\max_{i=1,2,\dots,k} \frac{1}{(e_i^{\mathsf{T}} h)^2} \geq k.$$

The classic clique cover as vector clique cover

- From the lemma, it follows that

$$\min_{i=1,2,\ldots,k} (e_i^{\mathsf{T}} h)^2 \leq \frac{1}{k}.$$

- Alternatively,

$$\max_{i=1,2,\ldots,k} \frac{1}{(e_i^{\mathsf{T}} h)^2} \geq k.$$

- If $h = 1/\sqrt{k}(e_1 + e_2 + \ldots + e_k)$ (unit vector):

$$\max_{i=1,2,\ldots,k} \frac{1}{(e_i^{\mathsf{T}} h)^2} = k.$$

The classic clique cover as vector clique cover

- From the lemma, it follows that

$$\min_{i=1,2,\ldots,k}(e_i^\mathsf{T} h)^2 \leq \frac{1}{k}.$$

- Alternatively,

$$\max_{i=1,2,\ldots,k}\frac{1}{(e_i^\mathsf{T} h)^2} \geq k.$$

- If $h = 1/\sqrt{k}(e_1 + e_2 + \ldots + e_k)$ (unit vector):

$$\max_{i=1,2,\ldots,k}\frac{1}{(e_i^\mathsf{T} h)^2} = k.$$

- Based on the above, if we "color" in the ONR, then

$$\min_h \max_{i=1,2,\ldots,k}\frac{1}{(e_i^\mathsf{T} h)^2} = k,$$

the classic clique cover's color demand.

The Lovász parameter

The Lovász parameter

Definition

For an ONR $((\rho_v)_{v\in V}, h)$ and a unit vector h (henceforth referred to as the handle), we assign a value:

$$\mathrm{Lov}(((\rho_v)_{v\in V}, h)) = \max_{v:v\in V} \frac{1}{(h^{\mathsf{T}}\rho_v)^2}.$$

The Lovász parameter

Definition

For an ONR $((\rho_v)_{v\in V}, h)$ and a unit vector h (henceforth referred to as the handle), we assign a value:

$$\mathrm{Lov}(((\rho_v)_{v\in V}, h)) = \max_{v:v\in V} \frac{1}{(h^{\mathsf{T}}\rho_v)^2}.$$

Definition

$$\mathrm{Lov}(G) = \inf\{\mathrm{Lov}(((\rho_v)_{v\in V}, h)) : (\rho_v)_{v\in V} \text{ is an ONR}, h \text{ is a handle}\}.$$

The Lovász parameter

Definition

For an ONR $((\rho_v)_{v\in V}, h)$ and a unit vector h (henceforth referred to as the handle), we assign a value:

$$\mathrm{Lov}(((\rho_v)_{v\in V}, h)) = \max_{v:v\in V} \frac{1}{(h^{\mathsf{T}}\rho_v)^2}.$$

Definition

$$\mathrm{Lov}(G) = \inf\{\mathrm{Lov}(((\rho_v)_{v\in V}, h)) : (\rho_v)_{v\in V} \text{ is an ONR}, h \text{ is a handle}\}.$$

Interpreting the classic clique cover as a vector clique cover provides a challenger in the definition of $\mathrm{Lov}(G)$. Thus,

The Lovász parameter

Definition

For an ONR $((\rho_v)_{v\in V}, h)$ and a unit vector h (henceforth referred to as the handle), we assign a value:

$$\mathrm{Lov}(((\rho_v)_{v\in V}, h)) = \max_{v:v\in V} \frac{1}{(h^{\mathsf{T}}\rho_v)^2}.$$

Definition

$$\mathrm{Lov}(G) = \inf\{\mathrm{Lov}(((\rho_v)_{v\in V}, h)) : (\rho_v)_{v\in V} \text{ is an ONR}, h \text{ is a handle}\}.$$

Interpreting the classic clique cover as a vector clique cover provides a challenger in the definition of $\mathrm{Lov}(G)$. Thus,

Consequence

$$\mathrm{Lov}(G) \le \overline{\chi}(G).$$

EXAMPLE: Umbrella construction

EXAMPLE: Umbrella construction

- Let $G = C_5$, suppose its vertex set is $\{1, 2, 3, 4, 5\}$ (neighborhood in "modulo 5 arithmetic 1 apart").

EXAMPLE: Umbrella construction

- Let $G = C_5$, suppose its vertex set is $\{1, 2, 3, 4, 5\}$ (neighborhood in "modulo 5 arithmetic 1 apart").
- Let $h = \rho_1 = \rho_2 = \rho_3 = \rho_4 = \rho_5$, six unit vectors glued together at one endpoint (obviously not an ONR).

EXAMPLE: Umbrella construction

- Let $G = C_5$, suppose its vertex set is $\{1, 2, 3, 4, 5\}$ (neighborhood in "modulo 5 arithmetic 1 apart").
- Let $h = \rho_1 = \rho_2 = \rho_3 = \rho_4 = \rho_5$, six unit vectors glued together at one endpoint (obviously not an ONR).
- Think of h as the handle of a collapsed umbrella, and $\rho_1 = \rho_2 = \rho_3 = \rho_4 = \rho_5$ as its ribs. Let's start opening the umbrella.

EXAMPLE: Umbrella construction

- Let $G = C_5$, suppose its vertex set is $\{1, 2, 3, 4, 5\}$ (neighborhood in "modulo 5 arithmetic 1 apart").
- Let $h = \rho_1 = \rho_2 = \rho_3 = \rho_4 = \rho_5$, six unit vectors glued together at one endpoint (obviously not an ONR).
- Think of h as the handle of a collapsed umbrella, and $\rho_1 = \rho_2 = \rho_3 = \rho_4 = \rho_5$ as its ribs. Let's start opening the umbrella.
- The handle points stably downwards, the ribs open symmetrically.

EXAMPLE: Umbrella construction

- Let $G = C_5$, suppose its vertex set is $\{1, 2, 3, 4, 5\}$ (neighborhood in "modulo 5 arithmetic 1 apart").
- Let $h = \rho_1 = \rho_2 = \rho_3 = \rho_4 = \rho_5$, six unit vectors glued together at one endpoint (obviously not an ONR).
- Think of h as the handle of a collapsed umbrella, and $\rho_1 = \rho_2 = \rho_3 = \rho_4 = \rho_5$ as its ribs. Let's start opening the umbrella.
- The handle points stably downwards, the ribs open symmetrically.
- At every moment, their endpoints lie in a plane, forming the vertices of a regular pentagon.

EXAMPLE: Umbrella construction

- Let $G = C_5$, suppose its vertex set is $\{1, 2, 3, 4, 5\}$ (neighborhood in "modulo 5 arithmetic 1 apart").
- Let $h = \rho_1 = \rho_2 = \rho_3 = \rho_4 = \rho_5$, six unit vectors glued together at one endpoint (obviously not an ONR).
- Think of h as the handle of a collapsed umbrella, and $\rho_1 = \rho_2 = \rho_3 = \rho_4 = \rho_5$ as its ribs. Let's start opening the umbrella.
- The handle points stably downwards, the ribs open symmetrically.
- At every moment, their endpoints lie in a plane, forming the vertices of a regular pentagon.
- In a suitable position, the handle and the five ribs provide an ONR of C_5. This is the umbrella representation of C_5.

EXAMPLE: Umbrella construction

- Let $G = C_5$, suppose its vertex set is $\{1, 2, 3, 4, 5\}$ (neighborhood in "modulo 5 arithmetic 1 apart").
- Let $h = \rho_1 = \rho_2 = \rho_3 = \rho_4 = \rho_5$, six unit vectors glued together at one endpoint (obviously not an ONR).
- Think of h as the handle of a collapsed umbrella, and $\rho_1 = \rho_2 = \rho_3 = \rho_4 = \rho_5$ as its ribs. Let's start opening the umbrella.
- The handle points stably downwards, the ribs open symmetrically.
- At every moment, their endpoints lie in a plane, forming the vertices of a regular pentagon.
- In a suitable position, the handle and the five ribs provide an ONR of C_5. This is the umbrella representation of C_5.
- A simple high school geometry calculation yields that the representation, along with the handle, has a value of $\sqrt{5}$.

Consequences, connections

Consequences, connections

For C_5, we have 3 as the value of $\overline{\chi}$.

Consequences, connections

For C_5, we have 3 as the value of $\overline{\chi}$. Interpreting the classic clique cover as a vector clique cover provides a challenger in the definition of $\mathrm{Lov}(C_5)$.

Consequences, connections

For C_5, we have 3 as the value of $\overline{\chi}$. Interpreting the classic clique cover as a vector clique cover provides a challenger in the definition of $\mathrm{Lov}(C_5)$. Distributing three pairwise orthogonal unit vectors among five vertices in an asymmetric way.

Consequences, connections

For C_5, we have 3 as the value of $\overline{\chi}$. Interpreting the classic clique cover as a vector clique cover provides a challenger in the definition of $\mathrm{Lov}(C_5)$. Distributing three pairwise orthogonal unit vectors among five vertices in an asymmetric way. Based on the umbrella construction, a better estimate for $\mathrm{Lov}(C_5)$ can be given.

Consequences, connections

For C_5, we have 3 as the value of $\overline{\chi}$. Interpreting the classic clique cover as a vector clique cover provides a challenger in the definition of $\text{Lov}(C_5)$. Distributing three pairwise orthogonal unit vectors among five vertices in an asymmetric way. Based on the umbrella construction, a better estimate for $\text{Lov}(C_5)$ can be given.

Consequence

The umbrella representation of C_5 proves that $\text{Lov}(C_5) \le \sqrt{5}$.

Consequences, connections

For C_5, we have 3 as the value of $\overline{\chi}$. Interpreting the classic clique cover as a vector clique cover provides a challenger in the definition of $\mathrm{Lov}(C_5)$. Distributing three pairwise orthogonal unit vectors among five vertices in an asymmetric way. Based on the umbrella construction, a better estimate for $\mathrm{Lov}(C_5)$ can be given.

Consequence

The umbrella representation of C_5 proves that $\mathrm{Lov}(C_5) \leq \sqrt{5}$.

There is a close relationship between the Lovász function and the Shannon capacity:

Consequences, connections

For C_5, we have 3 as the value of $\overline{\chi}$. Interpreting the classic clique cover as a vector clique cover provides a challenger in the definition of $\mathrm{Lov}(C_5)$. Distributing three pairwise orthogonal unit vectors among five vertices in an asymmetric way. Based on the umbrella construction, a better estimate for $\mathrm{Lov}(C_5)$ can be given.

Consequence

The umbrella representation of C_5 proves that $\mathrm{Lov}(C_5) \leq \sqrt{5}$.

There is a close relationship between the Lovász function and the Shannon capacity:

Theorem

(i) $\alpha(G) \leq \mathrm{Lov}(G)$.

(ii) $\mathrm{Sh}(G) \leq \mathrm{Lov}(G) \leq \overline{\chi}(G)$.

Proof of (i)

Proof of (i)

- Let G be an arbitrary graph with a maximal independent vertex set F.

Proof of (i)

- Let G be an arbitrary graph with a maximal independent vertex set F.
- Consider an arbitrary ONR of G with an arbitrary handle.

Proof of (i)

- Let G be an arbitrary graph with a maximal independent vertex set F.
- Consider an arbitrary ONR of G with an arbitrary handle.
- Associate pairwise orthogonal unit vectors with the elements of F.

Proof of (i)

- Let G be an arbitrary graph with a maximal independent vertex set F.
- Consider an arbitrary ONR of G with an arbitrary handle.
- Associate pairwise orthogonal unit vectors with the elements of F.
- Thus $\sum_{f\in F}(h^{\mathsf{T}}\rho_f)^2 \leq |h|^2 = 1$.

Proof of (i)

- Let G be an arbitrary graph with a maximal independent vertex set F.
- Consider an arbitrary ONR of G with an arbitrary handle.
- Associate pairwise orthogonal unit vectors with the elements of F.
- Thus $\sum_{f\in F}(h^{\mathsf{T}}\rho_f)^2 \leq |h|^2 = 1$.
- This implies $\min_{f\in F}(h^{\mathsf{T}}\rho_f)^2 \leq 1/|F|$.

Proof of (i)

- Let G be an arbitrary graph with a maximal independent vertex set F.
- Consider an arbitrary ONR of G with an arbitrary handle.
- Associate pairwise orthogonal unit vectors with the elements of F.
- Thus $\sum_{f\in F}(h^{\mathsf{T}}\rho_f)^2 \leq |h|^2 = 1$.
- This implies $\min_{f\in F}(h^{\mathsf{T}}\rho_f)^2 \leq 1/|F|$.
- Moreover,

$$\mathrm{Lov}(\rho, h) \geq \max_{f\in F} \frac{1}{(h^{\mathsf{T}}\rho_f)^2} \geq |F| = \alpha(G).$$

Proof of (ii)

Proof of (ii)

- Let $(\rho_v)_{v\in V(G)}, h$ be an ONR of G with a handle, corresponding to the parameter $\mathrm{Lov}(G)$.

Proof of (ii)

- Let $(\rho_v)_{v\in V(G)}, h$ be an ONR of G with a handle, corresponding to the parameter $\mathrm{Lov}(G)$.
- From this, we can easily construct an ONR of $G^{\boxtimes \ell}$ with a new handle, whose value will be $\mathrm{Lov}^{\ell}(G)$.

Proof of (ii)

- Let $(\rho_v)_{v\in V(G)}, h$ be an ONR of G with a handle, corresponding to the parameter $\mathrm{Lov}(G)$.
- From this, we can easily construct an ONR of $G^{\boxtimes \ell}$ with a new handle, whose value will be $\mathrm{Lov}^\ell(G)$.
- We assign an $(v_1, v_2, \ldots, v_\ell)$ vertex of the product graph to the vector $\rho_{v_1} \otimes \rho_{v_1} \otimes \ldots \otimes \rho_{v_\ell}$, while the handle becomes $h \otimes h \otimes \ldots \otimes h$.

Definition: tensor product of vectors

For vectors $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^e$, $x \otimes y \in \mathbb{R}^{d\cdot e}$, where the (i,j) component is $x_i y_j$. Alternatively, $x \otimes y \in \mathbb{R}^{d\times e}$ represents the matrix xy^T as a vector.

Completing the proof of (ii)

Completing the proof of (ii)

- The details rely on the relationship

$$(x_1 \otimes x_2 \otimes \ldots \otimes x_\ell)^\mathsf{T}(y_1 \otimes y_2 \otimes \ldots \otimes y_\ell) = (x_1^\mathsf{T} y_1)(x_2^\mathsf{T} y_2)\ldots(x_\ell^\mathsf{T} y_\ell)$$

.

Completing the proof of (ii)

- The details rely on the relationship

$$(x_1 \otimes x_2 \otimes \ldots \otimes x_\ell)^{\mathsf{T}}(y_1 \otimes y_2 \otimes \ldots \otimes y_\ell) = (x_1^{\mathsf{T}} y_1)(x_2^{\mathsf{T}} y_2) \ldots (x_\ell^{\mathsf{T}} y_\ell)$$

.

- The verification of this relationship and the details are left to the interested reader.

Completing the proof of (ii)

- The details rely on the relationship

$$(x_1 \otimes x_2 \otimes \ldots \otimes x_\ell)^{\mathsf{T}}(y_1 \otimes y_2 \otimes \ldots \otimes y_\ell) = (x_1^{\mathsf{T}} y_1)(x_2^{\mathsf{T}} y_2) \ldots (x_\ell^{\mathsf{T}} y_\ell)$$

.

- The verification of this relationship and the details are left to the interested reader.
- Then, it immediately follows that

$$\alpha(G^{\boxtimes \ell}) \le \mathrm{Lov}(G^{\boxtimes \ell}) \le \mathrm{Lov}^{\ell}(G).$$

Completing the proof of (ii)

- The details rely on the relationship

$$(x_1 \otimes x_2 \otimes \ldots \otimes x_\ell)^{\mathsf{T}}(y_1 \otimes y_2 \otimes \ldots \otimes y_\ell) = (x_1^{\mathsf{T}} y_1)(x_2^{\mathsf{T}} y_2)\ldots(x_\ell^{\mathsf{T}} y_\ell)$$

.

- The verification of this relationship and the details are left to the interested reader.
- Then, it immediately follows that

$$\alpha(G^{\boxtimes \ell}) \leq \mathrm{Lov}(G^{\boxtimes \ell}) \leq \mathrm{Lov}^{\ell}(G).$$

- From this, the assertion of (ii) can be easily derived.

Summary

Summary

Assembling our knowledge so far about C_5:

$$\sqrt{5} \leq \mathrm{Sh}(C_5) \leq \mathrm{Lov}(C_5) \leq \sqrt{5}.$$

Summary

Assembling our knowledge so far about C_5:

$$\sqrt{5} \leq \mathrm{Sh}(C_5) \leq \mathrm{Lov}(C_5) \leq \sqrt{5}.$$

Lovász László's theorem

$\mathrm{Sh}(C_5) = \mathrm{Lov}(C_5) = \sqrt{5}$.

Summary

Assembling our knowledge so far about C_5:

$$\sqrt{5} \leq \mathrm{Sh}(C_5) \leq \mathrm{Lov}(C_5) \leq \sqrt{5}.$$

Lovász László's theorem

$\mathrm{Sh}(C_5) = \mathrm{Lov}(C_5) = \sqrt{5}.$

Lovász László's theorem

$$\alpha(G) \leq \mathrm{Sh}(G) \leq \mathrm{Lov}(G) \leq \overline{\chi}(G).$$

Summary

Assembling our knowledge so far about C_5:

$$\sqrt{5} \leq \mathrm{Sh}(C_5) \leq \mathrm{Lov}(C_5) \leq \sqrt{5}.$$

Lovász László's theorem

$\mathrm{Sh}(C_5) = \mathrm{Lov}(C_5) = \sqrt{5}$.

Lovász László's theorem

$$\alpha(G) \leq \mathrm{Sh}(G) \leq \mathrm{Lov}(G) \leq \overline{\chi}(G).$$

We mention that for C_7 the value of Lovász's theta function can be determined without much trouble (the extension of the umbrella construction gives the optimal representation). The Shannon capacity of C_7 is still unknown to this day.

α, χ, Sh are hard. And Lov?

α, χ, Sh are hard. And Lov?

- Finally, we mention that determining Lov(G) can also be formulated as an SDP problem.

α, χ, Sh are hard. And Lov?

• Finally, we mention that determining Lov(G) can also be formulated as an SDP problem. This is not surprising. We need to find an optimal vector system.

α, χ, Sh are hard. And Lov?

- Finally, we mention that determining Lov(G) can also be formulated as an SDP problem. This is not surprising. We need to find an optimal vector system.

- This is determined up to isomorphism by the Gram matrix. So, we are actually looking for a special Gram matrix/positive semidefinite matrix.

α, χ, Sh are hard. And Lov?

- Finally, we mention that determining Lov(G) can also be formulated as an SDP problem. This is not surprising. We need to find an optimal vector system.

- This is determined up to isomorphism by the Gram matrix. So, we are actually looking for a special Gram matrix/positive semidefinite matrix.

- We can see that the optimization problem:

Minimize	$\lambda_{\max}(M)$-t
subject to	$M_{uu} = 1$ for every $u \in V$
	$M_{uv} = 1$ for every $uv \notin E$
	$M \in \mathcal{S}^n$.

has an optimal value of Lov(G).

Break

The Goal-Theorem: Lov(G) as an SDP

The Goal-Theorem: Lov(G) as an SDP

- That is, the Lovász theta-function, studied last week, coincides with the Lovász function introduced now.

The Goal-Theorem: Lov(G) as an SDP

- That is, the Lovász theta-function, studied last week, coincides with the Lovász function introduced now.

Theorem

$$\mathrm{Lov}(G) = \vartheta(G).$$

The Goal-Theorem: $\mathrm{Lov}(G)$ as an SDP

- That is, the Lovász theta-function, studied last week, coincides with the Lovász function introduced now.

Theorem

$$\mathrm{Lov}(G) = \vartheta(G).$$

- The theorem implies two-way inequality between the two optimal values.

The Goal-Theorem: Lov(G) as an SDP

• That is, the Lovász theta-function, studied last week, coincides with the Lovász function introduced now.

Theorem

$$\mathrm{Lov}(G) = \vartheta(G).$$

• The theorem implies two-way inequality between the two optimal values.

• However, our proof will be stronger. For both optimization problems, we will construct a possible solution of one from another, so that the value of the (appropriate) objective function does not increase.

From SDP solution to ONR-handle

Minimize	$\lambda_{\max}(M)$-t
subject to	$M_{uu} = 1$ for every $u \in V$
	$M_{uv} = 1$ for every $uv \notin E$
	$M \in \mathcal{S}^n$.

From SDP solution to ONR-handle

Minimize	$\lambda_{\max}(M)$-t
subject to	$M_{uu} = 1$ for every $u \in V$
	$M_{uv} = 1$ for every $uv \notin E$
	$M \in \mathcal{S}^n$.

- First, let M be a matrix that is a possible solution to the above problem.

From SDP solution to ONR-handle

Minimize	$\lambda_{\max}(M)$-t
subject to	$M_{uu} = 1$ for every $u \in V$
	$M_{uv} = 1$ for every $uv \notin E$
	$M \in \mathcal{S}^n$.

- First, let M be a matrix that is a possible solution to the above problem.
- Consider $\lambda_{\max}(M)I - M$.

From SDP solution to ONR-handle

Minimize	$\lambda_{\max}(M)$-t
subject to	$M_{uu} = 1$ for every $u \in V$
	$M_{uv} = 1$ for every $uv \notin E$
	$M \in \mathcal{S}^n$.

- First, let M be a matrix that is a possible solution to the above problem.
- Consider $\lambda_{\max}(M)I - M$.
- This is a positive semidefinite matrix (in fact, we know that its minimum eigenvalue is 0, specifically, it is not full rank).

From SDP solution to ONR-handle

Minimize	$\lambda_{\max}(M)$-t
subject to	$M_{uu} = 1$ for every $u \in V$
	$M_{uv} = 1$ for every $uv \notin E$
	$M \in \mathcal{S}^n$.

- First, let M be a matrix that is a possible solution to the above problem.
- Consider $\lambda_{\max}(M)I - M$.
- This is a positive semidefinite matrix (in fact, we know that its minimum eigenvalue is 0, specifically, it is not full rank).
- Thus, it is the Gram matrix of a vector system $(\pi_v)_{v \in V}$, (without exceeding the necessary vector space dimension, we don't need to go beyond $|V|$), and we can work even in $\mathbb{R}^{|V|-1}$ because of the lack of full rank.

From SDP solution to ONR-handle (continued)

From SDP solution to ONR-handle (continued)

- We know that

$$\pi_u^{\mathsf{T}}\pi_v = \begin{cases} \lambda_{\max} - 1, & \text{if } u = v \\ -1, & \text{if } uv \notin E(G) \end{cases}$$

From SDP solution to ONR-handle (continued)

- We know that

$$\pi_u^{\mathsf{T}}\pi_v = \begin{cases} \lambda_{\max} - 1, & \text{if } u = v \\ -1, & \text{if } uv \notin E(G) \end{cases}$$

- Let

$$\rho_v = \begin{pmatrix} 1 \\ \pi_v \end{pmatrix} \in \mathbb{R}^{|V|} \quad (v \in V), \qquad h = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \in \mathbb{R}^{|V|},$$

where $1 \in \mathbb{R}$, $\pi_v, 0 \in \mathbb{R}^{|V|-1}$.

From SDP solution to ONR-handle (continued)

- We know that

$$\pi_u^{\mathsf{T}}\pi_v = \begin{cases} \lambda_{\max} - 1, & \text{if } u = v \\ -1, & \text{if } uv \notin E(G) \end{cases}$$

- Let

$$\rho_v = \begin{pmatrix} 1 \\ \pi_v \end{pmatrix} \in \mathbb{R}^{|V|} \quad (v \in V), \qquad h = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \in \mathbb{R}^{|V|},$$

where $1 \in \mathbb{R}$, $\pi_v, 0 \in \mathbb{R}^{|V|-1}$.

- Then we know that

$$\rho_u^{\mathsf{T}}\rho_v = \begin{cases} \lambda_{\max}, & \text{if } u = v \\ 0, & \text{if } uv \notin E(G) \end{cases}$$

From SDP solution to ONR-handle (continued)

- We know that

$$\pi_u^{\mathsf{T}}\pi_v = \begin{cases} \lambda_{\max} - 1, & \text{if } u = v \\ -1, & \text{if } uv \notin E(G) \end{cases}$$

- Let

$$\rho_v = \begin{pmatrix} 1 \\ \pi_v \end{pmatrix} \in \mathbb{R}^{|V|} \quad (v \in V), \qquad h = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \in \mathbb{R}^{|V|},$$

where $1 \in \mathbb{R}$, $\pi_v, 0 \in \mathbb{R}^{|V|-1}$.

- Then we know that

$$\rho_u^{\mathsf{T}}\rho_v = \begin{cases} \lambda_{\max}, & \text{if } u = v \\ 0, & \text{if } uv \notin E(G) \end{cases}$$

- That is, ρ_v are identical (non-zero length) vectors, which are orthogonal if $uv \notin E$, and h is a unit vector.

From SDP solution to ONR-handle (completion)

From SDP solution to ONR-handle (completion)

- Let $\rho_v^0 = \frac{1}{|\rho_v|}\rho_v$, the normalized ρ_v vectors ($v \in V$).

From SDP solution to ONR-handle (completion)

- Let $\rho_v^0 = \frac{1}{|\rho_v|}\rho_v$, the normalized ρ_v vectors ($v \in V$).
- The first coordinates of ρ_v^0 vectors (i.e., the $h^\mathsf{T}\rho_v^0$ values) are all $\frac{1}{|\rho_v|}$.

From SDP solution to ONR-handle (completion)

- Let $\rho_v^0 = \frac{1}{|\rho_v|}\rho_v$, the normalized ρ_v vectors ($v \in V$).
- The first coordinates of ρ_v^0 vectors (i.e., the $h^\mathsf{T}\rho_v^0$ values) are all $\frac{1}{|\rho_v|}$.
- Thus, all $\frac{1}{(h^\mathsf{T}\rho_v^0)^2}$ values are $\lambda_{\max}$.

From SDP solution to ONR-handle (completion)

- Let $\rho_v^0 = \frac{1}{|\rho_v|}\rho_v$, the normalized ρ_v vectors ($v \in V$).
- The first coordinates of ρ_v^0 vectors (i.e., the $h^\mathsf{T}\rho_v^0$ values) are all $\frac{1}{|\rho_v|}$.
- Thus, all $\frac{1}{(h^\mathsf{T}\rho_v^0)^2}$ values are $\lambda_{\max}$.
- So, $(\rho_v^0)_{v\in V}$ is an ONR. Moreover, with the h handle, the Lovász parameter is $\lambda_{\max}(M)$.

From ONR-handle pair to SDP solution

From ONR-handle pair to SDP solution

- To achieve the reverse, start from an ONR $(\rho_v)_{v\in V}$ and a handle h.

From ONR-handle pair to SDP solution

- To achieve the reverse, start from an ONR $(\rho_v)_{v \in V}$ and a handle h.
- Scale the ρ's so that they point into the plane perpendicular to h.

From ONR-handle pair to SDP solution

- To achieve the reverse, start from an ONR $(\rho_v)_{v\in V}$ and a handle h.
- Scale the ρ's so that they point into the plane perpendicular to h.
- Take the vectors from h to each vertex. This gives us the vector system

$$\left(h - \frac{1}{h^{\mathsf{T}}\rho_u}\rho_u\right)_{u\in V}.$$

From ONR-handle pair to SDP solution

- To achieve the reverse, start from an ONR $(\rho_v)_{v\in V}$ and a handle h.
- Scale the ρ's so that they point into the plane perpendicular to h.
- Take the vectors from h to each vertex. This gives us the vector system

$$\left(h - \frac{1}{h^{\mathsf{T}}\rho_u}\rho_u\right)_{u\in V}.$$

- Examine the Gram matrix of this system.

From ONR-handle pair to SDP solution

- To achieve the reverse, start from an ONR $(\rho_v)_{v\in V}$ and a handle h.
- Scale the ρ's so that they point into the plane perpendicular to h.
- Take the vectors from h to each vertex. This gives us the vector system

$$\left(h - \frac{1}{h^{\mathsf{T}}\rho_u}\rho_u\right)_{u\in V}.$$

- Examine the Gram matrix of this system.
- The element in position uv is

$$M_{uv} = -1 + \frac{\rho_u^{\mathsf{T}}\rho_v}{(h^{\mathsf{T}}\rho_u)(h^{\mathsf{T}}\rho_v)}.$$

From ONR-handle pair to SDP solution

- To achieve the reverse, start from an ONR $(\rho_v)_{v\in V}$ and a handle h.
- Scale the ρ's so that they point into the plane perpendicular to h.
- Take the vectors from h to each vertex. This gives us the vector system

$$\left(h - \frac{1}{h^{\mathsf{T}}\rho_u}\rho_u\right)_{u\in V}.$$

- Examine the Gram matrix of this system.
- The element in position uv is

$$M_{uv} = -1 + \frac{\rho_u^{\mathsf{T}}\rho_v}{(h^{\mathsf{T}}\rho_u)(h^{\mathsf{T}}\rho_v)}.$$

- So, non-edge positions are -1, and the main diagonal is $-1 + 1/(h^{\mathsf{T}}\rho_v)^2$.

From ONR-handle pair to SDP solution (continued)

From ONR-handle pair to SDP solution (continued)

- Form the matrix $\widetilde{M}$ as follows: take M and round down its main diagonal elements to -1 (the rounding value is $-1/(h^{\mathsf{T}}\rho_v)^2$), then take the negative of our matrix.

From ONR-handle pair to SDP solution (continued)

- Form the matrix $\widetilde{M}$ as follows: take M and round down its main diagonal elements to -1 (the rounding value is $-1/(h^\mathsf{T}\rho_v)^2$), then take the negative of our matrix.
- $\widetilde{M}$ is a possible solution to our optimization problem (the symmetry of our matrix is obvious).

From ONR-handle pair to SDP solution (continued)

- Form the matrix $\widetilde{M}$ as follows: take M and round down its main diagonal elements to -1 (the rounding value is $-1/(h^{\mathsf{T}}\rho_v)^2$), then take the negative of our matrix.

- $\widetilde{M}$ is a possible solution to our optimization problem (the symmetry of our matrix is obvious).

- We show that the value of the objective function ($\lambda_{\max}(\widetilde{M})$) cannot be greater than the Lovász parameter of the ONR-handle pair ($\mathrm{Lov}(\{\rho_v\}_{v\in V}, h)$).

From ONR-handle pair to SDP solution (continued)

- Form the matrix $\widetilde{M}$ as follows: take M and round down its main diagonal elements to -1 (the rounding value is $-1/(h^{\mathsf{T}}\rho_v)^2$), then take the negative of our matrix.
- $\widetilde{M}$ is a possible solution to our optimization problem (the symmetry of our matrix is obvious).
- We show that the value of the objective function $(\lambda_{\max}(\widetilde{M}))$ cannot be greater than the Lovász parameter of the ONR-handle pair $(\mathrm{Lov}(\{\rho_v\}_{v\in V}, h))$.
- To do this, consider the matrix $\mathrm{Lov}(\{\rho_v\}_{v\in V}, h)I - \widetilde{M}$.

From ONR-handle pair to SDP solution (continued)

- Form the matrix $\widetilde{M}$ as follows: take M and round down its main diagonal elements to -1 (the rounding value is $-1/(h^{\mathsf{T}}\rho_v)^2$), then take the negative of our matrix.

- $\widetilde{M}$ is a possible solution to our optimization problem (the symmetry of our matrix is obvious).

- We show that the value of the objective function ($\lambda_{\max}(\widetilde{M})$) cannot be greater than the Lovász parameter of the ONR-handle pair ($\mathrm{Lov}(\{\rho_v\}_{v\in V}, h)$).

- To do this, consider the matrix $\mathrm{Lov}(\{\rho_v\}_{v\in V}, h)I - \widetilde{M}$.

- We show that this is positive semidefinite, which proves our goal.

From ONR-handle pair to SDP solution (completion)

From ONR-handle pair to SDP solution (completion)

• The modification of M (the Gram matrix) on the main diagonal is done. The modification consists of adding the original diagonal matrix ($\mathrm{Lov}(\{\rho_v\}_{v\in V}, h)I$) and the result of rounding down, i.e., we add to the value on the diagonal of M $\mathrm{Lov}(\{\rho_v\}_{v\in V}, h) - 1/(h^{\mathsf{T}}\rho_v)^2$. This is adding a nonnegative number.

From ONR-handle pair to SDP solution (completion)

- The modification of M (the Gram matrix) on the main diagonal is done. The modification consists of adding the original diagonal matrix ($\mathrm{Lov}(\{\rho_v\}_{v\in V}, h)I$) and the result of rounding down, i.e., we add to the value on the diagonal of M $\mathrm{Lov}(\{\rho_v\}_{v\in V}, h) - 1/(h^\mathsf{T}\rho_v)^2$. This is adding a nonnegative number.

- So our matrix is $M + \Delta$. Where Δ is a diagonal matrix with nonnegative elements, specifically positive semidefinite. Furthermore, M is a Gram matrix, specifically positive semidefinite.

From ONR-handle pair to SDP solution (completion)

- The modification of M (the Gram matrix) on the main diagonal is done. The modification consists of adding the original diagonal matrix $(\text{Lov}(\{\rho_v\}_{v\in V}, h)I)$ and the result of rounding down, i.e., we add to the value on the diagonal of M $\text{Lov}(\{\rho_v\}_{v\in V}, h) - 1/(h^\mathsf{T}\rho_v)^2$. This is adding a nonnegative number.

- So our matrix is $M + \Delta$. Where Δ is a diagonal matrix with nonnegative elements, specifically positive semidefinite. Furthermore, M is a Gram matrix, specifically positive semidefinite.

- Therefore, our matrix is the sum of two positive semidefinite matrices, and thus, it is also one.

Lov(G) complexity

Lov(G) complexity

Theorem

The Lovász parameter of a given G can be computed in polynomial time.

Lov(G) complexity

Theorem

The Lovász parameter of a given G can be computed in polynomial time.

• We have seen that computing Lov(G) can be formulated as an SDP problem. Thus, it is a manageable task.

Break

Maximum Cut

Maximum Cut

- Given a weighted graph $w : E(G) \to \mathbb{R}_+$. Find a cut (S, T) in V such that

$$w(\mathcal{V}) = w(E(\mathcal{V})) = \sum_{e \in E(\mathcal{V})} w(e)$$

is maximized.

Maximum Cut

- Given a weighted graph $w : E(G) \to \mathbb{R}_+$. Find a cut (S, T) in V such that

$$w(\mathcal{V}) = w(E(\mathcal{V})) = \sum_{e \in E(\mathcal{V})} w(e)$$

is maximized.

- Where

$$E(\mathcal{V}) = \{e = xy \in E(G) : x \in S, y \in T \text{ or } x \in T, y \in S\}.$$

Maximum Cut

- Given a weighted graph $w : E(G) \to \mathbb{R}_+$. Find a cut (S, T) in V such that

$$w(\mathcal{V}) = w(E(\mathcal{V})) = \sum_{e \in E(\mathcal{V})} w(e)$$

is maximized.

- Where

$$E(\mathcal{V}) = \{e = xy \in E(G) : x \in S, y \in T \text{ or } x \in T, y \in S\}.$$

- It is known that the problem is $\mathcal{N}P$-hard, and finding an efficient solution seems hopeless.

Maximum Cut

- Given a weighted graph $w : E(G) \to \mathbb{R}_+$. Find a cut (S, T) in V such that

$$w(\mathcal{V}) = w(E(\mathcal{V})) = \sum_{e \in E(\mathcal{V})} w(e)$$

is maximized.

- Where

$$E(\mathcal{V}) = \{e = xy \in E(G) : x \in S, y \in T \text{ or } x \in T, y \in S\}.$$

- It is known that the problem is $\mathcal{N}P$-hard, and finding an efficient solution seems hopeless.

- Two trivial approximate algorithms are mentioned. Both are associated with Erdős Pál.

Greedy algorithm

Greedy algorithm

- Choose $e = xy$ of maximum weight, put x into S and y into T.

Greedy algorithm

- Choose $e = xy$ of maximum weight, put x into S and y into T.
- Examine the remaining vertices $v_3, v_4, \ldots, v_n$ sequentially:

$v_i \to S?/T?$:

Greedy algorithm

- Choose $e = xy$ of maximum weight, put x into S and y into T.
- Examine the remaining vertices $v_3, v_4, \ldots, v_n$ sequentially:

$v_i \to S?/T?$: assign v_i to the set where it achieves the greater increase.

Greedy algorithm

- Choose $e = xy$ of maximum weight, put x into S and y into T.
- Examine the remaining vertices $v_3, v_4, \ldots, v_n$ sequentially:

$v_i \to S?/T?$: assign v_i to the set where it achieves the greater increase.

Note

The cut $\mathcal{V} = (S, T)$ formed by the greedy algorithm satisfies:

$$w(\mathcal{V}) \geq \frac{1}{2} \sum w(e) \geq \frac{1}{2} w(E(G)).$$

Greedy algorithm

- Choose $e = xy$ of maximum weight, put x into S and y into T.
- Examine the remaining vertices $v_3, v_4, \dots, v_n$ sequentially:

$v_i \to S?/T?$: assign v_i to the set where it achieves the greater increase.

Note

The cut $\mathcal{V} = (S, T)$ formed by the greedy algorithm satisfies:

$$w(\mathcal{V}) \geq \frac{1}{2} \sum w(e) \geq \frac{1}{2} w(E(G)).$$

- Indeed. Assigning each vertex to one of the partitions modifies the sums $w(\mathcal{V})$ and $w(E(G) - E(\mathcal{V}))$. The greedy algorithm ensures that $w(\mathcal{V}) > w(E(G) - E(\mathcal{V}))$ holds initially and remains true.

Randomized algorithm

Randomized algorithm

- For each vertex $x \in V$, assign it to S with probability $\frac{1}{2}$, and to T with probability $\frac{1}{2}$ (decisions are independent for different vertices). Let $\underline{\mathcal{V}}$ be the resulting cut (a random variable).

Randomized algorithm

- For each vertex $x \in V$, assign it to S with probability $\frac{1}{2}$, and to T with probability $\frac{1}{2}$ (decisions are independent for different vertices). Let $\underline{\mathcal{V}}$ be the resulting cut (a random variable).
- Let

$$\xi_e = \begin{cases} 1, & \text{if } e \text{ has endpoints in different sets,} \\ 0, & \text{otherwise.} \end{cases}$$

Randomized algorithm

- For each vertex $x \in V$, assign it to S with probability $\frac{1}{2}$, and to T with probability $\frac{1}{2}$ (decisions are independent for different vertices). Let $\underline{\mathcal{V}}$ be the resulting cut (a random variable).
- Let

$$\xi_e = \begin{cases} 1, & \text{if } e \text{ has endpoints in different sets,} \\ 0, & \text{otherwise.} \end{cases}$$

- Then

$$w(\underline{\mathcal{V}}) = \sum \xi_e w_e,$$

and

$$\mathbb{E}(w(\underline{\mathcal{V}})) = \sum_{e \in E} w_e \mathbb{E}\xi_e = \frac{1}{2} \sum_{e \in E} w_e.$$

Hastad's theorem

Hastad's theorem

The other side of the coin is the following *negative* result:

Hastad's theorem

The other side of the coin is the following *negative* result:

Håstad

If there is a polynomial-time algorithm that computes a cut $((G, w) \mapsto \mathcal{V})$, such that $w(\mathcal{V}) \geq \frac{16}{17} w(\mathcal{V}_{\text{opt}})$, then $P = \mathcal{N}P$.

Hastad's theorem

The other side of the coin is the following *negative* result:

Håstad

If there is a polynomial-time algorithm that computes a cut $((G, w) \mapsto \mathcal{V})$, such that $w(\mathcal{V}) \geq \frac{16}{17} w(\mathcal{V}_{\text{opt}})$, then $P = \mathcal{N}P$.

After this, any improvement of the obvious (Erdős-type) algorithms represents significant progress:

Goemans—Williamson Theorem and Basic Idea

Goemans—Williamson Theorem and Basic Idea

(Goemans—Williamson, 1994)

There exists a randomized algorithm $((G, w) \to \mathcal{V})$, such that

$$\mathbb{E}(w(\mathcal{V})) \geq 0.8789 w(\mathcal{V}_{opt}).$$

Goemans—Williamson Theorem and Basic Idea

(Goemans—Williamson, 1994)

There exists a randomized algorithm $((G, w) \to \mathcal{V})$, such that

$$\mathbb{E}(w(\mathcal{V})) \geq 0.8789 w(\mathcal{V}_{opt}).$$

Goemans—Williamson Algorithm, Version 0

Goemans—Williamson Theorem and Basic Idea

(Goemans—Williamson, 1994)

There exists a randomized algorithm $((G, w) \to \mathcal{V})$, such that

$$\mathbb{E}(w(\mathcal{V})) \geq 0.8789 w(\mathcal{V}_{opt}).$$

Goemans—Williamson Algorithm, Version 0

(1) Select a vector representation $\rho : V(G) \to \mathbb{S}^{n-1} \subset \mathbb{R}^n$ for the vertices (where $n = |V(G)|$, $\mathbb{S}^{n-1} = \{x \in \mathbb{R}^n : x^\mathsf{T} x = 1\}$).

Goemans—Williamson Theorem and Basic Idea

(Goemans—Williamson, 1994)

There exists a randomized algorithm $((G,w) \to \mathcal{V})$, such that

$$\mathbb{E}(w(\mathcal{V})) \geq 0.8789 w(\mathcal{V}_{opt}).$$

Goemans—Williamson Algorithm, Version 0

(1) Select a vector representation $\rho : V(G) \to \mathbb{S}^{n-1} \subset \mathbb{R}^n$ for the vertices (where $n = |V(G)|$, $\mathbb{S}^{n-1} = \{x \in \mathbb{R}^n : x^{\mathsf{T}} x = 1\}$).

(2) Choose a random vector $\nu \in S^{n-1}$.

Goemans—Williamson Theorem and Basic Idea

(Goemans—Williamson, 1994)

There exists a randomized algorithm $((G, w) \to \mathcal{V})$, such that

$$\mathbb{E}(w(\mathcal{V})) \geq 0.8789 w(\mathcal{V}_{opt}).$$

Goemans—Williamson Algorithm, Version 0

(1) Select a vector representation $\rho : V(G) \to \mathbb{S}^{n-1} \subset \mathbb{R}^n$ for the vertices (where $n = |V(G)|$, $\mathbb{S}^{n-1} = \{x \in \mathbb{R}^n : x^\mathsf{T} x = 1\}$).

(2) Choose a random vector $\nu \in S^{n-1}$.

(3) Output: $S = \{v : \nu^\mathsf{T} \rho(v) < 0\}$, $T = \{v : \nu^\mathsf{T} \rho(v) > 0\}$.

Goemans—Williamson Theorem and Basic Idea

(Goemans—Williamson, 1994)

There exists a randomized algorithm $((G, w) \to \mathcal{V})$, such that

$$\mathbb{E}(w(\mathcal{V})) \geq 0.8789 w(\mathcal{V}_{opt}).$$

Goemans—Williamson Algorithm, Version 0

(1) Select a vector representation $\rho : V(G) \to \mathbb{S}^{n-1} \subset \mathbb{R}^n$ for the vertices (where $n = |V(G)|$, $\mathbb{S}^{n-1} = \{x \in \mathbb{R}^n : x^\mathsf{T} x = 1\}$).

(2) Choose a random vector $\nu \in S^{n-1}$.

(3) Output: $S = \{v : \nu^\mathsf{T} \rho(v) < 0\}$, $T = \{v : \nu^\mathsf{T} \rho(v) > 0\}$.

// $V(G) = S \dot\cup T$ with probability 1.

Questions

Questions

- The implementation of step (2) is a stochastic problem. Its solution is well known: generate ν's n components independently, each as a normally distributed random variable with mean 0 and standard deviation 1, and normalize to unit vectors.

Questions

- The implementation of step (2) is a stochastic problem. Its solution is well known: generate ν's n components independently, each as a normally distributed random variable with mean 0 and standard deviation 1, and normalize to unit vectors.
- How do we choose ρ in step (1)? We highlight three possibilities.

Questions

- The implementation of step (2) is a stochastic problem. Its solution is well known: generate ν's n components independently, each as a normally distributed random variable with mean 0 and standard deviation 1, and normalize to unit vectors.
- How do we choose ρ in step (1)? We highlight three possibilities.
  - If we knew the optimal (S, T) cut, then $\rho|_S : x \mapsto e$, $\rho|_T : x \mapsto -e$ would lead to a *non-computable* vector representation for the optimal cut.

Questions

- The implementation of step (2) is a stochastic problem. Its solution is well known: generate ν's n components independently, each as a normally distributed random variable with mean 0 and standard deviation 1, and normalize to unit vectors.
- How do we choose ρ in step (1)? We highlight three possibilities.
  - If we knew the optimal (S, T) cut, then $\rho|_S : x \mapsto e$, $\rho|_T : x \mapsto -e$ would lead to a *non-computable* vector representation for the optimal cut.
  - If ρ is random, then we recover Erdős's random algorithm.

Questions

- The implementation of step (2) is a stochastic problem. Its solution is well known: generate ν's n components independently, each as a normally distributed random variable with mean 0 and standard deviation 1, and normalize to unit vectors.
- How do we choose ρ in step (1)? We highlight three possibilities.
  - If we knew the optimal (S, T) cut, then $\rho|_S : x \mapsto e$, $\rho|_T : x \mapsto -e$ would lead to a *non-computable* vector representation for the optimal cut.
  - If ρ is random, then we recover Erdős's random algorithm.
  - Use a *computable* algorithm to determine a *clever* vector representation.

Questions

- The implementation of step (2) is a stochastic problem. Its solution is well known: generate ν's n components independently, each as a normally distributed random variable with mean 0 and standard deviation 1, and normalize to unit vectors.
- How do we choose ρ in step (1)? We highlight three possibilities.
  - If we knew the optimal (S, T) cut, then $\rho|_S : x \mapsto e$, $\rho|_T : x \mapsto -e$ would lead to a *non-computable* vector representation for the optimal cut.
  - If ρ is random, then we recover Erdős's random algorithm.
  - Use a *computable* algorithm to determine a *clever* vector representation.

It is obvious that the third way is the feasible one. Its realization is the *essence* of the Goemans—Williamson algorithm.

What Do We Expect from Our Procedure's Output?

What Do We Expect from Our Procedure's Output?

Let $e = xy \in E$

$$\xi_e = \begin{cases} 1, & \text{if } x \text{ and } y \text{ fall into different classes,} \\ 0, & \text{otherwise,} \end{cases}$$

and let α be the angle between the vectors ρ_x and ρ_y.

What Do We Expect from Our Procedure's Output?

Let $e = xy \in E$

$$\xi_e = \begin{cases} 1, & \text{if } x \text{ and } y \text{ fall into different classes,} \\ 0, & \text{otherwise,} \end{cases}$$

and let α be the angle between the vectors ρ_x and ρ_y.

Then

$$\mathbb{E}\xi_e = \mathbb{P}(\xi_e = 1) = \frac{2\alpha}{2\pi} = \frac{\alpha}{\pi} = \frac{\arccos \rho_x^T \rho_y}{\pi}.$$

What Do We Expect from Our Procedure's Output?

Let $e = xy \in E$

$$\xi_e = \begin{cases} 1, & \text{if } x \text{ and } y \text{ fall into different classes,} \\ 0, & \text{otherwise,} \end{cases}$$

and let α be the angle between the vectors ρ_x and ρ_y.

Then

$$\mathbb{E}\xi_e = \mathbb{P}(\xi_e = 1) = \frac{2\alpha}{2\pi} = \frac{\alpha}{\pi} = \frac{\arccos \rho_x^{\mathsf{T}} \rho_y}{\pi}.$$

Consequence

$$\mathbb{E}w(\mathcal{V}) = \sum_{e=xy\in E} w_e \frac{\arccos \rho_x^{\mathsf{T}} \rho_y}{\pi}.$$

What Do We Expect from Our Procedure's Output?

Let $e = xy \in E$

$$\xi_e = \begin{cases} 1, & \text{if } x \text{ and } y \text{ fall into different classes,} \\ 0, & \text{otherwise,} \end{cases}$$

and let α be the angle between the vectors ρ_x and ρ_y.

Then

$$\mathbb{E}\xi_e = \mathbb{P}(\xi_e = 1) = \frac{2\alpha}{2\pi} = \frac{\alpha}{\pi} = \frac{\arccos \rho_x^T \rho_y}{\pi}.$$

Consequence

$$\mathbb{E}w(\mathcal{V}) = \sum_{e=xy\in E} w_e \frac{\arccos \rho_x^T \rho_y}{\pi}.$$

If our goal was to determine a ρ where this expected value is maximized, then we would face a too difficult problem.

Elementary Analysis

Elementary Analysis

Lemma

$$\frac{1}{\pi}\arccos x \geq 0.87856 \cdot \frac{1}{2}(1-x).$$

Elementary Analysis

Lemma

$$\frac{1}{\pi}\arccos x \geq 0.87856 \cdot \frac{1}{2}(1-x).$$

The lemma is a simple calculus exercise. We leave its verification, calculation to the interested student.

Elementary Analysis

Lemma

$$\frac{1}{\pi}\arccos x \geq 0.87856 \cdot \frac{1}{2}(1-x).$$

The lemma is a simple calculus exercise. We leave its verification, calculation to the interested student.

Consequence

$$\mathbb{E}(w(\mathcal{V})) \geq 0.87856 \sum_{e=xy\in E} w(e)\frac{1}{2}(1-\rho_x^{\mathsf{T}}\rho_y).$$

Elementary Analysis

Lemma

$$\frac{1}{\pi}\arccos x \geq 0.87856 \cdot \frac{1}{2}(1-x).$$

The lemma is a simple calculus exercise. We leave its verification, calculation to the interested student.

Consequence

$$\mathbb{E}(w(\mathcal{V})) \geq 0.87856 \sum_{e=xy\in E} w(e)\frac{1}{2}(1-\rho_x^{\mathsf{T}}\rho_y).$$

Now we can designate our goal: Let's take a ρ where the sum appearing in the lower bound above is maximized.

Our Goal as an SDP Problem

Our Goal as an SDP Problem

- The sought-after ρ vectors can be expressed based on the inner product matrix M or Gram matrix.

Our Goal as an SDP Problem

- The sought-after ρ vectors can be expressed based on the inner product matrix M or Gram matrix.

- This is a positive semidefinite matrix. The desired optimization problem is a semidefinite optimization problem, manageable:

Maximize	$\frac{1}{2}\langle W, (1 - M)\rangle$-t
subject to	$M_{vv} = 1$, for all $v \in V$
	$M \succeq 0$,

where W is the matrix describing the weights, i.e., replacing the 1's in the adjacency matrix with the corresponding edge weights.

Our Goal as an SDP Problem

- The sought-after ρ vectors can be expressed based on the inner product matrix M or Gram matrix.

- This is a positive semidefinite matrix. The desired optimization problem is a semidefinite optimization problem, manageable:

Maximize	$\frac{1}{2}\langle W, (1-M)\rangle$-t
subject to	$M_{vv} = 1$, for all $v \in V$
	$M \succeq 0$,

where W is the matrix describing the weights, i.e., replacing the 1's in the adjacency matrix with the corresponding edge weights.

- The solution to this optimization problem yields a Gram matrix M.

Our Goal as an SDP Problem

- The sought-after ρ vectors can be expressed based on the inner product matrix M or Gram matrix.

- This is a positive semidefinite matrix. The desired optimization problem is a semidefinite optimization problem, manageable:

Maximize	$\frac{1}{2}\langle W, (1-M)\rangle$-t
subject to	$M_{vv} = 1$, for all $v \in V$
	$M \succeq 0$,

where W is the matrix describing the weights, i.e., replacing the 1's in the adjacency matrix with the corresponding edge weights.

- The solution to this optimization problem yields a Gram matrix M.

- From this, we can calculate a system of unit vectors $\{\rho_v\}_{v\in V}$ corresponding to this, i.e., a vector representation for the vertices of our graph.

Our Goal as an SDP Problem

- The sought-after ρ vectors can be expressed based on the inner product matrix M or Gram matrix.
- This is a positive semidefinite matrix. The desired optimization problem is a semidefinite optimization problem, manageable:

Maximize	$\frac{1}{2}\langle W, (1-M)\rangle$-t
subject to	$M_{vv} = 1$, for all $v \in V$
	$M \succeq 0$,

where W is the matrix describing the weights, i.e., replacing the 1's in the adjacency matrix with the corresponding edge weights.

- The solution to this optimization problem yields a Gram matrix M.
- From this, we can calculate a system of unit vectors $\{\rho_v\}_{v\in V}$ corresponding to this, i.e., a vector representation for the vertices of our graph.

Analysis of the GW Algorithm

Analysis of the GW Algorithm

With this, the description of the algorithm is complete. The analysis based on our previous observations can be easily put together:

Analysis of the GW Algorithm

With this, the description of the algorithm is complete. The analysis based on our previous observations can be easily put together:

Theorem

Let $\mathcal{V}_{GW}$ be the cut computed by the algorithm. Then

$$\mathbb{E}(w(\mathcal{V}_{GW})) \geq 0.87856 \cdot w(\mathcal{V}_{opt}).$$

Analysis of the GW Algorithm

With this, the description of the algorithm is complete. The analysis based on our previous observations can be easily put together:

Theorem

Let $\mathcal{V}_{GW}$ be the cut computed by the algorithm. Then

$$\mathbb{E}(w(\mathcal{V}_{GW})) \geq 0.87856 \cdot w(\mathcal{V}_{opt}).$$

Proof:

$$\begin{aligned}\mathbb{E}(w(\mathcal{V}_{GW})) = \sum_{e \in E} w_e \frac{\arccos \rho_x^{\mathsf{T}} \rho_y}{\pi} \geq 0.87856 \sum w(e) \frac{1}{2}(1 - \rho_x^{\mathsf{T}} \rho_y) \\ = 0.87856 \cdot p^* \geq 0.87856 w(\mathcal{V}_{opt}),\end{aligned}$$

where $\mathcal{V}_{GW}$ is the Goemans—Williamson choice, and $\mathcal{V}_{opt}$ is the (unknown) optimal cut, but one possible solution to the optimization problem we are considering.

Break

Coloring Problems

Coloring Problems

- Given a graph G. Can it be colored with 2 colors? If yes, then color it properly with 2 colors.

Coloring Problems

- Given a graph G. Can it be colored with 2 colors? If yes, then color it properly with 2 colors.
- This problem is easily solvable based on a BSc Combinatorics course.

Coloring Problems

- Given a graph G. Can it be colored with 2 colors? If yes, then color it properly with 2 colors.
- This problem is easily solvable based on a BSc Combinatorics course.
- Given a graph G, can it be colored with 3 colors?

Coloring Problems

- Given a graph G. Can it be colored with 2 colors? If yes, then color it properly with 2 colors.

- This problem is easily solvable based on a BSc Combinatorics course.

- Given a graph G, can it be colored with 3 colors?

- This problem is $\mathcal{NP}$-complete. According to current scientific knowledge, it's considered to be hopelessly difficult.

The Basic Question

The Basic Question

The Basic Question

Let's consider a relaxed problem: Given a graph G, we know that $\chi(G) = 3$, i.e., it's guaranteed to be 3-colorable. Color it with as few colors as possible.

The Basic Question

The Basic Question

Let's consider a relaxed problem: Given a graph G, we know that $\chi(G) = 3$, i.e., it's guaranteed to be 3-colorable. Color it with as few colors as possible.

The relaxed problem proves to be difficult as well. It still remains at the forefront of research.

The Initial Algorithm

The Initial Algorithm

Let's look at the basic algorithm from which everything starts.

Wigderson's Algorithm

1. case: If for every x vertex $d(x) \le \tau = \sqrt{n}$, then color it greedily.

// Each degree is at most $\sqrt{n}$, so the color requirement is at most $\sqrt{n} + 1$.

2. case: If there exists a vertex x such that $d(x) > \tau = \sqrt{n}$, then

// Let N be the set of neighbors of x.

// $G|_N$ is bipartite, since G is 3-colorable.

- $G|_N$ can be properly colored with 2 colors.
- $G \leftarrow G - N$

// "Bite off" N.

- Return to the beginning of the algorithm.

Analysis of Wigderson's Algorithm

Analysis of Wigderson's Algorithm

The analysis of the algorithm is straightforward:

Analysis of Wigderson's Algorithm

The analysis of the algorithm is straightforward:

Lemma

The color requirement of Wigderson's Algorithm is at most $3\sqrt{n}+1$.

Analysis of Wigderson's Algorithm

The analysis of the algorithm is straightforward:

Lemma

The color requirement of Wigderson's Algorithm is at most $3\sqrt{n}+1$.

- Indeed, each bite reduces the number of vertices by at least $\sqrt{n}$.

Analysis of Wigderson's Algorithm

The analysis of the algorithm is straightforward:

Lemma

The color requirement of Wigderson's Algorithm is at most $3\sqrt{n}+1$.

- Indeed, each bite reduces the number of vertices by at least $\sqrt{n}$.
- So there can be at most $\sqrt{n}$ bites, each using two new colors.

Analysis of Wigderson's Algorithm

The analysis of the algorithm is straightforward:

Lemma

The color requirement of Wigderson's Algorithm is at most $3\sqrt{n}+1$.

- Indeed, each bite reduces the number of vertices by at least $\sqrt{n}$.
- So there can be at most $\sqrt{n}$ bites, each using two new colors.
- After the bites, everything can be colored with at most $\sqrt{n}+1$ colors.

Direction of Improvement

Direction of Improvement

- It's easy to see that the distinguishing parameter τ between the two significantly different cases can be chosen more cleverly, but the order of magnitude of the color requirement $\sqrt{n}$ does not improve.

Direction of Improvement

- It's easy to see that the distinguishing parameter τ between the two significantly different cases can be chosen more cleverly, but the order of magnitude of the color requirement $\sqrt{n}$ does not improve.
- Our later algorithm uses a similar structure. For the greedy coloring, it employs a smarter method.

Direction of Improvement

• It's easy to see that the distinguishing parameter τ between the two significantly different cases can be chosen more cleverly, but the order of magnitude of the color requirement $\sqrt{n}$ does not improve.

• Our later algorithm uses a similar structure. For the greedy coloring, it employs a smarter method.

• Thus, with a better τ distinguishing parameter, we work with a better (expected) color requirement for our algorithm.

The Structure of Improvement

The Structure of Improvement

- The parameters of the coloring algorithm that replaces the greedy algorithm are summarized by the following theorem.

The Structure of Improvement

- The parameters of the coloring algorithm that replaces the greedy algorithm are summarized by the following theorem.
- In fact, this describes a step towards the creation of a complete coloring.

The Structure of Improvement

- The parameters of the coloring algorithm that replaces the greedy algorithm are summarized by the following theorem.

- In fact, this describes a step towards the creation of a complete coloring.

- It calculates a *partial coloring*, where at least half of the vertices receive a color (in a proper way), but there is also the possibility to leave one vertex uncolored (no more than half of the vertices).

The Structure of Improvement

- The parameters of the coloring algorithm that replaces the greedy algorithm are summarized by the following theorem.

- In fact, this describes a step towards the creation of a complete coloring.

- It calculates a *partial coloring*, where at least half of the vertices receive a color (in a proper way), but there is also the possibility to leave one vertex uncolored (no more than half of the vertices).

- To achieve a good coloring, this process must be iterated on the remaining uncolored vertices. After $\log n$ iterations, we obtain a well-colored graph with a color requirement that is $\log n$ times the color requirement stated in the theorem.

Karger—Motwani—Sudan Theorem

Karger—Motwani—Sudan Theorem

Karger—Motwani—Sudan Theorem

There exists a randomized algorithm that *knows*: If given a 3-colorable graph G with no degree greater than τ, then the algorithm computes a *good partial coloring*, with a color requirement of $\mathcal{O}(\tau^{0.632})$. The expected running time of the algorithm is polynomial.

Karger—Motwani—Sudan Theorem

Karger—Motwani—Sudan Theorem

There exists a randomized algorithm that *knows*: If given a 3-colorable graph G with no degree greater than τ, then the algorithm computes a *good partial coloring*, with a color requirement of $\mathcal{O}(\tau^{0.632})$. The expected running time of the algorithm is polynomial.

The proof is an algorithm. Once again, instead of assigning colors to vertices, vectors are assigned to them.

Karger—Motwani—Sudan Algorithm

Karger—Motwani—Sudan Partial Coloring Algorithm

Karger—Motwani—Sudan Algorithm

Karger—Motwani—Sudan Partial Coloring Algorithm

(1) Choose a *smart* vector representation $\rho : V \to \mathbb{S}^{n-1}$.

Karger—Motwani—Sudan Algorithm

Karger—Motwani—Sudan Partial Coloring Algorithm

(1) Choose a *smart* vector representation $\rho : V \to \mathbb{S}^{n-1}$.

(2) Choose independently $\nu_1, \nu_2, \ldots, \nu_e \in \mathbb{S}^{n-1}$ as random independent unit vectors/directions.

Karger—Motwani—Sudan Algorithm

Karger—Motwani—Sudan Partial Coloring Algorithm

(1) Choose a *smart* vector representation $\rho : V \to \mathbb{S}^{n-1}$.

(2) Choose independently $\nu_1, \nu_2, \ldots, \nu_e \in \mathbb{S}^{n-1}$ as random independent unit vectors/directions.

(2a) Let $v \mapsto (\mathrm{sign}(\nu_i^{\mathsf{T}} \rho(v)))_{i=1}^{\ell}$, where

$$\mathrm{sign}(x) = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{if } x = 0, \\ -1, & \text{if } x < 0. \end{cases}$$

// The probability of the 0 component is 0, there are 2^{ℓ} possible outcomes/colors.

Karger—Motwani—Sudan Algorithm (continued)

Karger—Motwani—Sudan Partial Coloring Algorithm

Karger—Motwani—Sudan Algorithm (continued)

Karger—Motwani—Sudan Partial Coloring Algorithm

(2b) Select the improperly colored edges and remove the color from one endpoint. This yields a good partial coloring.

Karger—Motwani—Sudan Algorithm (continued)

Karger—Motwani—Sudan Partial Coloring Algorithm

(2b) Select the improperly colored edges and remove the color from one endpoint. This yields a good partial coloring.

(2c) If at least half of the vertices are colored, then STOP. If fewer than half of the vertices remain colored, return to step (2).

The Basic Idea Behind the Choice

The Basic Idea Behind the Choice

- The essential question again is the choice of ρ in point (1), the good/smart vector representation.

The Basic Idea Behind the Choice

- The essential question again is the choice of ρ in point (1), the good/smart vector representation.
- Let

$$\xi_e = \begin{cases} 1, & \text{if edge } xy \text{ is improperly colored,} \\ 0, & \text{otherwise.} \end{cases}$$

The Basic Idea Behind the Choice

- The essential question again is the choice of ρ in point (1), the good/smart vector representation.
- Let

$$\xi_e = \begin{cases} 1, & \text{if edge } xy \text{ is improperly colored,} \\ 0, & \text{otherwise.} \end{cases}$$

- What is its expected value?

$$\mathbb{E}\xi_e = \mathbb{P}(xy \text{ is improperly colored}) = \left(1 - \frac{\arccos \rho_x^{\mathsf{T}} \rho_y}{\pi}\right)^{\ell}.$$

The Goal of the Choice

The Goal of the Choice

- **The goal:** Choose ρ such that for every edge xy,

$$\mathbb{E}\xi_e = \mathbb{P}(xy \text{ is improperly colored}) = \left(1 - \frac{\arccos \rho_x^\mathsf{T} \rho_y}{\pi}\right)^\ell.$$

is *small*.

The Goal of the Choice

- **The goal:** Choose ρ such that for every edge xy,

$$\mathbb{E}\xi_e = \mathbb{P}(xy \text{ is improperly colored}) = \left(1 - \frac{\arccos \rho_x^{\mathsf{T}} \rho_y}{\pi}\right)^{\ell}.$$

is *small*.

- That is, choose ρ such that for every edge xy,

$$\frac{\arccos \rho_y^{\mathsf{T}} \rho_y}{\pi}$$

is *large*.

The Goal of the Choice

- **The goal:** Choose ρ such that for every edge xy,

$$\mathbb{E}\xi_e = \mathbb{P}(xy \text{ is improperly colored}) = \left(1 - \frac{\arccos \rho_x^{\mathsf{T}} \rho_y}{\pi}\right)^{\ell}.$$

is *small*.

- That is, choose ρ such that for every edge xy,

$$\frac{\arccos \rho_y^{\mathsf{T}} \rho_y}{\pi}$$

is *large*.

- That is, choose ρ such that for every edge xy,

$$\rho_x^{\mathsf{T}} \rho_y$$

is *small*.

The Precise Choice

The Precise Choice

- Refinement of point (1) of the algorithm: Choose ρ as the optimal solution vector system derived from the following SDP problem from an optimal matrix $G \in \mathbb{R}^{V \times V}$:

The Precise Choice

- Refinement of point (1) of the algorithm: Choose ρ as the optimal solution vector system derived from the following SDP problem from an optimal matrix $G \in \mathbb{R}^{V \times V}$:

Minimize	μ-t
subject to	$G \succeq 0$,
	$G_{uu} = 1$ for every vertex u,
	$G_{uv} \leq \mu$ for every edge $uv \in E$.

The Precise Choice

- Refinement of point (1) of the algorithm: Choose ρ as the optimal solution vector system derived from the following SDP problem from an optimal matrix $G \in \mathbb{R}^{V \times V}$:

Minimize	μ-t
subject to	$G \succeq 0$,
	$G_{uu} = 1$ for every vertex u,
	$G_{uv} \leq \mu$ for every edge $uv \in E$.

- Solving this provides an optimal value p^* and an optimal location G (optimal Gram matrix). From this, unit vectors ($G_{uu} = 1$) corresponding to the vertices of our graph can be derived, providing a vector representation for the vertices of the graph.

The Precise Choice

• Refinement of point (1) of the algorithm: Choose ρ as the optimal solution vector system derived from the following SDP problem from an optimal matrix $G \in \mathbb{R}^{V \times V}$:

Minimize	μ-t
subject to	$G \succeq 0$,
	$G_{uu} = 1$ for every vertex u,
	$G_{uv} \leq \mu$ for every edge $uv \in E$.

• Solving this provides an optimal value p^* and an optimal location G (optimal Gram matrix). From this, unit vectors ($G_{uu} = 1$) corresponding to the vertices of our graph can be derived, providing a vector representation for the vertices of the graph.

• This is the precise description of step (1) of the Karger—Motwani—Sudan partial coloring algorithm.

Analysis of the Algorithm

Analysis of the Algorithm

- The analysis of the algorithm is straightforward:

Analysis of the Algorithm

- The analysis of the algorithm is straightforward:
- First, let's estimate the value of p^*.

Analysis of the Algorithm

- The analysis of the algorithm is straightforward:
- First, let's estimate the value of p^*.
- To do this, let's take an optimization problem solution as follows: for a good $c : V(G) \to \{1, 2, 3\}$-coloring of G, let $\rho_v = e_{c(v)}$, where e_1, e_2, e_3 are three unit vectors pointing to the vertices of a regular triangle in the plane.

Analysis of the Algorithm

- The analysis of the algorithm is straightforward:
- First, let's estimate the value of p^*.
- To do this, let's take an optimization problem solution as follows: for a good $c : V(G) \to \{1, 2, 3\}$-coloring of G, let $\rho_v = e_{c(v)}$, where e_1, e_2, e_3 are three unit vectors pointing to the vertices of a regular triangle in the plane.
- Then the value of the objective function is $2\pi/3$, so $p^* \leq -1/2$, meaning that $\arccos p^* \geq \arccos(-1/2) = 2\pi/3$.

Analysis of the Algorithm

- The analysis of the algorithm is straightforward:
- First, let's estimate the value of p^*.
- To do this, let's take an optimization problem solution as follows: for a good $c : V(G) \to \{1,2,3\}$-coloring of G, let $\rho_v = e_{c(v)}$, where e_1, e_2, e_3 are three unit vectors pointing to the vertices of a regular triangle in the plane.
- Then the value of the objective function is $2\pi/3$, so $p^* \leq -1/2$, meaning that $\arccos p^* \geq \arccos(-1/2) = 2\pi/3$.
- From this, we can refine our estimate of the expected value of the degree of improper coloring:

$$\mathbb{P}(\xi_e) = \left(1 - \frac{1}{\pi}\arccos(\rho_u^\mathsf{T}\rho_v)\right)^\ell \leq \left(\frac{1}{3}\right)^\ell = \frac{1}{9\tau},$$

if we choose ℓ so that $(1/3)^\ell = 1/9\tau$.

Analysis (continued)

Analysis (continued)

- Let $\mathcal{E}_{\text{improper}}$ be the number of edges improperly colored by the first random coloring.

Analysis (continued)

- Let $\mathcal{E}_{\text{improper}}$ be the number of edges improperly colored by the first random coloring.
- Then
$$\mathbb{E}(\mathcal{E}_{\text{improper}}) \leq \frac{1}{9\tau}|E| \leq \frac{1}{9\tau}\frac{|V|\tau}{2} = \frac{|V|}{18}.$$

Analysis (continued)

- Let $\mathcal{E}_{\text{improper}}$ be the number of edges improperly colored by the first random coloring.
- Then
$$\mathbb{E}(\mathcal{E}_{\text{improper}}) \le \frac{1}{9\tau}|E| \le \frac{1}{9\tau}\frac{|V|\tau}{2} = \frac{|V|}{18}.$$
- So, by the Markov inequality, the probability is small that one coloring does not find the output.

Analysis (continued)

- Let $\mathcal{E}_{\text{improper}}$ be the number of edges improperly colored by the first random coloring.
- Then
$$\mathbb{E}(\mathcal{E}_{\text{improper}}) \leq \frac{1}{9\tau}|E| \leq \frac{1}{9\tau}\frac{|V|\tau}{2} = \frac{|V|}{18}.$$
- So, by the Markov inequality, the probability is small that one coloring does not find the output.
- The expected value of the number of repetitions of colorings can be easily estimated.

Analysis (continued)

- Let $\mathcal{E}_{\text{improper}}$ be the number of edges improperly colored by the first random coloring.
- Then
$$\mathbb{E}(\mathcal{E}_{\text{improper}}) \leq \frac{1}{9\tau}|E| \leq \frac{1}{9\tau}\frac{|V|\tau}{2} = \frac{|V|}{18}.$$
- So, by the Markov inequality, the probability is small that one coloring does not find the output.
- The expected value of the number of repetitions of colorings can be easily estimated.
- With the choice of ℓ, the color requirement of 2^ℓ is $\mathcal{O}(\tau^{0.632})$, which yields the theorem.

Putting the Details Together

Putting the Details Together

- The rest is only outlined:

Putting the Details Together

- The rest is only outlined:

- The iteration of the partial coloring algorithm provides a good coloring algorithm, whose dependence on τ is better than that of the greedy algorithm.

Putting the Details Together

- The rest is only outlined:
- The iteration of the partial coloring algorithm provides a good coloring algorithm, whose dependence on τ is better than that of the greedy algorithm.
- Thus, by working with this instead of the greedy algorithm in the Wigderson scheme, we get a better procedure.

Putting the Details Together

- The rest is only outlined:
- The iteration of the partial coloring algorithm provides a good coloring algorithm, whose dependence on τ is better than that of the greedy algorithm.
- Thus, by working with this instead of the greedy algorithm in the Wigderson scheme, we get a better procedure.
- Only the final result is stated.

Karger—Motwani—Sudan Coloring Algorithm

The Las Vegas algorithm described above properly colors a 3-colorable graph with n vertices using $\mathcal{O}(n^{0.39} \cdot \log n)$ colors.

Putting the Details Together

- The rest is only outlined:
- The iteration of the partial coloring algorithm provides a good coloring algorithm, whose dependence on τ is better than that of the greedy algorithm.
- Thus, by working with this instead of the greedy algorithm in the Wigderson scheme, we get a better procedure.
- Only the final result is stated.

Karger—Motwani—Sudan Coloring Algorithm

The Las Vegas algorithm described above properly colors a 3-colorable graph with n vertices using $\mathcal{O}(n^{0.39} \cdot \log n)$ colors.

- There are further refinements, but this is all we have time for.

This is the End!

Thank you for your attention!