Tensors and Graphs I: structures

Training Workshop at Tensors: Algebra-Geometry-Applications

Youming Qiao Youming.Qiao@uts.edu.au
University of Technology Sydney

29 May 2025

Some opening lines

* This is Youming dialling in from Sydney, Australia

* Thanks go to the organisers and you for this opportunity!

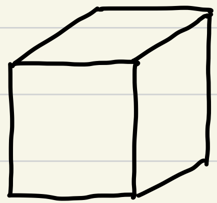
* We will explore some interesting connections between tensors and graphs

* Two lectures: first one about structures, second one about techniques

* Some of the materials will be useful for tensor isomorphism :)
  - Watch out for Xiaorui's talk next week.

Tensors and graphs: an unexpected match?

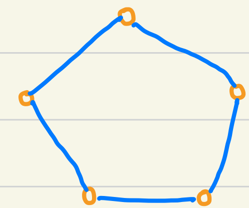
Tensors



* Multi-way arrays
* Hypermatrices
* Multilinear algebra
* Determinantal varieties
* Bilinear maps

...

Graphs



* Objects and relations
* Combinatorics
* Networks
* Graph colouring
* Extremal and probabilistic questions

...

Matrices of linear forms: where graphs and tensors meet

* You've (probably) seen matrices of linear forms
  - One way to encode 3-tensors

e.g.
$$\begin{bmatrix} 3x_1 - 2x_2 & -x_1 + x_2 - x_3 \\ 5x_2 + 3x_3 & x_2 - \frac{1}{2}x_3 \end{bmatrix}$$

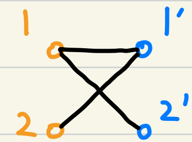
Matrices of linear forms: where graphs and tensors meet

* You've (probably) seen matrices of linear forms
  - One way to encode 3-tensors

e.g.
$$\begin{bmatrix} 3x_1 - 2x_2 & -x_1 + x_2 - x_3 \\ 5x_2 + 3x_3 & x_2 - \frac{1}{2}x_3 \end{bmatrix}$$

* From a graph, we can build special matrices of linear forms
  - From Tutte and Lovász

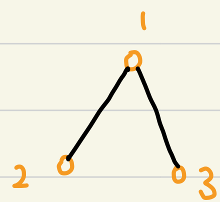
e.g.



$$\Rightarrow \begin{array}{c|c} & \begin{array}{cc} 1' & 2' \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & 0 \end{bmatrix} \end{array}$$

(1) Bipartite: each variable appears in (at most) one position;

(2) Undirected simple: each variable appears in (at most) two positions.



$$\Rightarrow \begin{array}{c|c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 0 & x_{12} & x_{13} \\ -x_{12} & 0 & 0 \\ -x_{13} & 0 & 0 \end{bmatrix} \end{array}$$

From graphs to tensors: the formal recipe

* $[n] := \{1, 2, \cdots, n\}$

$[m'] := \{1', 2', \cdots, m'\}$

$X := \{x_{11}, x_{12}, \cdots, x_{nm}\}$

* $G = ([n] \uplus [m'], E)$: bipartite graph $E \subseteq [n] \times [m']$

From graphs to tensors: the formal recipe

* $[n] := \{1, 2, \cdots, n\}$

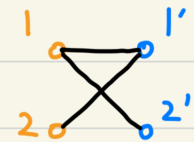
  $[m'] := \{1', 2', \cdots, m'\}$

  $X := \{x_{11}, x_{12}, \cdots, x_{nm}\}$

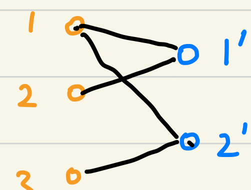
* $G = ([n] \uplus [m'], E)$: bipartite graph $\quad E \subseteq [n] \times [m']$

* $$M_G = \begin{array}{cc} & j \\ i & \left[\;\bigcirc\;\right]_{n\times m} \end{array}$$

$$\bigcirc = \begin{cases} x_{ij} & \text{if } (i,j') \in E \\ 0 & \text{o/w} \end{cases}$$



$$\Rightarrow \begin{array}{ccc} & 1' & 2' \\ 1 & \multicolumn{2}{c}{\multirow{2}{*}{$\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & 0 \end{bmatrix}$}} \\ 2 & & \end{array}$$



$$\Rightarrow \begin{array}{c|cc} & 1' & 2' \\ \hline 1 & x_{11} & x_{12} \\ 2 & x_{21} & 0 \\ 3 & 0 & x_{32} \end{array}$$

From graphs to tensors: the formal recipe

* $[n] := \{1, 2, \cdots, n\}$

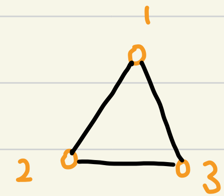
  $X := \{ x_{ij} \mid 1 \le i < j \le n \}$

* $G = ([n], E)$: simple undirected graph. $E \subseteq \binom{[n]}{2}$ → the set of subsets of size 2 of $[n]$

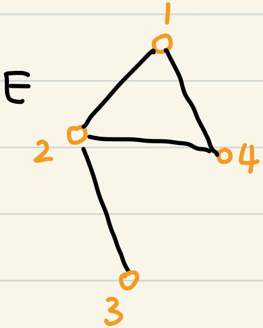
* $$M_G = \begin{array}{c|cc} & i & j \\ \hline i & & \circ \\ j & \circ & \end{array}_{n \times n}$$

  (i,j): $= \begin{cases} x_{ij} & \text{if } \{i,j\} \in E \\ 0 & \text{o/w} \end{cases}$

  (j,i): $= \begin{cases} -x_{ij} & \text{if } \{i,j\} \in E \\ 0 & \text{o/w} \end{cases}$



$$\Rightarrow \quad \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & x_{11} & x_{12} \\ 2 & -x_{11} & 0 & x_{13} \\ 3 & -x_{12} & -x_{13} & 0 \end{array}$$



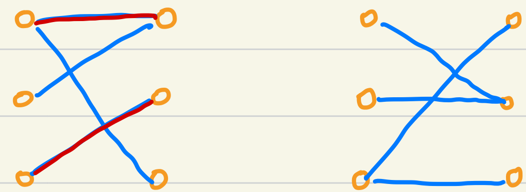
$$\Rightarrow \quad \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & x_{12} & 0 & x_{14} \\ 2 & -x_{12} & 0 & x_{23} & x_{24} \\ 3 & 0 & -x_{23} & 0 & 0 \\ 4 & -x_{14} & -x_{24} & 0 & 0 \end{array}$$

Graph structures versus tensor structures

* We've transformed graphs to tensors. But why bother to do that?

* Let's go back to ask Tutte and Lovász :)

* Graph matching: a set of disjoint edges



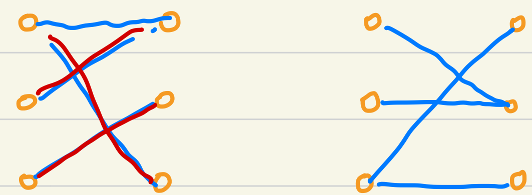
Graph structures versus tensor structures

* We've transformed graphs to tensors. But why bother to do that?

* Let's go back to ask Tutte and Lovász :)

* Graph matching: a set of disjoint edges

* Structural and algorithmic questions about perfect and maximum matchings



- Tutte: how to characterise graphs without perfect matchings?
- Lovász: how to efficiently decide if a graph has a perfect matching '?z

Graph structures versus tensor structures

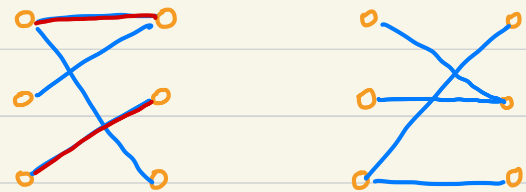
* We've transformed graphs to tensors. But why bother to do that?

* Let's go back to ask Tutte and Lovász :)

* Graph matching: a set of disjoint edges

* Structural and algorithmic questions about perfect and maximum matchings

* Turns out that the tensor viewpoint is helpful!



- Tutte: how to characterise graphs without perfect matchings?
- Lovász: how to efficiently decide if a graph has a perfect matching

An algebraic algorithm for bipartite graph perfect matching

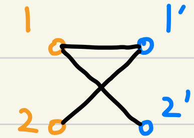
* $[n] := \{1, 2, \cdots, n\}$

  $X := \{x_{11}, x_{12}, \cdots, x_{nn}\}$

* $G = ([n] \uplus [n'], E)$: bipartite graph $E \subseteq [n] \times [n']$

* $$M_G = \begin{array}{cc} & j \\ i & \begin{bmatrix} & & \\ & \bigcirc & \\ & & \end{bmatrix}_{n \times n} \end{array}$$

$$\bigcirc = \begin{cases} x_{ij} & \text{if } (i, j') \in E \\ 0 & \text{o/w} \end{cases}$$



$$\Rightarrow \begin{array}{ccc} & 1' & 2' \\ 1 & \left[\begin{matrix} x_{11} \\ x_{21} \end{matrix}\right. & \left.\begin{matrix} x_{12} \\ 0 \end{matrix}\right] \\ 2 & & \end{array}$$

Obs. G has a perfect matching

$\Leftrightarrow$ $\det(M_G)$ is not the zero poly.

An algebraic algorithm for bipartite graph perfect matching

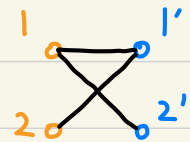
* $[n] := \{1, 2, \cdots, n\}$

  $X := \{x_{11}, x_{12}, \cdots, x_{nn}\}$

* $G = ([n] \uplus [n'], E)$: bipartite graph $E \subseteq [n] \times [n']$

* $$M_G = \begin{array}{c c} & j \\ i & \left[\quad \circ \quad \right]_{n \times n} \end{array}$$

  $$\circ = \begin{cases} x_{ij} & \text{if } (i, j') \in E \\ 0 & \text{o/w} \end{cases}$$



$$\Rightarrow \begin{array}{c c} & \begin{array}{cc} 1' & 2' \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & 0 \end{bmatrix} \end{array}$$

Obs. G has a perfect matching

$\Leftrightarrow$ $\det(M_G)$ is not the zero poly.

Pf. (1) A p.m. in G is $\{(1, j_1), \cdots, (n, j_n)\}$ s.t. $\pi : [n] \to [n]$, $\pi(i) = j_i$ is a permutation.

(2) Each monomial in $\det(\underbrace{X}_{=[x_{ij}]})$ is of the form $\mathrm{sgn}(\pi) \cdot x_{1\pi(1)} \cdots x_{n\pi(n)}$ $\blacksquare$

An algebraic algorithm for bipartite graph perfect matching

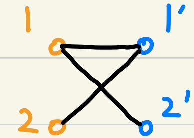
* $[n] := \{1, 2, \cdots, n\}$

  $X := \{x_{11}, x_{12}, \cdots, x_{nn}\}$

* $G = ([n] \uplus [n'], E)$: bipartite graph $E \subseteq [n] \times [n']$

* $$M_G = \begin{array}{cc} & j \\ i & \begin{bmatrix} & & \\ & \circ & \\ & & \end{bmatrix}_{n \times n} \end{array}$$

  $$\circ = \begin{cases} x_{ij} & \text{if } (i, j') \in E \\ 0 & \text{o/w} \end{cases}$$



$$\Rightarrow \begin{array}{cc} & \begin{array}{cc} 1' & 2' \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & 0 \end{bmatrix} \end{array}$$

Obs. G has a perfect matching

$\Longleftrightarrow$ $\det(M_G)$ is not the zero poly.

Q. How to test if $\det(M_G) \equiv 0$?

A. Examining each $x_{1\pi(1)} \cdots x_{n\pi(n)}$ takes $n!$ time.

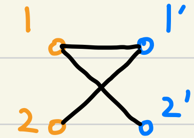
An algebraic algorithm for bipartite graph perfect matching

* $[n] := \{1, 2, \cdots, n\}$

  $X := \{x_{11}, x_{12}, \cdots, x_{nn}\}$

* $G = ([n] \uplus [n'], E)$: bipartite graph $E \subseteq [n] \times [n']$

* $M_G = \begin{array}{c} \\ i \end{array}\begin{bmatrix} & j & \\ & \circ & \\ & & \end{bmatrix}_{n \times n}$, where the (i, j) entry $= \begin{cases} x_{ij} & \text{if } (i, j') \in E \\ 0 & \text{o/w} \end{cases}$



$\Rightarrow \begin{array}{c c} & \begin{array}{cc} 1' & 2' \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & 0 \end{bmatrix} \end{array}$

Obs. G has a perfect matching $\Leftrightarrow$ $\det(M_G)$ is not the zero poly.

Q. How to test if $\det(M_G) \equiv 0$?

A. (1) Randomly substitute each x_{ij} with $a_{ij} \in [2n]$

(2) Compute the determinant of the resulting matrix of numbers ✓

An algebraic algorithm for bipartite graph perfect matching

* $[n] := \{1, 2, \cdots, n\}$

  $X := \{x_{11}, x_{12}, \cdots, x_{nn}\}$

* $G = ([n] \uplus [n'], E)$: bipartite graph $E \subseteq [n] \times [n']$

* $$M_G = \begin{array}{cc} & j \\ i & \begin{bmatrix} & \bigcirc & \\ & & \end{bmatrix}_{n \times n} \end{array}$$

$$\bigcirc = \begin{cases} x_{ij} & \text{if } (i, j') \in E \\ 0 & \text{o/w} \end{cases}$$

Obs. G has a perfect matching

$\Leftrightarrow$ $\det(M_G)$ is not the zero poly.

Q. How to test if $\det(M_G) \equiv 0$?

A. (1) Randomly substitute each x_{ij} with $a_{ij} \in [2n]$

(2) Compute the determinant of the resulting matrix $:= A_G$.

Lemma. [Schwartz-Zippel] If $\det(M_G) \not\equiv 0$,

$$\Pr_{a_{ij} \in_R [2n]} [\det(A_G) \neq 0] \geq 1/2$$

An algebraic algorithm for general graph perfect matching

* $[n] := \{1, 2, \cdots, n\}$

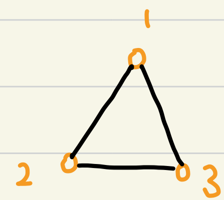
  $X := \{ x_{ij} \mid 1 \le i < j \le n \}$

* $G = ([n], E)$: simple undirected graph. $E \subseteq \binom{[n]}{2}$

* $M_G = $ an $n \times n$ matrix with rows and columns indexed i, j, where

$$(M_G)_{ij} = \begin{cases} x_{ij} & \text{if } \{i,j\} \in E \\ 0 & \text{o/w} \end{cases}$$

$$(M_G)_{ji} = \begin{cases} -x_{ij} & \text{if } \{i,j\} \in E \\ 0 & \text{o/w} \end{cases}$$



$$\Rightarrow \begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 0 & x_{11} & x_{12} \\ -x_{11} & 0 & x_{13} \\ -x_{12} & -x_{13} & 0 \end{bmatrix} \end{array}$$

Obs. G has a perfect matching

$\Leftrightarrow$ $\underbrace{\qquad\qquad}_{\text{Pfaffian}(M_G)}$ is not the zero poly.

From randomised to deterministic?

* We've seen a randomised efficient algorithm for graph perfect matching
    1. Build a matrix of linear forms
    2. Substitute variables with random values
    3. Compute the determinant

* Matrices of linear forms from graphs are special
    1. Bipartite: each variable appears in (at most) one position;
    2. Undirected simple: each variable appears in (at most) two positions.

* What about testing if Det(general matrices of linear forms)=0?
    1. The same random algorithm still applies
    2. Can one devise a deterministic efficient algorithm for this?

From randomised to deterministic?

* We've seen a randomised efficient algorithm for graph perfect matching

* Matrices of linear forms from graphs are special

* What about testing if Det(general matrices of linear forms)=0?

Theorem. [Edmonds] There exist deterministic efficient algorithms for graph perfect matching.

Theorem. [Kabanets-Impagliazzo] A deterministic efficient algorithm for testing Det(general matrices of linear forms)=0 implies that "alg-P neq alg-NP".

- The hardness versus randomness principle [Yao, Nisan-Wigderson]

More correspondences between graphs and tensors

* A recipe from graphs to matrices of linear forms (and therefore tensors)

* A correspondence between perfect matchings and non-zero determinants

* Two more correspondences:

1. Graph isomorphism and tensor isomorphism
2. Independent sets and totally-isotropic spaces

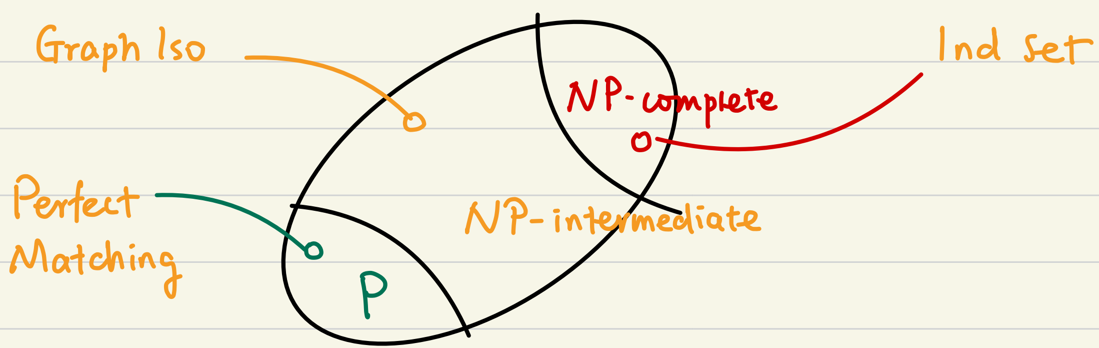
More correspondences between graphs and tensors

* A recipe from graphs to matrices of linear forms (and therefore tensors)

* A correspondence between perfect matchings and non-zero determinants

* Two more correspondences:

1. Graph isomorphism and tensor isomorphism
2. Independent sets and totally-isotropic spaces



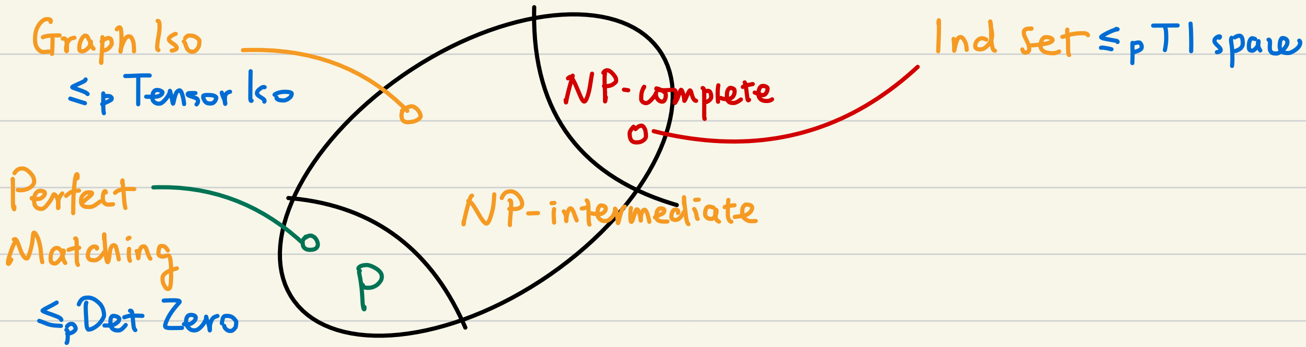
More correspondences between graphs and tensors

* A recipe from graphs to matrices of linear forms (and therefore tensors)

* A correspondence between perfect matchings and non-zero determinants

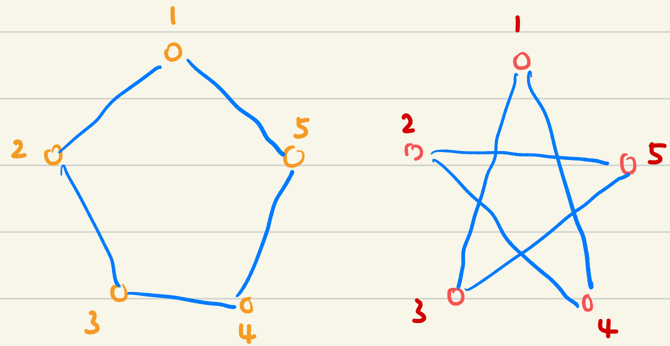
* Two more correspondences:

1. Graph isomorphism and tensor isomorphism
2. Independent sets and totally-isotropic (TI) spaces



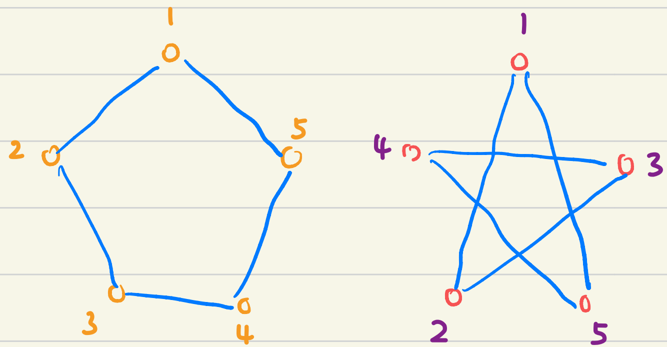
Graph isomorphism and tensor isomorphism

* Graph Isomorphism: Given two graphs, decide if they are the same up to relabelling the vertices



Graph isomorphism and tensor isomorphism

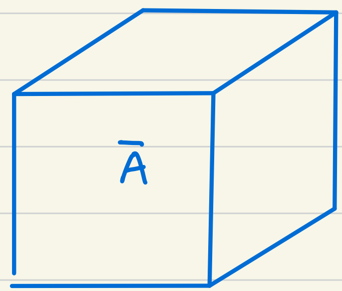
* Graph Isomorphism: Given two graphs, decide if they are the same up to relabelling the vertices

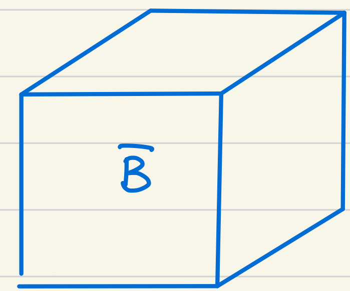


Graph isomorphism and tensor isomorphism

* Graph Isomorphism: Given two graphs, decide if they are the same up to relabelling the vertices

* Tensor Isomorphism: Given two 3-tensors, decide if they are the same up to basis changes of the three directions





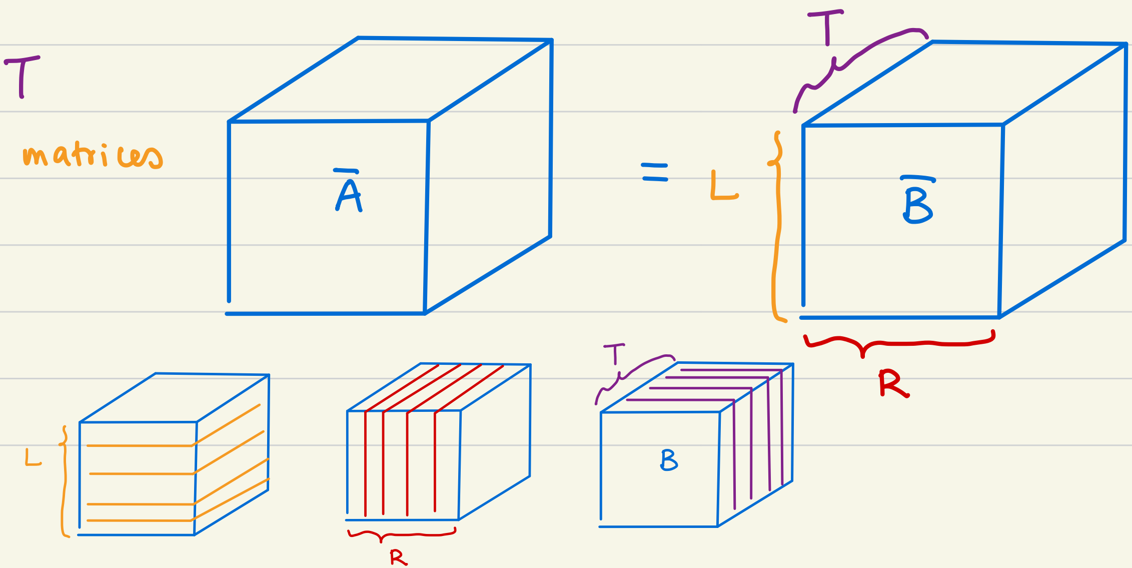
Graph isomorphism and tensor isomorphism

* Graph Isomorphism: Given two graphs, decide if they are the same up to relabelling the vertices

* Tensor Isomorphism: Given two 3-tensors, decide if they are the same up to basis changes of the three directions

$\exists\, L, R, T$

invertible matrices



Graph isomorphism and tensor isomorphism

* **Graph Isomorphism:** Given two graphs, decide if they are the same up to relabelling the vertices

* **Tensor Isomorphism:** Given two 3-tensors, decide if they are the same up to basis changes of the three directions

Definition. Let $\bar{A} = (A_1, \cdots, A_n)$, $\bar{B} = (B_1, \cdots, B_n)$, A_i, B_j : $n \times n$ matrices over $\mathbb{F}$.

Decide if $\exists$ $n \times n$ invertible matrices L, R, $T = (t_{ij})$, s.t.

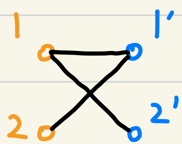
$$\forall i \in [n], \quad A_i = \sum_{j=1}^{n} t_{ij} L B_j R$$

Graph isomorphism and tensor isomorphism

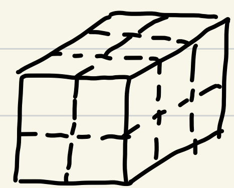
* Graph Isomorphism: Given two graphs, decide if they are the same up to relabelling the vertices

* Tensor Isomorphism: Given two 3-tensors, decide if they are the same up to basis changes of the three directions

Bip graph G $\Rightarrow$ Matrix of linear forms M_G = 3-tensor T_G



$$\Rightarrow \begin{array}{cc} & \begin{array}{cc} 1' & 2' \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & 0 \end{bmatrix} \end{array} = \left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \right)$$



: $2 \times 2 \times 3$ tensor

Graph isomorphism and tensor isomorphism

* **Graph Isomorphism:** Given two graphs, decide if they are the same up to relabelling the vertices

* **Tensor Isomorphism:** Given two 3-tensors, decide if they are the same up to basis changes of the three directions

Bip graph G $\Rightarrow$ Matrix of linear forms M_G = 3-tensor T_G

Bip graph H $\Rightarrow$ Matrix of linear forms M_H = 3-tensor T_H

Graph isomorphism and tensor isomorphism

* Graph Isomorphism: Given two graphs, decide if they are the same up to relabelling the vertices

* Tensor Isomorphism: Given two 3-tensors, decide if they are the same up to basis changes of the three directions

Bip graph G $\Rightarrow$ Matrix of linear forms M_G $=$ 3-tensor T_G

Bip graph H $\Rightarrow$ Matrix of linear forms M_H $=$ 3-tensor T_H

Proposition. $G \cong H$ if and only if $T_G \cong T_H$

– [Li-Q-Wigderson-Wigderson-Zhang]

Graph isomorphism and tensor isomorphism

Proposition. $G \cong H$ if and only if $T_G \cong T_H$

Proof. Suppose $G = ([n] \cup [m'], E)$

$H = ([n] \cup [m'], F)$ $\quad E, F \subseteq [n] \times [m']$. $|E| = |F| = \ell$

Suppose $M_G = A_1 x_1 + \cdots + A_\ell x_\ell$

$M_H = B_1 x_1 + \cdots + B_\ell x_\ell$, $\quad A_i, B_j$: $n \times m$, elementary matrices

Let $S_G = \text{span}\{A_1, \cdots, A_\ell\} \leq M(n \times m, \mathbb{F})$

$S_H = \text{span}\{B_1, \cdots, B_\ell\} \leq M(n \times m, \mathbb{F})$

Graph isomorphism and tensor isomorphism

Proposition. $G \cong H$ if and only if $T_G \cong T_H$

Proof. Suppose $G = ([n] \cup [m'], E)$

$H = ([n] \cup [m'], F)$ $E, F \subseteq [n] \times [m']$. $|E| = |F| = \ell$

Suppose $M_G = A_1 x_1 + \cdots + A_\ell x_\ell$

$M_H = B_1 x_1 + \cdots + B_\ell x_\ell$, A_i, B_j : $n \times m$, elementary matrices

Let $S_G = \mathrm{span}\{A_1, \cdots, A_\ell\} \leq M(n \times m, \mathbb{F})$

$S_H = \mathrm{span}\{B_1, \cdots, B_\ell\} \leq M(n \times m, \mathbb{F})$

$G \cong H$: $\exists\, \pi \in S_n, \tau \in S_m$, s.t $E \xrightarrow{(\pi, \tau)} F$

$T_G \cong T_H$: $\exists\, L \in GL(n, \mathbb{F}), R \in GL(m, \mathbb{F})$. s.t. $\underset{\substack{\| \\ \{LAR^t \mid A \in S_G\}}}{L S_G R^t} = S_H$.

Graph isomorphism and tensor isomorphism

Proposition. $G \cong H$ if and only if $T_G \cong T_H$

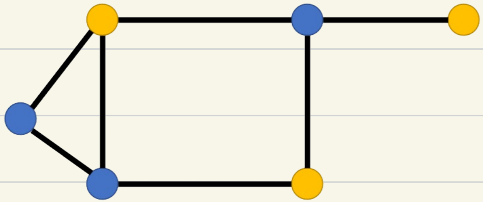
* A reduction from graph iso to tensor iso
  - Tensor iso cannot be too easy from the worst-case algorithm viewpoint

* The analogous result for general graphs (where we get skew-symmetric matrices of linear forms) also holds [He-Q]

* Borrowing techniques from graph iso to tensor iso?
  - Such as individualisation and refinement, Weisfeiler-Leman...?

Independent sets and totally-isotropic spaces

* Independent sets: a subset of vertices with no edges connecting any two of them



* $\bar{A} = A_1 x_1 + \cdots + A_m x_m$

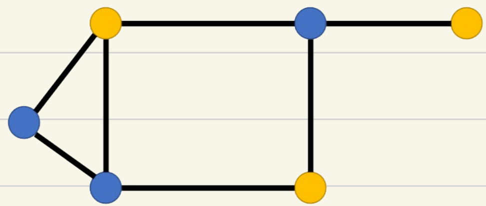
  A_i : $n \times n$ skew-sym matrix

Def $S \le \mathbb{F}^n$ is a totally-isotropic space of $\bar{A}$, if $\forall u, v \in S$, $\forall i \in [m]$,

$$u^t A_i v = 0$$

Independent sets and totally-isotropic spaces

* **Independent sets**: a subset of vertices with no edges connecting any two of them



* $\bar{A} = A_1 x_1 + \cdots + A_m x_m$

  A_i : $n \times n$ skew-sym matrix

Def $S \leq \mathbb{F}^n$ is a **totally-isotropic space** of $\bar{A}$, if $\forall u, v \in S$, $\forall i \in [m]$,

$$u^t A_i v = 0$$

General graph G $\Rightarrow$ Skew-sym matrix of linear forms M_G

Proposition. Max size of independent sets in G = Max dim of TI spaces in M_G

– [Bei - Chen - Guan - Q - Sun]

Summary

* Recipes of constructing tensors from graphs

* Three correspondences of structures

Graphs	Tensors
Perfect matching	Non-zero det
Isomorphism	Isomorphism
Independent sets	Totally-isotropic space

More about these correspondences

* Some correspondences have group-theoretic interpretations

Graphs	Tensors	Groups → class-2, exp-p. p-groups Brahana groups
Perfect matching	Non-zero det	?
Isomorphism	Isomorphism	Isomorphism
Independent sets	Totally-isotropic space	Abelian subgroups

* Not just structures, but also questions and techniques (next lecture :))

Thank you!

And questions please :)