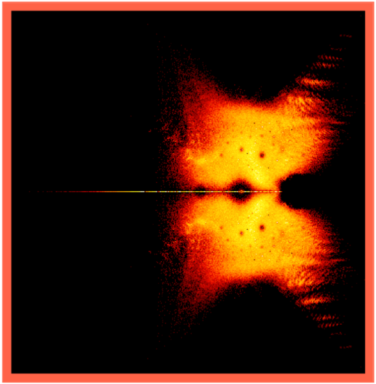
Big data approaches to representation theory

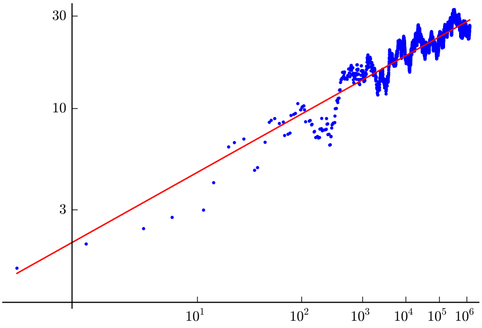
Or: How to waste 1/4 century

~~Accept~~Change what you cannot ~~change~~accept



I report on work of Dłotko–Gurnari–Sazdanovic + Zhang (knot part) and Lacabanne–Vaz (KL part)

The art of conjecturing



- Mathematics is, at least partially, about good conjectures
- Computers are nowadays key for the art of conjecturing
- Early example The Birch–Swinnerton-Dyer conjecture was discovered by computer
- There are 3 stages of conjecturing

The art of conjecturing

```
danieltu@danieltu:/usr/local/magma/IHecke-main$ magma -b type:=A4 I:=[1,2] Examples/BSBasis2.m
```

	id	3	4	32	34	43	432	343	321	324	3214	3243	4321	3432	34321	32432	32143	324321	321432	3214321
id	1																			
3		1																		
4			1																	
32				1																
34					1															
43						1														
432							1													
343		1						1												
321									1											
324										1										
3214											1									
3243												1								
4321													1							
3432				1										1						
34321									1						1					
32432										1						1				
32143																	1			
324321											1							1		
321432																			1	
3214321																	1			1

```
danieltu@danieltu:/usr/local/magma/IHecke-main$ magma -b type:=A4 I:=[2,3] Examples/BSBasis2.m
```

	id	1	4	12	14	43	123	124	143	432	1432	1234	1243	4321	12343	14321	12432	124321	123432	1234321
id	1																			
1		1																		
4			1																	
12				1																
14					1															
43						1														
123							1													
124								1												
143									1											
432										1										
1432											1									
1234												1								
1243													1							
4321														1						
12343							1								1					
14321									1							1				
12432								1									1			
124321					1													1		
123432				1															1	
1234321		1																		1

- **Stage 1** Computer assisted conjectures
- Conjectures are **often** born from calculations, e.g. from by hand calculated prime tables
- Since $\sim$1950 computers have successively replaced by hand calculations and one gets more data for conjectures

Example

For an elliptic curve E of rank r setup the following:

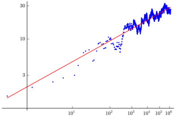
(i) Let N_p=number of points on E modulo p Easy to get

(ii) Set $f\colon \mathbb{N} \to \mathbb{Q}, x \mapsto \prod_{p \leq x} N_p/p$ Easy to get

(iii) Conjecture We have asymptotically $f \sim \text{const.} \log(x)^r$

(iv) The conjecture is true if $r = 0$

- Here is an example for $y^2 = x^3 - 5$ (red=expected value, blue=actual value):

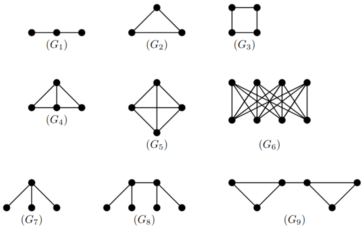


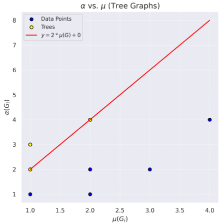
- There is a more general version which is the actual conjecture

The Birch–Swinnerton-Dyer conjecture was discovered on an EDSAC-II


This is an early example of conjectures via data visualization

The art of conjecturing





name	n	μ	α	$n-\mu$	$n-\delta$	Δ^2	connected	tree	regular	bipartite
G_1	3	1	2	2	2	4	True	True	False	True
G_2	3	1	1	2	1	4	True	False	True	False
G_3	4	2	2	2	2	4	True	False	True	True
G_4	4	2	2	2	2	9	True	False	False	False
G_5	4	2	1	2	1	9	True	False	True	False
G_6	8	4	4	4	4	16	True	False	True	True
G_7	4	1	3	3	3	9	True	True	False	True
G_8	6	2	4	4	5	9	True	True	False	True
G_9	6	3	2	3	4	9	True	False	False	False

- Stage 3 Automated conjecturing
- Graffiti (a program that knows certain graphs and graph properties, ∼1985) creates conjectures by data search , trying to match graph+property
- Bait-and-catch No human input at all, but the setting is very restricted and almost all conjectures are rather boring

Example

Conjecture	Graph Family	Authors and Publication
$\alpha(G) \leq \mu(G)$	regular graphs	Caro et al. [64]
$Z(G) \leq \beta(G)$	claw-free graphs	Brimkov et al. [65]
$\alpha(G) \leq \frac{3}{2}\gamma_t(G)$	cubic graphs	Caro et al. [66]
$\alpha(G) \leq \gamma_2(G)$	claw-free graphs	Caro et al. [66]
$\gamma_e(G) \geq \frac{3}{5}\mu(G)$	cubic graphs	Caro et al. [66]
$Z(G) \leq 2\gamma(G)$	cubic graphs	Davila and Henning [67]
$Z_t(G) \leq \frac{3}{2}\gamma_t(G)$	cubic graphs	Davila and Henning [68]
$Z(G) \leq \gamma(G) + 2$	cubic claw-free graphs	Davila [69]

Table 2 Notable conjectures in graph theory generated by *TxGraffiti* and their corresponding publications.

It is impressive what Graffiti and follow-ups conjectured, and a lot of it was proven, e.g.:

Listing 7 Example Conjecture

```
Conjecture 9. If G is connected and regular, then matching_number(G) >=
    independence_number(G). This bound is sharp on 3 graphs.
```

Theorem 1 (Caro et al. [64]). *If G is an r-regular graph with $r > 0$, then*

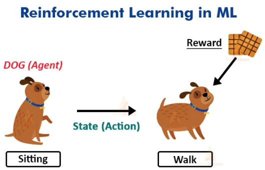
$$\alpha(G) \leq \mu(G),$$

and this bound is sharp.

This is an early example of conjectures via data separation

almost all conjectures are rather boring

The art of conjecturing



Grandmaster level in StarCraft II using multi-agent reinforcement learning

https://doi.org/10.1038/s41586-019-1724-z
Received: 30 August 2019
Accepted: 10 October 2019
Published online: 30 October 2019

Oriol Vinyals[1,3*], Igor Babuschkin[1,3], Wojciech M. Czarnecki[1,3], Michaël Mathieu[1,3], Andrew Dudzik[1,3], Junyoung Chung[1,3], David H. Choi[1,3], Richard Powell[1,3], Timo Ewalds[1,3], Petko Georgiev[1,3], Junhyuk Oh[1,3], Dan Horgan[1,3], Manuel Kroiss[1,3], Ivo Danihelka[1,3], Aja Huang[1,3], Laurent Sifre[1,3], Trevor Cai[1,3], John P. Agapiou[1,3], Max Jaderberg[1], Alexander S. Vezhnevets[1], Rémi Leblond[1], Tobias Pohlen[1], Valentin Dalibard[1], David Budden[1], Yury Sulsky[1], James Molloy[1], Tom L. Paine[1], Caglar Gulcehre[1], Ziyu Wang[1], Tobias Pfaff[1], Yuhuai Wu[1], Roman Ring[1], Dani Yogatama[1], Dario Wünsch[2], Katrina McKinney[1], Oliver Smith[1], Tom Schaul[1], Timothy Lillicrap[1], Koray Kavukcuoglu[1], Demis Hassabis[1], Chris Apps[1,3] & David Silver[1,3*]

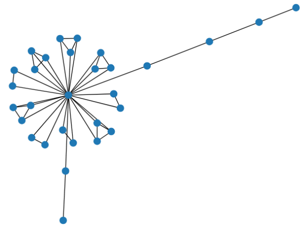
- Stage 2 AI assisted conjecturing
- Machine learning has tools that can effectively detect patterns in data
- Example Reinforcement learning can be used to learn games

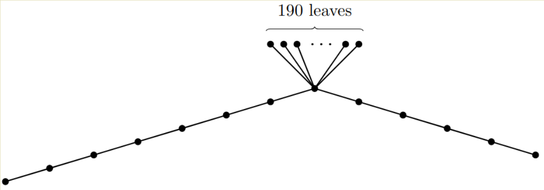
Example

Conjecture 2.3 (Auchiche–Hansen [6]). *Let G be a connected graph on $n \geq 4$ vertices with diameter D, proximity π and distance spectrum $\partial_1 \geq \ldots \geq \partial_n$. Then*

$$\pi + \partial_{\lfloor \frac{2D}{3} \rfloor} > 0.$$

Wagner $\sim$**2021** used reinforcement learning to disprove the above conjecture
Roughly: give points if $\pi + \delta$ is small $\rightsquigarrow$ get examples $\rightsquigarrow$
disprove conjecture by generalizing observed patterns



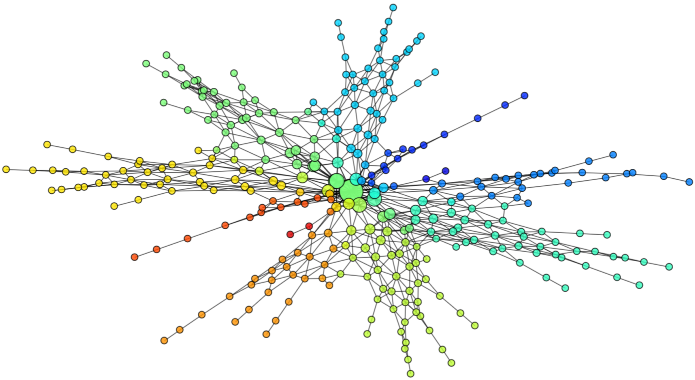


This is an example of conjectures via (counter)example generation

- Example Reinforcement learning can be used to learn games

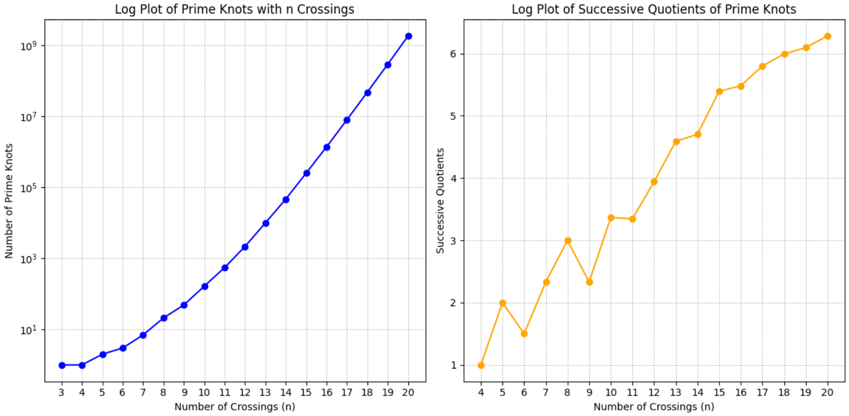
The art of conjecturing

The Jones polynomial:



- Today Explain how stages 1+3 can be applied in quantum topology/ algebra
- Example 1 Knot invariants à la Jones **(Dłotko–Gurnari–Sazdanovic ~2021, Zhang ~2024)**
- Example 2 Kazhdan–Lusztig (KL) polynomials for S_n **(Lacabanne–Vaz ~2024)**

Crucial

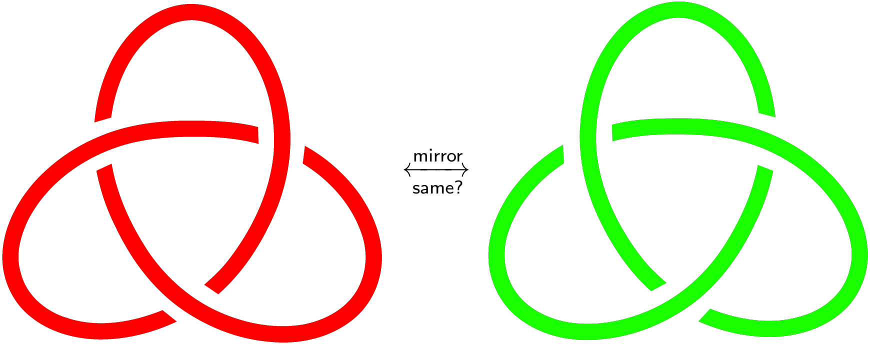


For this to work we need a lot of data; and we are lucky:

Ernst–Sumners $\sim$1987 The number of knots grows at least exponential

KL polynomials are indexed by pairs of partitions $\rightsquigarrow (n!)^2$ or $n!$ (fix one to be trivial)

Big data and knots

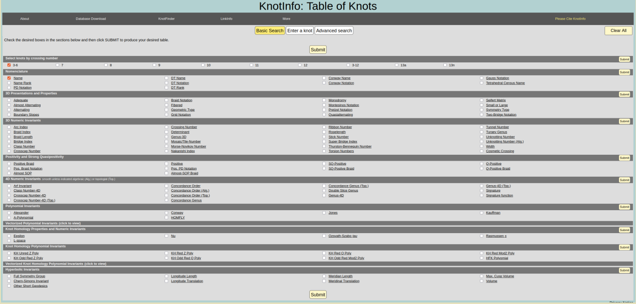


- Problem Deciding whether two knot projections are the same knot is difficult
- Task Find an invariant. Sounds easy? Well, most knot invariants are pretty bad...so: find a 'good' knot invariant
- Task Find a way to decide how good a knot invariant is

Example (of invariants)

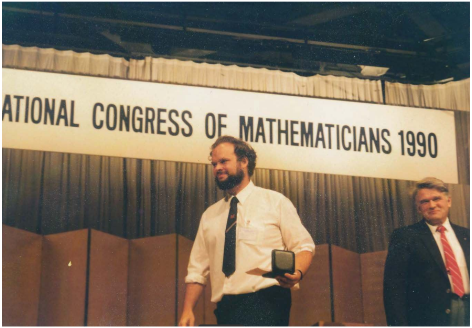
Knot invariants: have the same value on isotopic knots
but might fail to distinguish them

There are more than 50 knot invariants of various types:
components, linkings, colorings, group of colorings, knot group
Alexander, Jones, Kauffman, Khovanov



You don't need to know any of them: they all come in large data bases

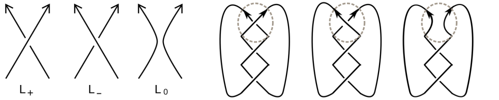
Big data and knots



- Kyoto 1990 Jones receives the fields medal (with Faddeev in the background)
- Quote "Jones discovered an astonishing relationship between von Neumann algebras and geometric topology. As a result, they found a new polynomial invariant for knots and links in 3-space."
- Today The focus is on the quantum knot invariants à la Jones

Big data and knots

Example (of quantum invariants)



- Alexander $(q^{1/2} - q^{-1/2}) \cdot \Delta_{L_0}(q) = \Delta_{L_+}(q) - \Delta_{L_-}(q)$
- Jones polynomial:
  - Skein relation $(q^{1/2} - q^{-1/2}) \cdot J_{L_0}(q) = q^{-1} \cdot J_{L_+}(q) - q \cdot J_{L_-}(q)$
  - Hecke algebra of the braid group
  - Quantum field theory as the unknot normalized vacuum expectation value of the Wilson loop operator in $SU(2)$ Chern–Simons gauge theory
- HOMFLY-PT: $z \cdot H_{L_0}(q) = a \cdot H_{L_+}(q) - a^{-1} \cdot H_{L_-}(q)$
- Khovanov homology – categorification of the Jones polynomial

Everyone loves them (I have spend **1/4 of a century** studying them) and they triggered a lot of research in

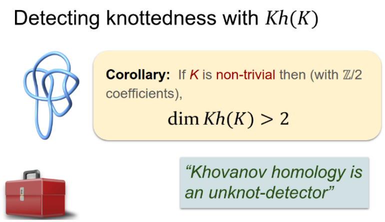
Tensor categories, quantum symmetries and mathematical physics

Question How good are these invariants (say, on prime knots)?

- Ky… …kground)
- Qu… …eumann alge… …nomial inva…
- Today The focus is on the quantum knot invariants à la Jones

Big data and knots

Kronheimer–Mrowka gave a beautiful ICM talk about this (and related) breakthrough(s) Google 'Kronheimer Mrowka ICM 2018'



- First measure Put all (prime) knots in a bag, grab one randomly, how likely distinguishes, say, J the knot (from all others)?
- More formally What is

$$\lim_{n\to\infty} \#(\text{different } J \text{ with } \leq n \text{ crossings})/\#(\text{knots with } \leq n \text{ crossings})?$$

Big data and knots

Kro

brea

Small number coincidences?

KHOVANOV HOMOLOGY DETECTS:

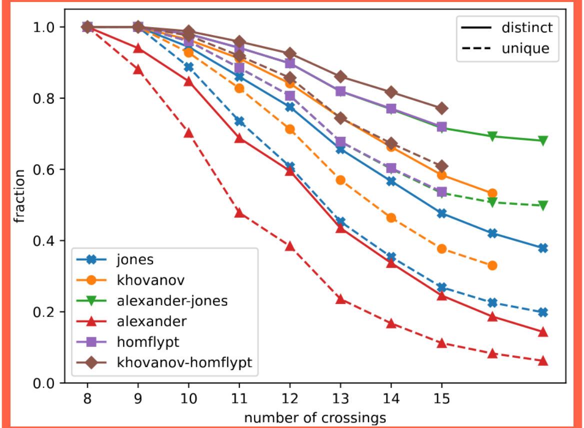
- The unknot: Kronheimer–Mrowka (2010)
- The unlink Hedden–Ni (2013), Batson–Seed (2015)
- The trefoils Baldwin–Sivek (2018)
- The Hopf link Baldwin–Sivek–Xie (2018)
- $2_1 \# 2_1$, the torus link $T(2,4)$ Xie–Zhang (2019)
- Split links Lipshitz–Sarkar (2019)
- The torus link $T(2,6)$ Martin (2020)
- $L6n1$ Xie–Zhang (2020)
- $L7n1$, $2_1 \# 3_1$ Li–Xie–Zhang (2020)
- Cinquefoil $T(5,2)$, non-fibered Baldwin, Siwek (2022)

▶ First meas … omly, how likely distinguishe

▶ More formally What is

$$\lim_{n\to\infty} \#(\text{different } J \text{ with} \leq n \text{ crossings})/\#(\text{knots with} \leq n \text{ crossings})?$$

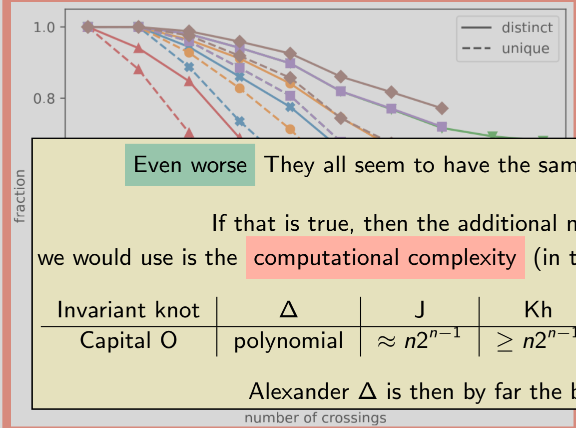
Big data and knots



Unique	% knots
Δ	0.11
J	0.27
Kh	0.38
Δ+J	0.53
H	0.54
H+ Kh	0.61
Distinct	% knots
Δ	0.25
J	0.48
Kh	0.58
Δ+J	0.716
H	0.719
H+ Kh	0.77

- 1/4 century wasted!? They all distinguish knots with probability zero
- Data visualization gives us this conjecture

Big data and knots



Unique	% knots
Δ	0.11
J	0.27
Kh	0.38

Even worse They all seem to have the same dropping rate

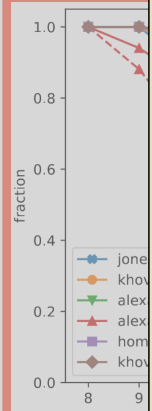
If that is true, then the additional measure
we would use is the computational complexity (in the number of crossings)

Invariant knot	Δ	J	Kh	H
Capital O	polynomial	$\approx n2^{n-1}$	$\geq n2^{n-1}$	superexponential!?

Alexander Δ is then by far the best

- 1/4 century wasted!? They all distinguish knots with probability zero
- Data visualization gives us this conjecture

Big data and k



% knots
0.11
0.27
0.38
0.53
0.54
0.61
% knots
0.25
0.48
0.58
0.716
0.719
0.77

Some good news If we ask for measure 2:
Put all (prime) knots in a bag, grab two randomly how likely distinguishes, say, J these two?

Data visualization gives us the conjecture that the probability is 1 (for all of them)

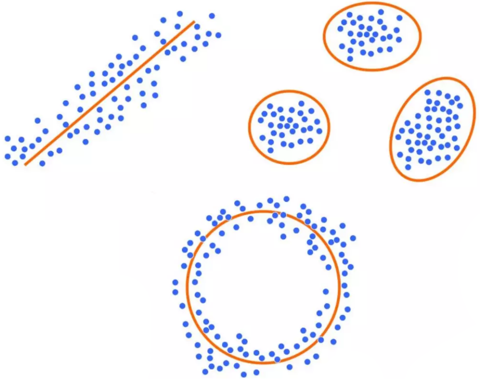
KHOVANOV HOMOLOGY DETECTS:

- The unknot: Kronheimer–Mrowka (2010)
- The unlink Hedden–Ni (2013), Batson–Seed (2015)
- The trefoils Baldwin–Sivek (2018)
- The Hopf link Baldwin–Sivek–Xie (2018)
- $2_1\#2_1$, the torus link $T(2,4)$ Xie–Zhang (2019)
- Split links Lipshitz–Sarkar (2019)
- The torus link $T(2,6)$ Martin (2020)
- $L6n1$ Xie–Zhang (2020)
- $L7n1$, $2_1\#3_1$ Li–Xie–Zhang (2020)
- Cinquefoil $T(5,2)$, non-fibered Baldwin, Siwek (2022)

The complexity questions is however still lurking

- 1/4 centu ... ility zero
- Data visua

Big data and knots - TDA



- TDA (topological data analysis) is the art of finding the shape of data
- Question What shape are quantum knot invariants?
- Question Can the shape measure how good they are?

Big data and knots - TDA

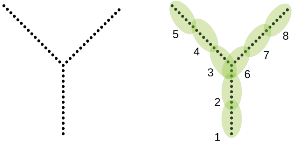
Knots form point clouds!

	q^{-3}	q^{-2}	q^{-1}	q^{0}	q^{1}	q^{2}	q^{3}	q^{4}	q^{5}	q^{6}	q^{7}
$J(0_1)$	0	0	0	1	0	0	0	0	0	0	0
$J(\mathrm{mir}(3_1))$	0	0	0	0	1	0	1	-1	0	0	0
$J(4_1)$	0	1	-1	1	-1	1	0	0	0	0	0
$J(\mathrm{mir}(5_1))$	0	0	0	0	0	1	0	1	-1	1	-1
$J(\mathrm{mir}(5_2))$	0	0	0	0	1	-1	2	-1	1	-1	0
$J(\mathrm{mir}(6_1))$	0	1	-1	2	-2	1	-1	1	0	0	0
$J(\mathrm{mir}(6_2))$	0	0	1	-1	2	-2	2	-2	1	0	0
$J(6_3)$	-1	2	-2	3	-2	2	-1	0	0	0	0

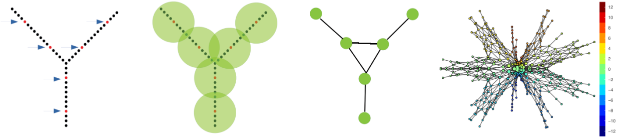
These are vectors in a 11d space

- **Question** Can the shape measure how good they are?

Big data and knots - TDA



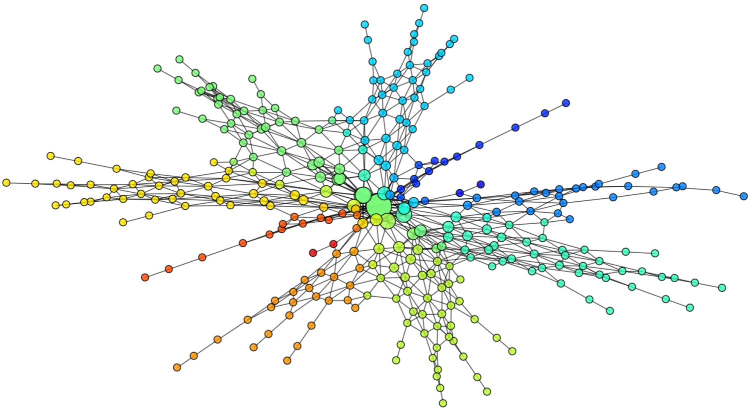
Point cloud | Cover | 1-dim nerve | Coloring



Landmarks | Cover | Ball Mapper | Example

- (Ball) Mapper = a way to turn point clouds into a graph
- Coloring gives additional information
- We see this in examples momentarily

Big data and knots - TDA

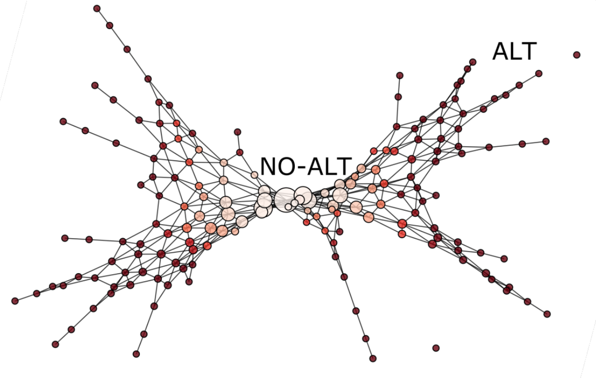


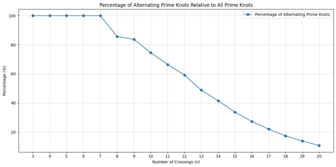
- Now live Ball mapper on knot data
- Play here https://dioscuri-tda.org/BallMapperKnots.html
  https://dustbringer.github.io/web–knot-invariant-comparison/

Data visualization

gives again many possible conjectures
and comparisons

An explanation why detecting alternating knots (but there a not many) is easy:





Most patterns that exists are probably to difficult to prove

https://dustbringer.github.io/web–knot-invariant-comparison/

Big data and KL polynomials

$$\mathrm{ch}(L_w) = \sum_{y\leq w} (-1)^{\ell(w)-\ell(y)} P_{y,w}(1)\,\mathrm{ch}(M_y)$$
$$\mathrm{ch}(M_w) = \sum_{y\leq w} P_{w_0w,w_0y}(1)\,\mathrm{ch}(L_y)$$

CONSTRUCTION OF ARBITRARY KAZHDAN-LUSZTIG POLYNOMIALS IN SYMMETRIC GROUPS

PATRICK POLO

ABSTRACT. To each polynomial P with integral nonnegative coefficients and constant term equal to 1, of degree d, we associate a certain pair of elements (y, w) in the symmetric group S_n, where $n = 1 + d + P(1)$, such that the Kazhdan-Lusztig polynomial $P_{y,w}$ equals P. This pair satisfies $\ell(w) - \ell(y) = 2d + P(1) - 1$, where $\ell(w)$ denotes the number of inversions of w.

STRUCTURE OF CERTAIN INDUCED REPRESENTATIONS OF COMPLEX SEMISIMPLE LIE ALGEBRAS[1]

BY DAYA-NAND VERMA[2]

Communicated by C. W. Curtis, June 14, 1967

THEOREM 7. *The composition factors of $\mathfrak{V}_\Lambda$ are all nonisomorphic and consist of $\mathfrak{M}_M$ for all those M for which $\mathfrak{V}_\Lambda$ contains a copy of $\mathfrak{V}_M$.*

- KL polynomials $P_{u,w}$ (for $u, w \in S_n = \mathrm{Aut}\{1,...,n\}$) = graded base change between Verma and simple modules of $\mathfrak{sl}_{n-1}(\mathbb{C})$
- The only facts you need to know for today
  - People like them (I have spend **1/4 of a century** studying them)
  - They are of the form $1 + \mathbb{N}[q]$
  - **Verma** $\sim$**67** Every KL polynomial is trivial
  - **Polo** $\sim$**99** Every polynomial in $1 + \mathbb{N}[q]$ is a KL polynomial

Big data and KL polynomials

$$\mathrm{ch}(L_w) = \sum_{y \le w} (-1)^{\ell(w)-\ell(y)} P_{y,w}(1)\, \mathrm{ch}(M_y)$$
$$\mathrm{ch}(M_w) = \sum_{y \le w} P_{w_0 w, w_0 y}(1)\, \mathrm{ch}(L_y)$$

CONSTRUCTION OF ARBITRARY KAZHDAN-LUSZTIG POLYNOMIALS IN SYMMETRIC GROUPS

PATRICK POLO

ABSTRACT. To each polynomial P with integral nonnegative coefficients and constant term equal to 1, of degree d, we associate a certain pair of elements (y, w) in the symmetric group S_n, where $n = 1 + d + P(1)$, such that the Kazhdan-Lusztig polynomial $P_{y,w}$ equals P. This pair satisfies $\ell(w) - \ell(y) = 2d + P(1) - 1$, where $\ell(w)$ denotes the number of inversions of w.

STRUCTURE OF CERTAIN INDUCED REPRESENTATIONS OF COMPLEX SEMISIMPLE LIE ALGEBRAS[1]

BY DAYA-NAND VERMA[2]

Communicated by C. W. Curtis, June 14, 1967

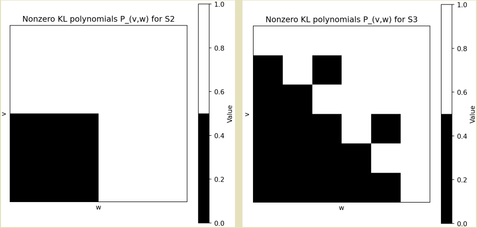
THEOREM 7. *The composition factors of* $\mathfrak{V}_\Lambda$ *are all nonisomorphic and consist of* $\mathfrak{M}_M$ *for all those* M *for which* $\mathfrak{V}_\Lambda$ *contains a copy of* $\mathfrak{V}_M$.

- KL polynomials $P_{u,w}$ (for $u, w \in S_n = \mathrm{Aut}\{1, \dots, n\}$) = graded base change between Verma and simple modules of $\mathfrak{sl}_{n-1}(\mathbb{C})$
- The only facts you need to know for today
  - People like them (I have spend **1/4 of a century** studying them)
  - They are of the form $1 + \mathbb{N}[q]$
  - **Verma ∼67** Every KL polynomial is trivial
  - **Polo ∼99** Every polynomial in $1 + \mathbb{N}[q]$ is a KL polynomial

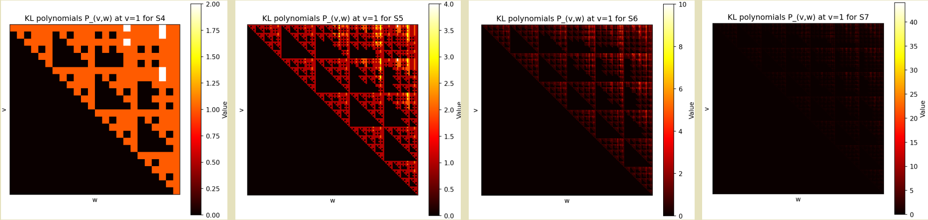
Turns out that both, Verma and Polo, are 'wrong' ☺

Verma is ‘wrong’

If we illustrate the KL polynomials for $q = 1$
as a $n!$-by-$n!$ matrix, we get:



$\mathfrak{sl}_2$ and $\mathfrak{sl}_3$ - so far so good - but:

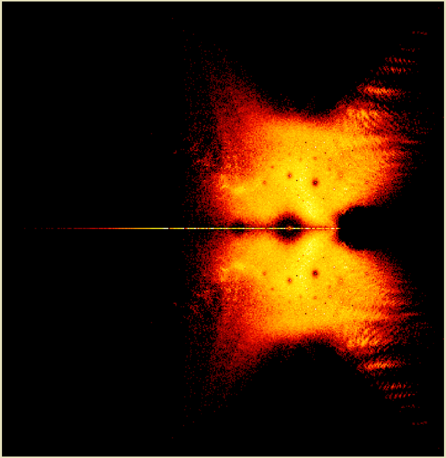


Verma is ‘wrong’ because of a small number coincidence

Theorem (from data visualization) Density bounded between $\Omega(0.708^n)$ and $O(n^{-2})$

Polo is ‘wrong’

If we illustrate the roots of the KL polynomials for S_{10}, we get:



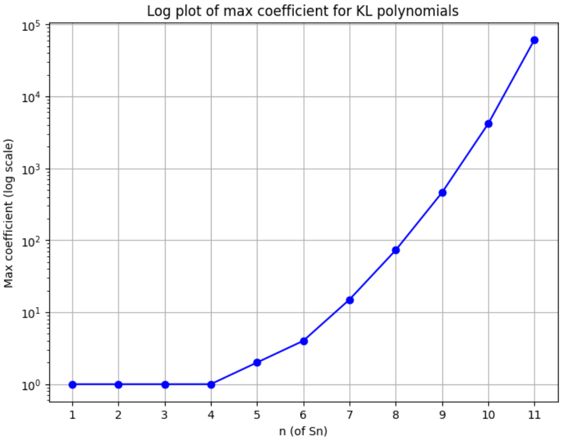
In an equal distribution one would expect a different, more later, pattern

Polo is ‘wrong’ because not all polynomial appear equally likely – far from so!

- **Polo** ∼**99** Every polynomial in $1 + \mathbb{N}[q]$ is a KL polynomial

Big data and KL polynomials

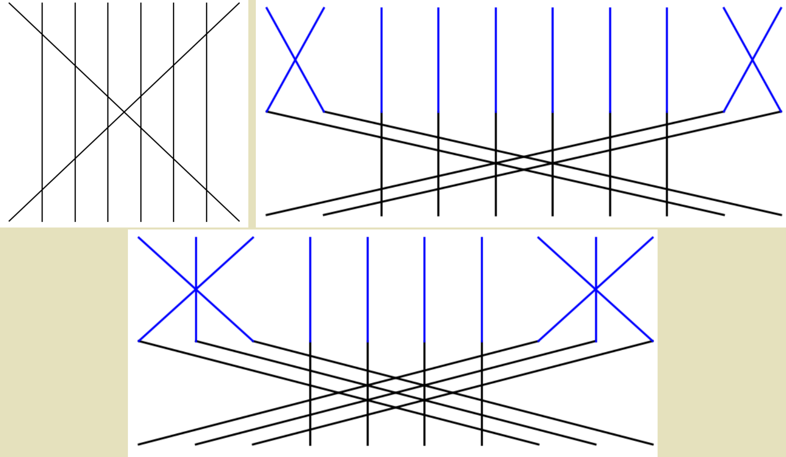
From now on $P_{1,w}$



- **Above** The maximal coefficients of KL polynomials by rank (rank $n \leftrightsquigarrow S_{n+1}$)
- **Theorem** The coefficients of the KL polynomials grow superexponential in n

This theorem was discovered by example generation

It works as follows: take the permutation $w_k^l \in S_{k+l}$ as below

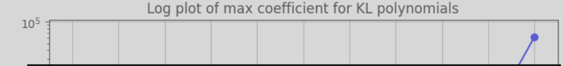


These are w_6^1, w_6^2, w_4^3

▶ … $S_{n+1})$

▶ Theorem The coefficients of the KL polynomials grow superexponential in n

Big data and KL polynomials



Continue

Then $P_{1,w_k^l} = (1+q+\ldots+q^l)^{k-1}$ (*)

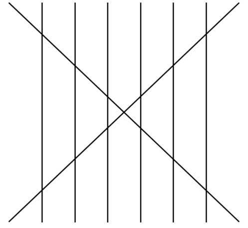
These are the multinomial coefficients $\Rightarrow$ done

To show star, use that $(1+q+\ldots+q^l)^{k-1}$ corresponds to the cohomology ring of $(\mathbb{CP}^l)^{\times k-1})$

$\mathbb{CP}^1$: 

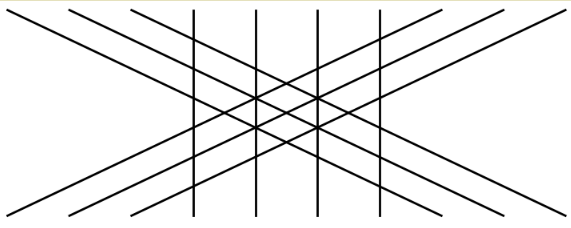
- Above The … (rank $n \leftrightsquigarrow S_{n+1}$)
- Theorem The coefficients of the KL polynomials grow superexponential in n

The case of the binomials ($H^*(S^2)$) was known **Shapiro–Shapiro–Vainshtein ∼95**



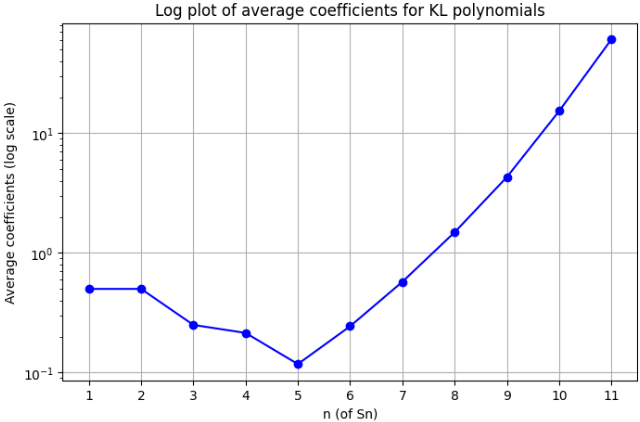
I have however no idea how to guess a generalization without a big data approach

Problem These permutations (the naive generalizations)



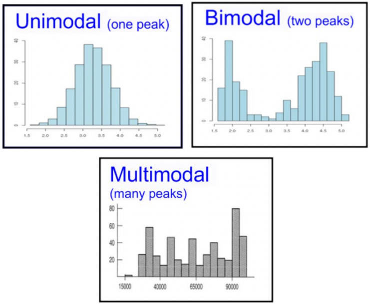
have complicated KL polynomials

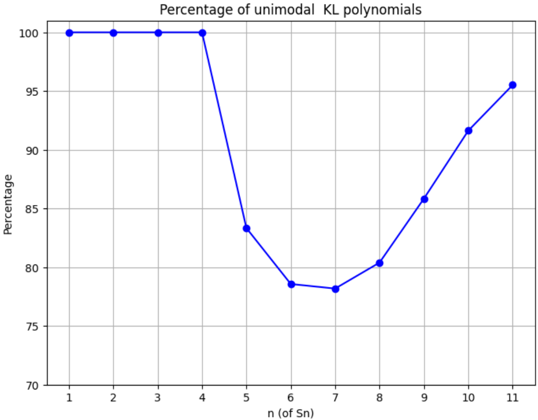
Big data and KL polynomials



- Above The average coefficients of KL polynomials by rank
- Conjecture (data visualization) Verma is 'maximally wrong' in the sense that almost all KL polynomials have large coefficients

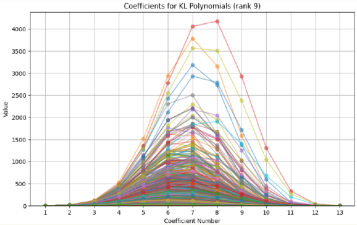
Big data and KL polynomials



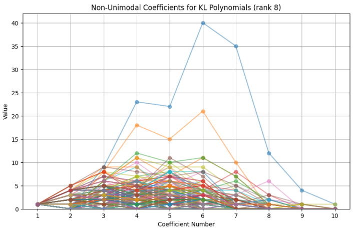


- Above The percentage of KL polynomials that are unimodal
- Conjecture (data visualization) Polo is 'maximally wrong' in the sense that almost all KL polynomials are unimodal (while almost no polynomial is)

Another supporting plot (all KL polynomials for S in one go)



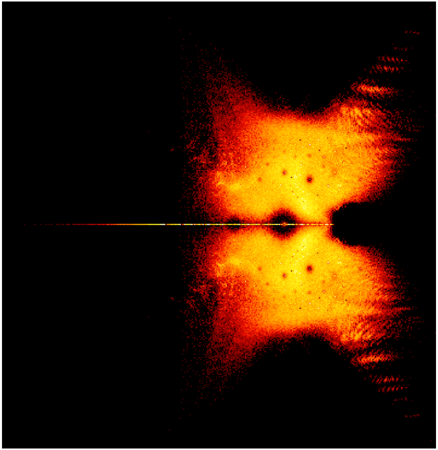
Conjecture (beefed up) Among the non unimodal KL polynomials almost all are bimodal

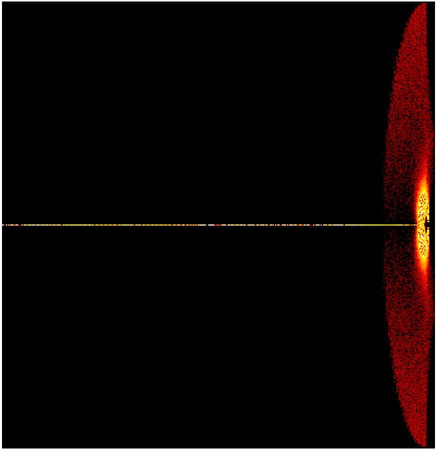


Among the non bimodal KL polynomials almost all are trimodal etc.

Conjecture via data visualization

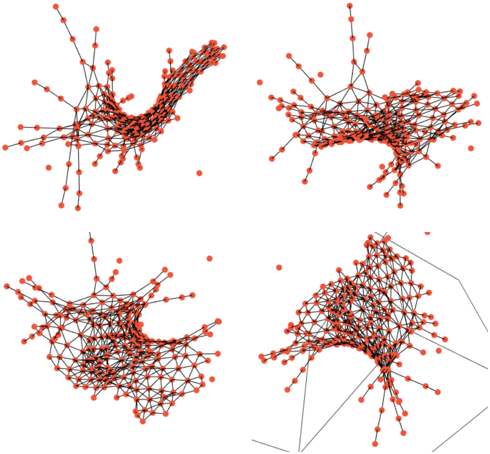
Big data and KL polynomials - TDA





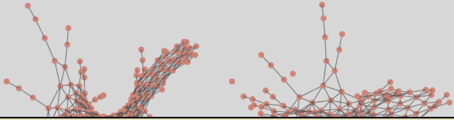
- Above The roots of the KL polynomials for S_{10}
- Below The roots of 100000 randomly generated polynomials in $1 + \mathbb{N}[q]$

Big data and KL polynomials - TDA



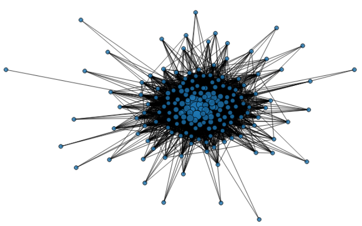
- Now live Ball mapper on KL data
- Play here Now live! (Not yet online in Nov. 2024)

Big data and KL polynomials - TDA



The ball mapper on random polynomials is a

random graph



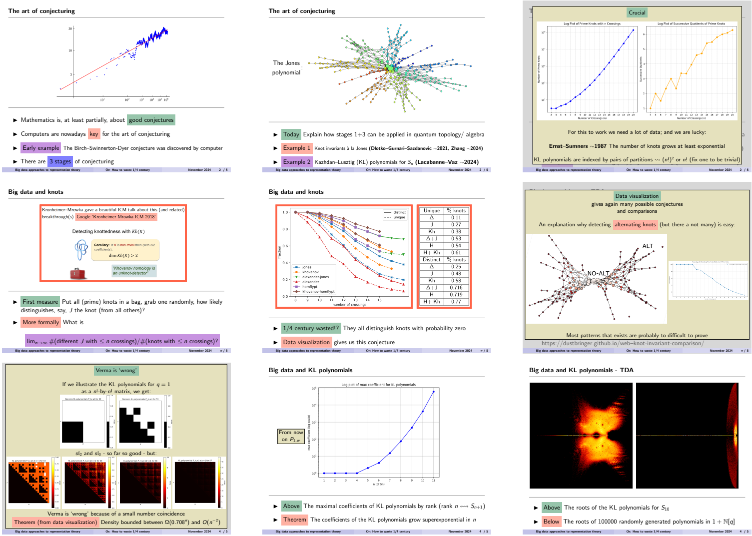
This is very different from the KL ball mapper graph

- Now live Ball mapper on KL data
- Play here Now live! (Not yet online in Nov. 2024)

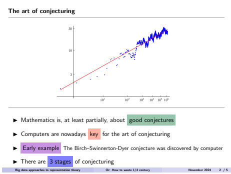
Big data and KL polynomials - TDA

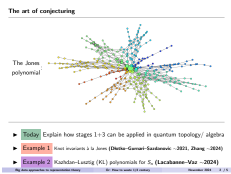
The Alexander polynomial :

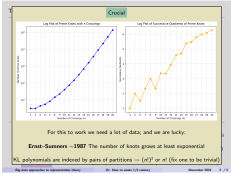
- Above Alexander ball mapper colored by signature mod 4
- There are many more patterns that I have not addressed or computed – let me know if you are interested in any
- To be done Data search approaches

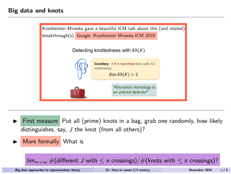


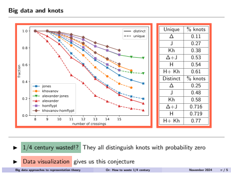
There is still much to do...

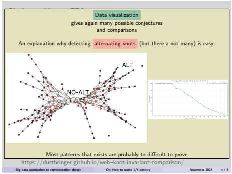


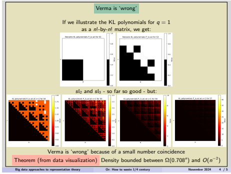


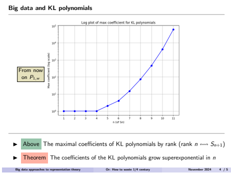


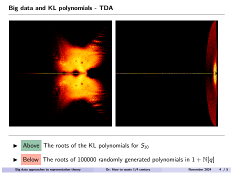












Thanks for your attention!