I see your results – how hard was your data?

Classifying the difficulty of data…

- As illustrated on graphclasses.org…if your problem is in some way related to a graph formulated problem – then a lot is already known about the difficulty of solving problems on that graph class….
- BUT that information is basically “classical complexity theory”: polynomial or exponentially hard (which only captures one aspect – not nuanced gradation of difficulty) – and for ALL algorithms (typically not for a class of algorithms etc.). More recent computational complexity theory looks at finer gradations in computational cost + at classes of algorithms - ”fine grained complexity”
- Some of my recent work and interest seems to fall into this new area of interest – albeit restricted to my problems/methods of interest…

The era of algorithm leaderboards…and deep learning…

- Typically a new idea is tested on some synthetic data (where in principle you can control the complexity - but as we see later – maybe not so clearly controlled – and generally know the “right answer”). But the main interest is on how it performs on real data and against the current “challenging data sets”. Leaderboards of ranked performances of recently proposed and older approaches “benchmark” the success of a new approach
- In some sense, this is what started the new generation of AI – the collection of challenging data (for both training and testing…Imagenet) and the “astounding success” of a deep learning approach on that dataset
- Careers (and money) are made on having an algorithm at the top of a leaderboard…

The era of algorithm leaderboards…and deep learning…

- So what is wrong with such an approach to selecting good solution methods…
  - Well it *is* better than “weaker testing” (not comparing algorithms fairly and on the same dataset(s) – just reporting the results on your own special data etc.
  - But…
    - It can lead to overtuning to the data set….with or without realizing it….
    - It can lead to declaring winners on tiny margins that really aren’t that meaningful (although the “community” will usually begin to declare that dataset no longer challenging enough and go for a new larger and assumed more complex data set)
    - There are usually more then one sensible way to measure good performance and methods don’t always rank the same under difference performance measures
    - Usually a method gets published based on a mixture of how good it is on the leaderboards and on its perceived novelty….emphasis on perceived…

The era of algorithm leaderboards…and deep learning…

Albeit in the area of Evolutionary/Bio-inspired approaches to optimization…this (and related) papers/discussion illustrate some of the issues:

https://link.springer.com/article/10.1007/s11721-019-00165-y

https://link.springer.com/chapter/10.1007/978-3-030-60376-2_10

http://www.dennisweyland.net/blog/?p=12

https://onlinelibrary.wiley.com/doi/epdf/10.1111/itor.12001

The era of algorithm leaderboards…and deep learning…

Note there are at least two distinct criticisms in those articles discussions – **one** is on how simply naming/declaring something as being inspired by some idea (say the way ants presumably work well together) doesn't actually mean the resulting algorithm is novel (or even in keeping with the claimed inspiration) and **the second** is that some datasets on whichg leaderboards are constructed may have some inherent bias (eg., "zero centred") that algorithms and the tuning of these algorithms may exploit to achieve good performance on those datasets (whether or not the authors were aware that this was happening).

My point is not to take sides in the above deabtes/controversy (by saying the particular researchers named are guilty of anything – including overlooping the possibility that their results don't mean what they claim) – my point is the ISSUES touched on are there, and in general can occur across computer science/engineering.

So....some small steps…and restricted to my areas of research..

- If I am given some data – how do I know how hard that data is?
- Do I at least know that (and why) the data is easy for say method X or a class of methods?
- Could I understand better why a certain “tuning”/”tweaking” of the approach worked well on that data?

Difficulty of data….what “measures used”

- Percentage of outliers
- “Structure” of outliers – whether clustered…, whether one-sided, whether near the inlier structure (cut both ways….),
- “Amount” of inlier noise…(std dev)

While some are clearly quantitative and well-defined (percentage of outliers, std dev of inlier noise) – others are ill-defined/loose “clustered outliers”

But in any case – what might be said about inherent difficulty/simplicity of data for ANY method (inlier noise doesn’t affect methods that don’t select “inliers” and fit - like RANSAC)

CVPR 2021 paper proved that a measure known to mathematicians (“Influence”) is high for outliers and low for inliers *for the ideal single structure setting”. Used sampling to *estimate* influence. Removed highest influence one at a time (and re-estimate). Present work sheds more light on the need to do this “iteratively”.

Notion is that *when you are “close” to ideal – then – by continuity – surely the above mostly still holds…. Problem – “how far” does “contiuity” go….

OK – so we already have a notion that sort of answers our question….

Search the 2^n Boolean cube classifying each subset of data as to whether feasible or not.

Count the upper zeros (maximal faces of the independence complex of the minimal infeasible subset hypergraph)….more <- -> further from ideal <- -> harder

But is that the best answer we can give…?

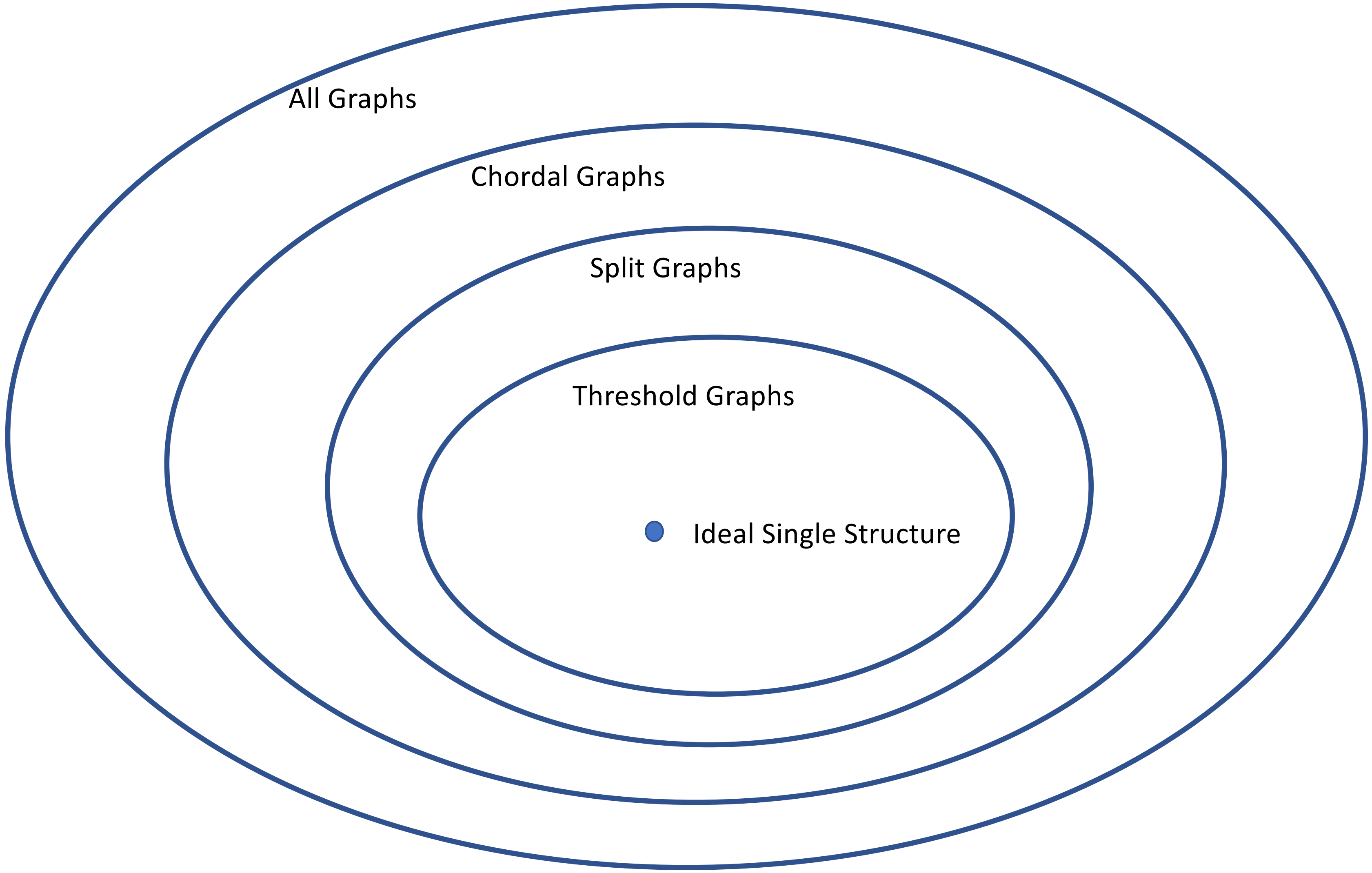
It doesn’t seem to tell us much…..

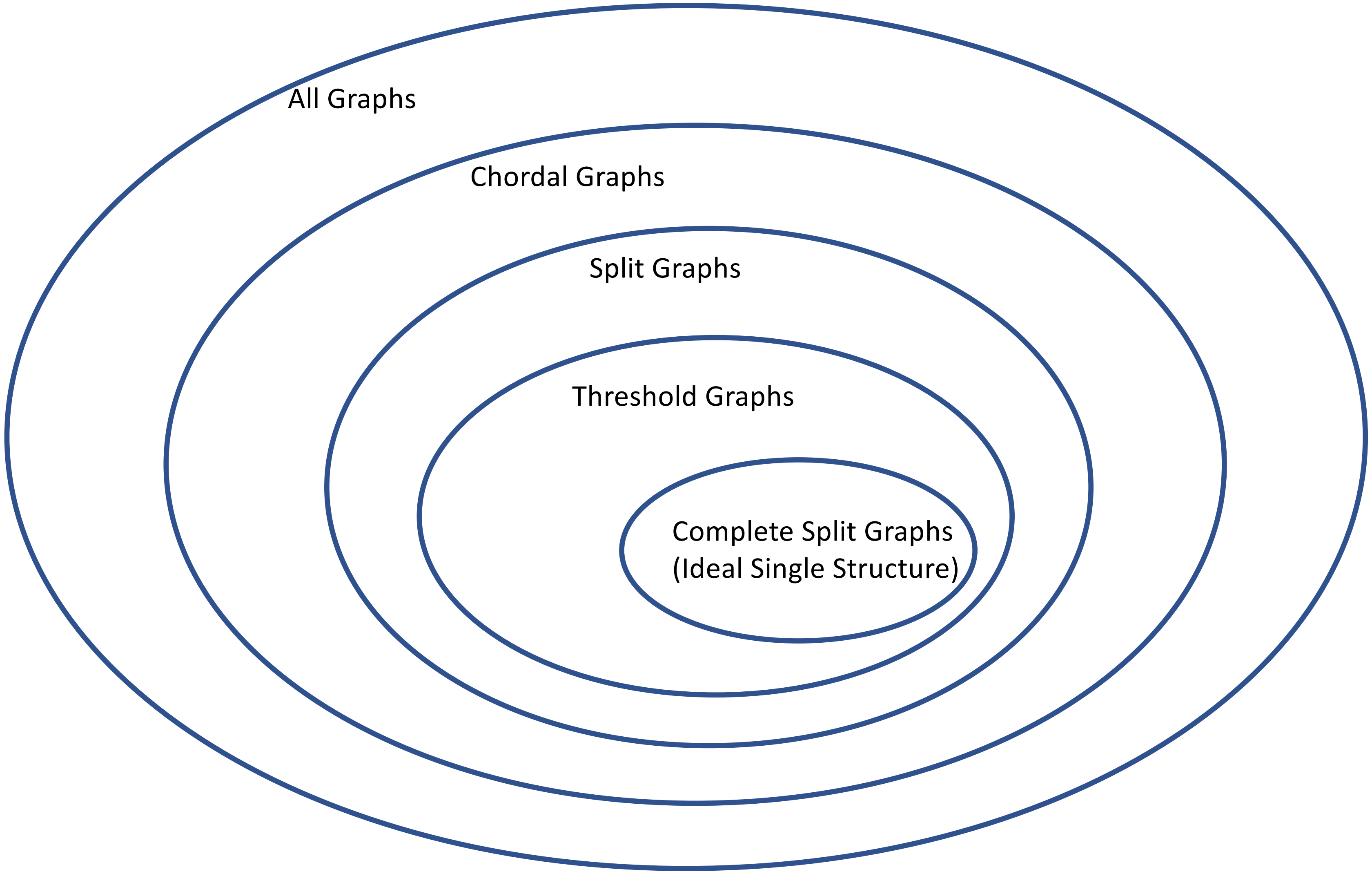
Is there some more intuitive/useful structure to this question (and answer)???

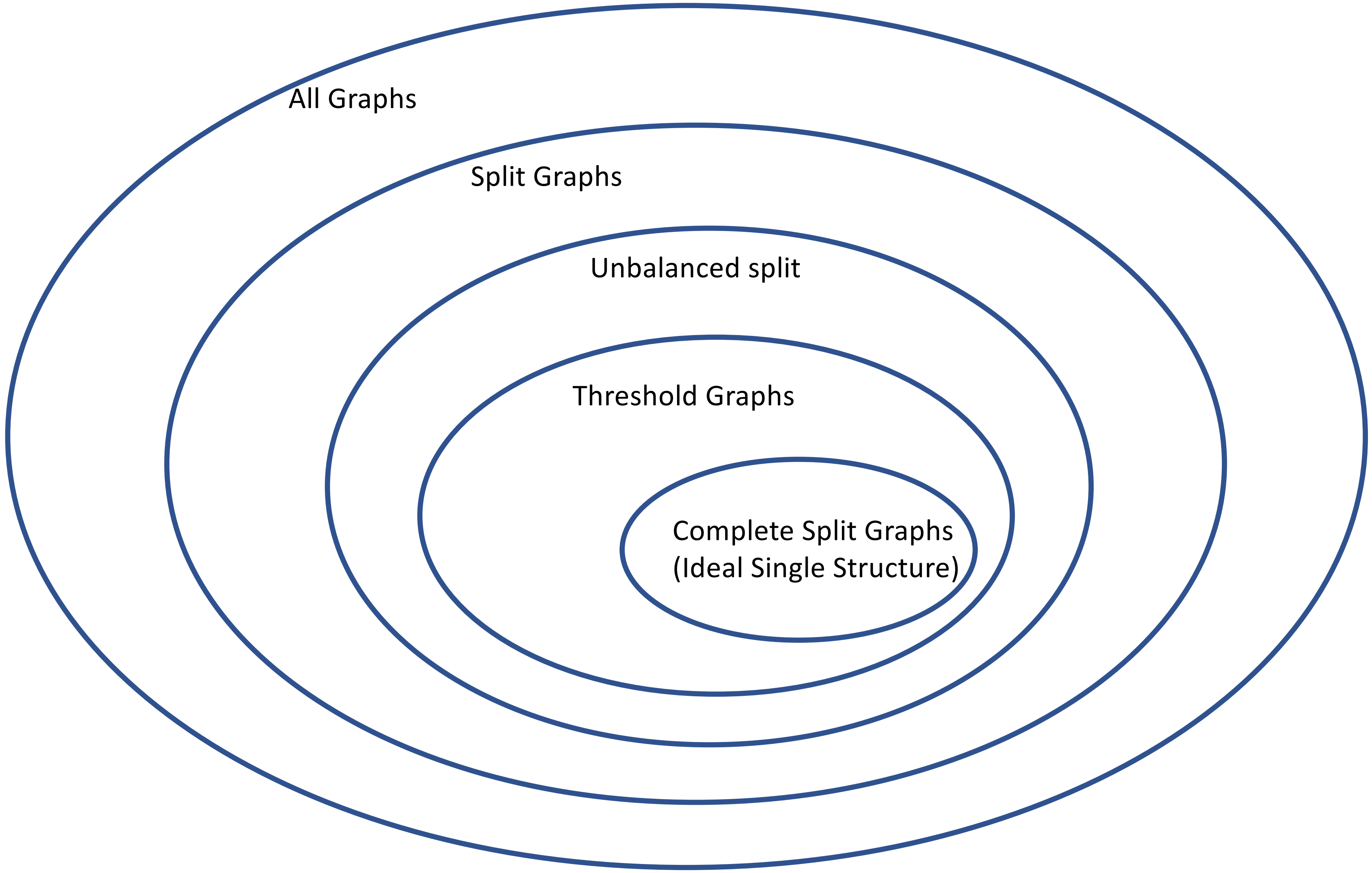
Spoiler…..

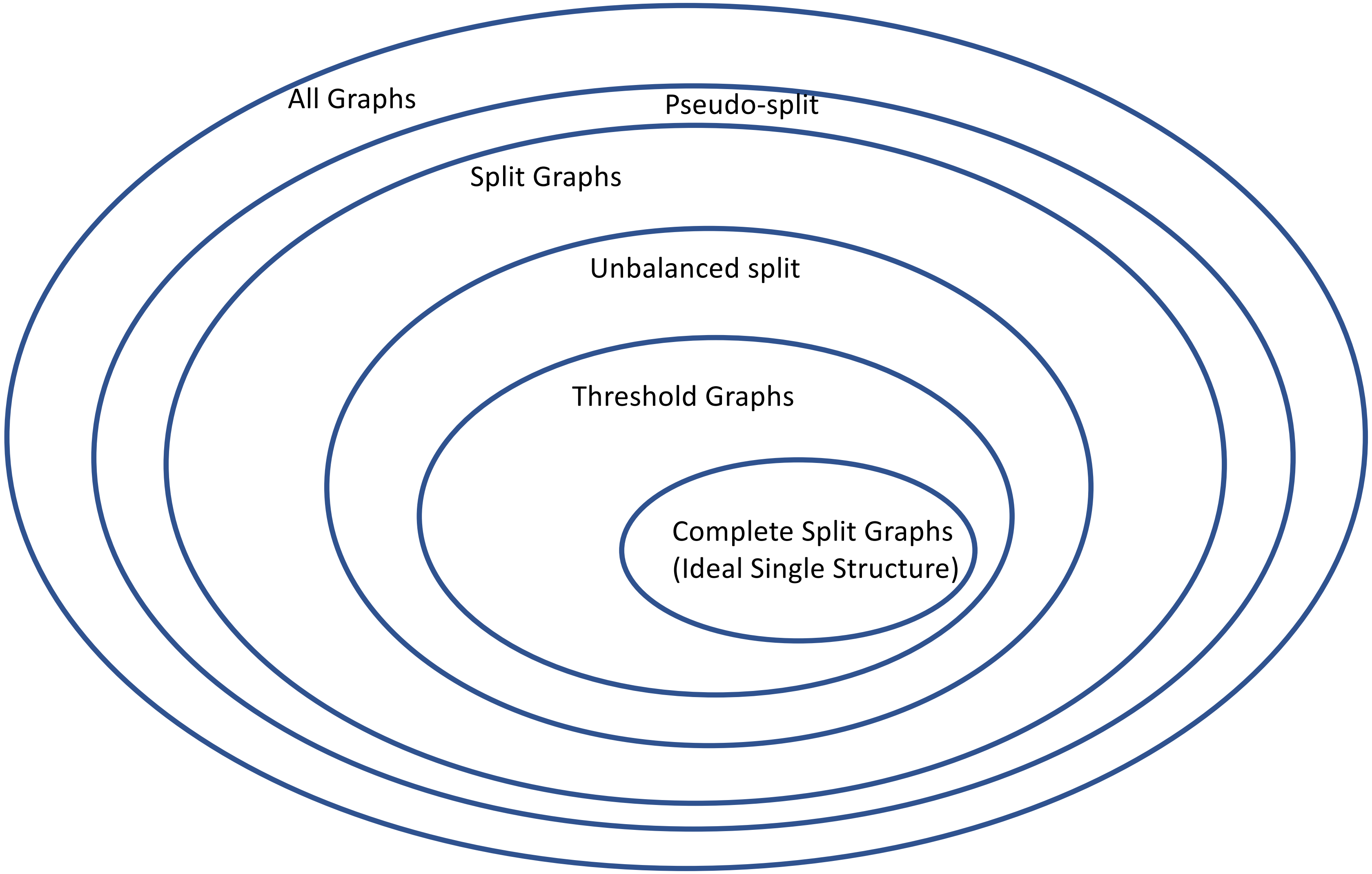
Main message is that (hyper)graph classes may be useful in understanding the complexity of data for MaxCon

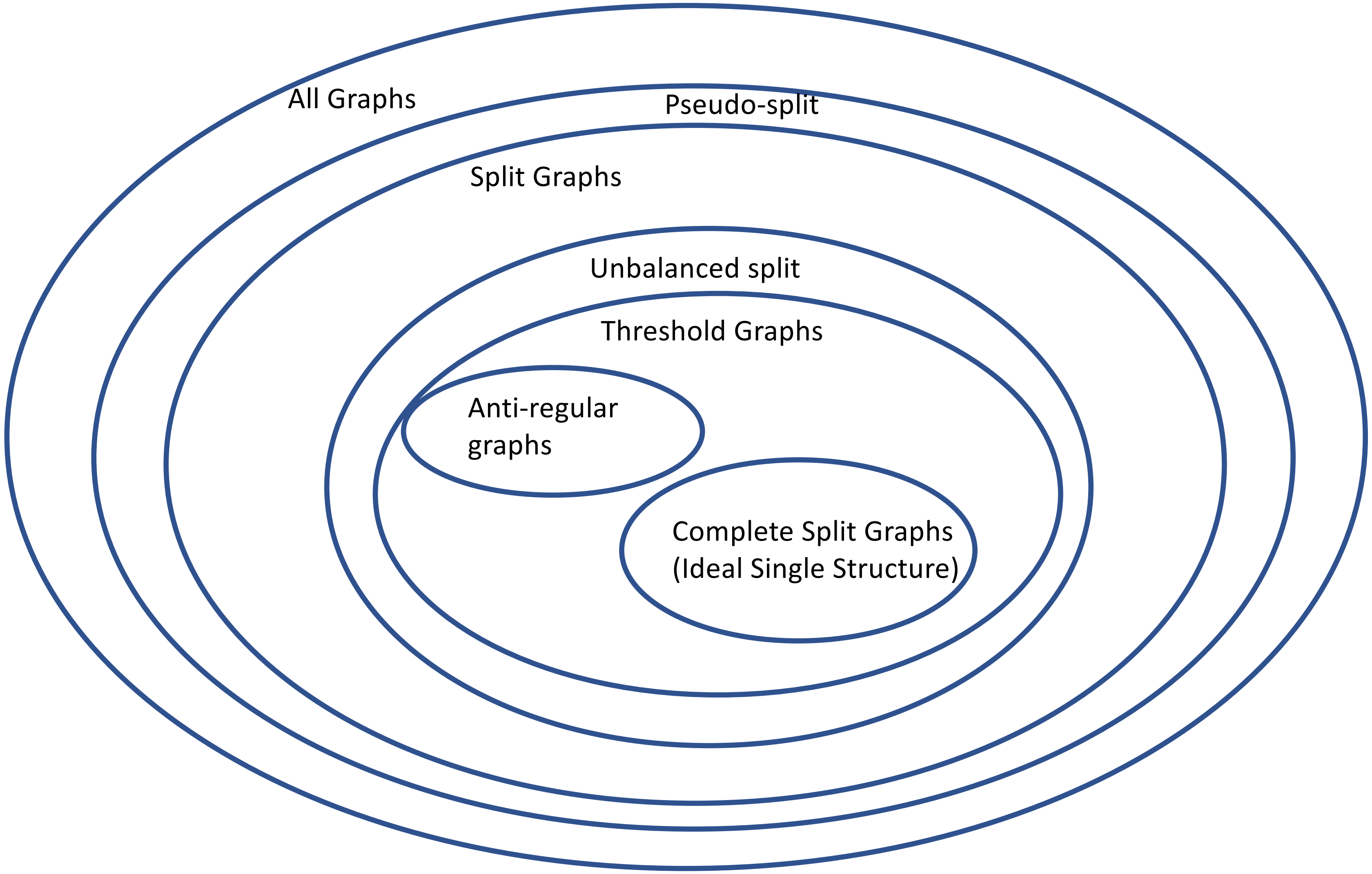
(great website graphclasses.org)

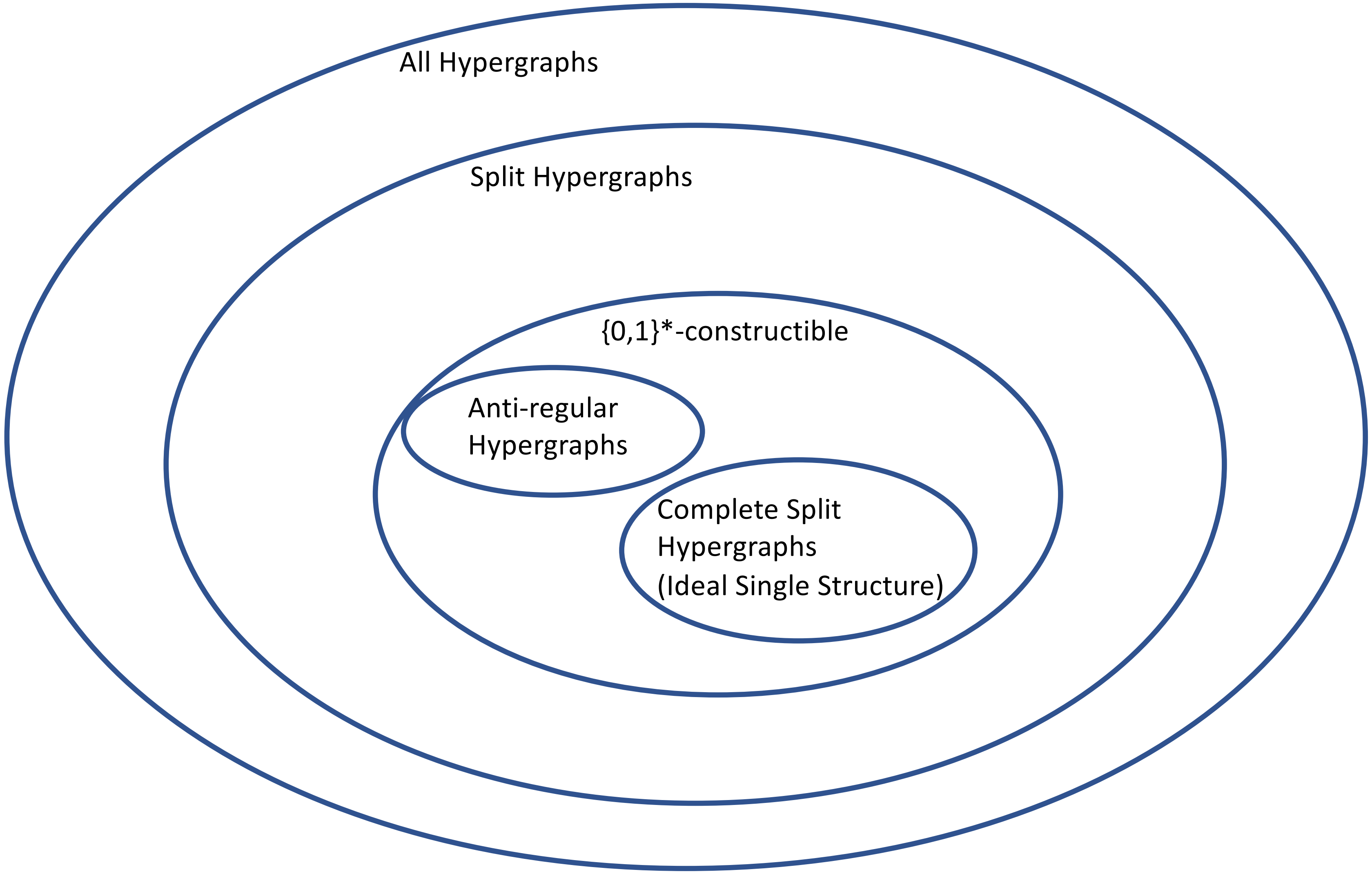

All Graphs
Chordal Graphs
Split Graphs
Threshold Graphs
Ideal Single Structure


All Graphs
Chordal Graphs
Split Graphs
Threshold Graphs
Complete Split Graphs
(Ideal Single Structure)


All Graphs
Split Graphs
Unbalanced split
Threshold Graphs
Complete Split Graphs
(Ideal Single Structure)


All Graphs
Pseudo-split
Split Graphs
Unbalanced split
Threshold Graphs
Complete Split Graphs
(Ideal Single Structure)


All Graphs
Pseudo-split
Split Graphs
Unbalanced split
Threshold Graphs
Anti-regular graphs
Complete Split Graphs (Ideal Single Structure)


All Hypergraphs
Split Hypergraphs
{0,1}*-constructible
Anti-regular Hypergraphs
Complete Split Hypergraphs (Ideal Single Structure)

{0,1}* constructible (hyper)graphs

- Construction string…. e.g., 00010111011
- Describes incremental construction:
  - 0 – add an isolated vertex
  - 1 – add a totally connected (to every existing vertex) vertex (dominating vertex)
- First “few” `bits’ are ambiguous – e.g., for a graph first entry can only be an isolated vertex.

For graphs – these are exactly the “threshold graphs”.

Key Background ideas

- Basis size – smallest number of data points that can be infeasible – for affine line fitting (y=ax+b) basis size is 3.
- If a set of data points is infeasible then it must contain a basis that is infeasible (subset of points that are "extreme" for that set of points)
- Graph theorists would recognize this as corresponding to independent/dependent sets vertices
- Corollary – the set of all feasible points is the independence complex of a hypergraph (all infeasible bases/mimimal sized subsets)
- Relates to Boolean Monotone function of the power set of the data (Boolean Cube). 0 if in the independence complex, 1 if not.

Key Background ideas

- Influence is basically the (weighted/normalized) count of edges (in the hypercube! – not to be confused with edges in the infeasibility hypergraph!) between 0 and 1 of the Boolean Monotone Function (0 feasible, 1 infeasible) – in each “direction” (one direction for each data point/vertex - influence of that data point)

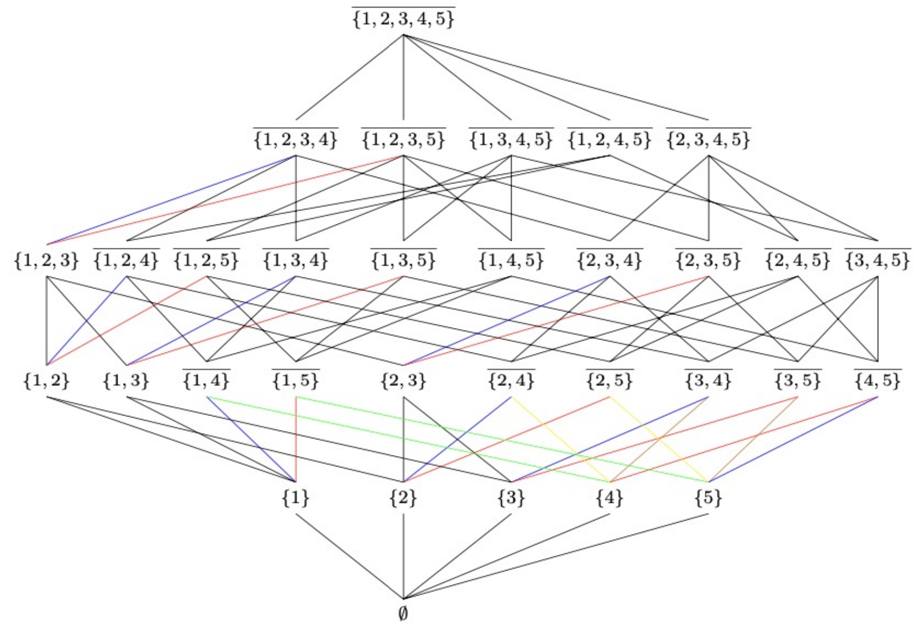

$\overline{\{1,2,3,4,5\}}$
$\overline{\{1,2,3,4\}}$ $\overline{\{1,2,3,5\}}$ $\overline{\{1,3,4,5\}}$ $\overline{\{1,2,4,5\}}$ $\overline{\{2,3,4,5\}}$
$\{1,2,3\}$ $\overline{\{1,2,4\}}$ $\overline{\{1,2,5\}}$ $\overline{\{1,3,4\}}$ $\overline{\{1,3,5\}}$ $\overline{\{1,4,5\}}$ $\overline{\{2,3,4\}}$ $\overline{\{2,3,5\}}$ $\overline{\{2,4,5\}}$ $\overline{\{3,4,5\}}$
$\{1,2\}$ $\{1,3\}$ $\overline{\{1,4\}}$ $\overline{\{1,5\}}$ $\{2,3\}$ $\overline{\{2,4\}}$ $\overline{\{2,5\}}$ $\overline{\{3,4\}}$ $\overline{\{3,5\}}$ $\overline{\{4,5\}}$
$\{1\}$ $\{2\}$ $\{3\}$ $\{4\}$ $\{5\}$
$\emptyset$

Fig. 3. Boolean Cube on 5 vertices (data points) - Independence Complex of a graph In the Boolean Cube we have annotated with an overbar the dependent sets of the 00011 graph (describing the ideal single structure of size 3, and 2 outliers). This defines a Boolean Monotone Function (independent sets assigned value 0 and dependent sets a value 1. For this function, edges that transition from 0 to 1 contribute an increment of 1 to the influence of the vertex (data point), which is added to the feasible subset to make the infeasible subset.
This is depicting example 1. Transitions are colour coded using the following palette v_1 green, v_2 yellow, v_3 brown, v_4 blue, v_5 red. Counting the edges of the appropriate colour gives the influence counts of example 1.

MaxCon	(Hyper)graph Concept
Inlier (to main structure)	vertex that belongs to the Maximum Independent set (of the hypergraph of minimal sized infeasible subsets)
Outlier (to main structure)	vertex that belongs to the Minimum Vertyex Cover (of the hypergraph of minimal sized infeasible subsets)
Feasible set	Member of the independence complex (of the hypergraph of…..)
MaxCon solution	Maximum independent set (of the hypergraph of….)
Ideal Single Structure data	Hypergraph if minimal sized infeasible sets belongs to Complete Split Hypergraph
Essentially single structure data	?? Split Hypergraph? Maybe some superclass
Inlier/Outlier separable (in principle) by thresholding influence	?? At least {0,1}*-constructible hypergraphs ….maybe some superset

Special Oracles – that general (Hyper)graph problems do not have

MaxCon	(Hyper)graph Concept
l_infinity fit	Oracle of independence of corresponding vertices – if max residual is less than or equal to epsilon then independent, delse dependent
Cover of a p+1 subset of the data	Subset of vertices that are independent if cover vertices are independent

Consequences – Maximum independent set for maxCon problem is 0(n^{p+1})…..in worst case….(ignoring fixed cost of p+1-sized l_infinity fit and of cover of p+1 subset).

1-D *subspace* model

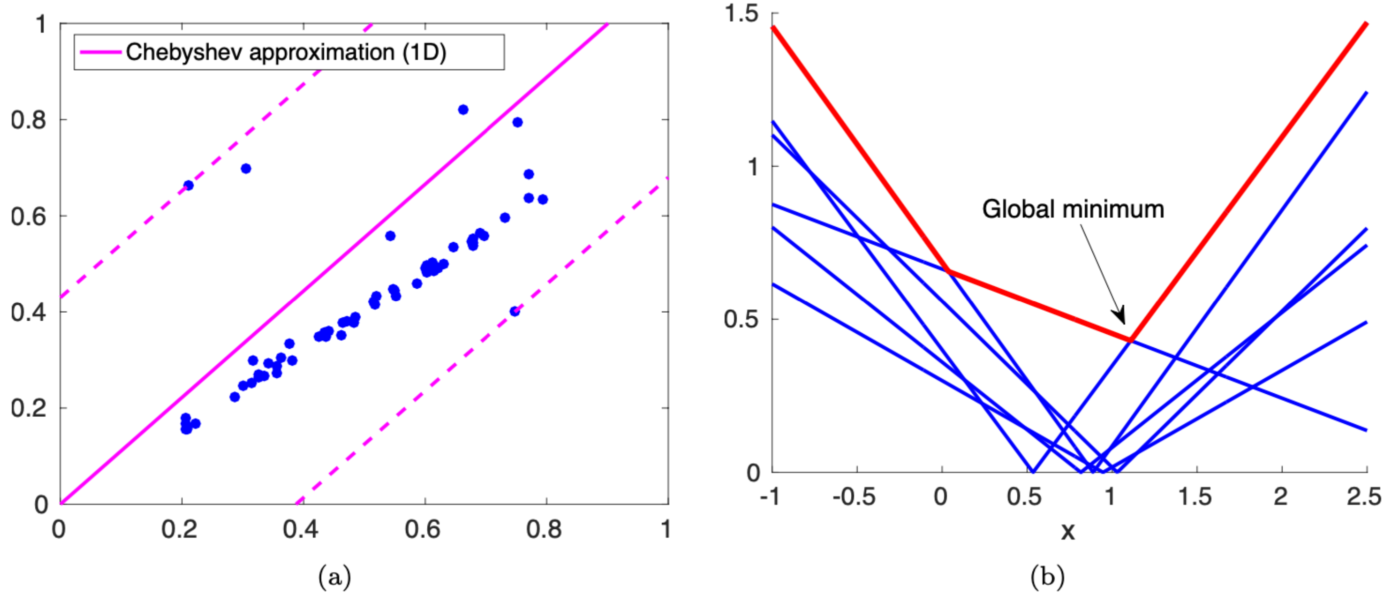
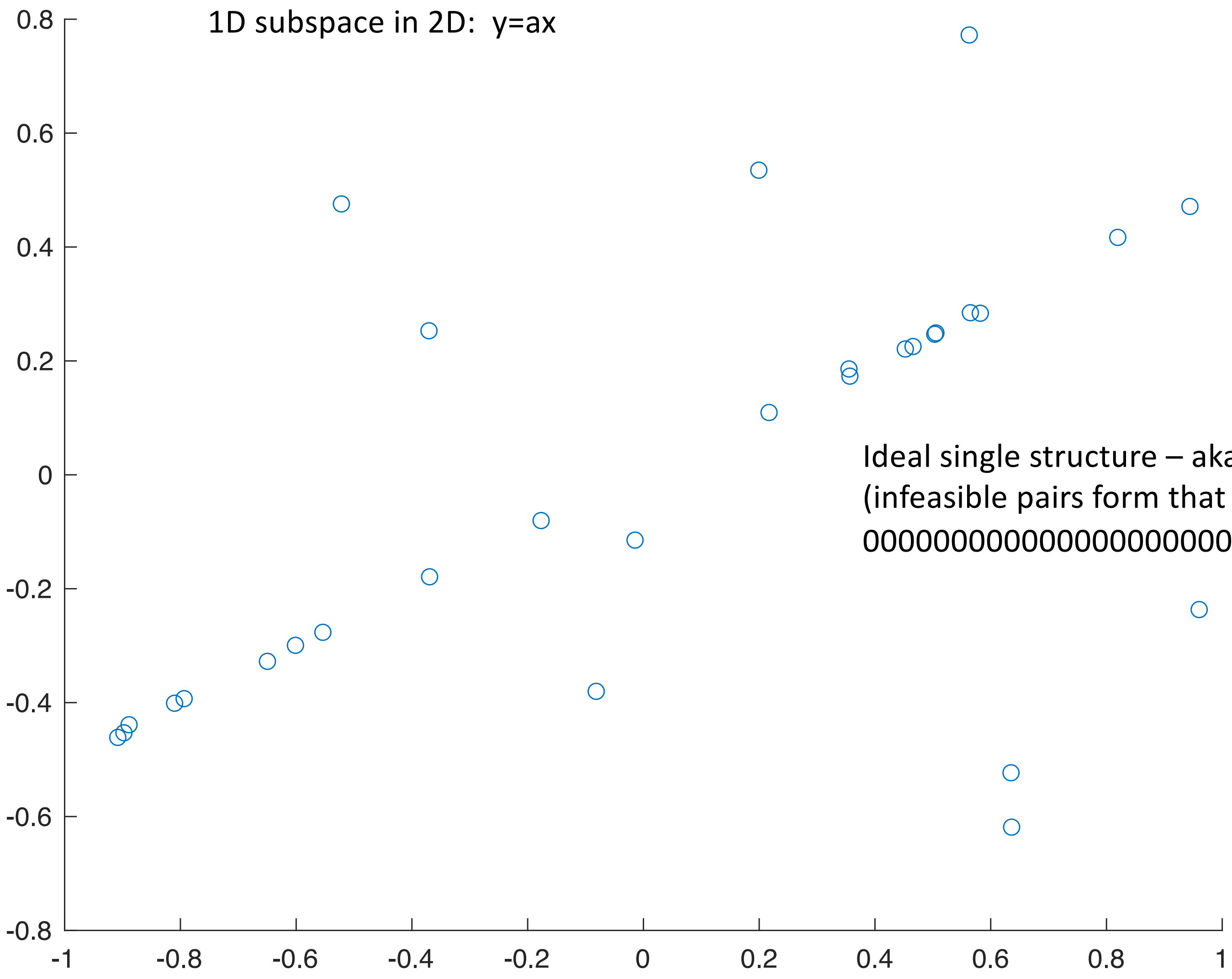


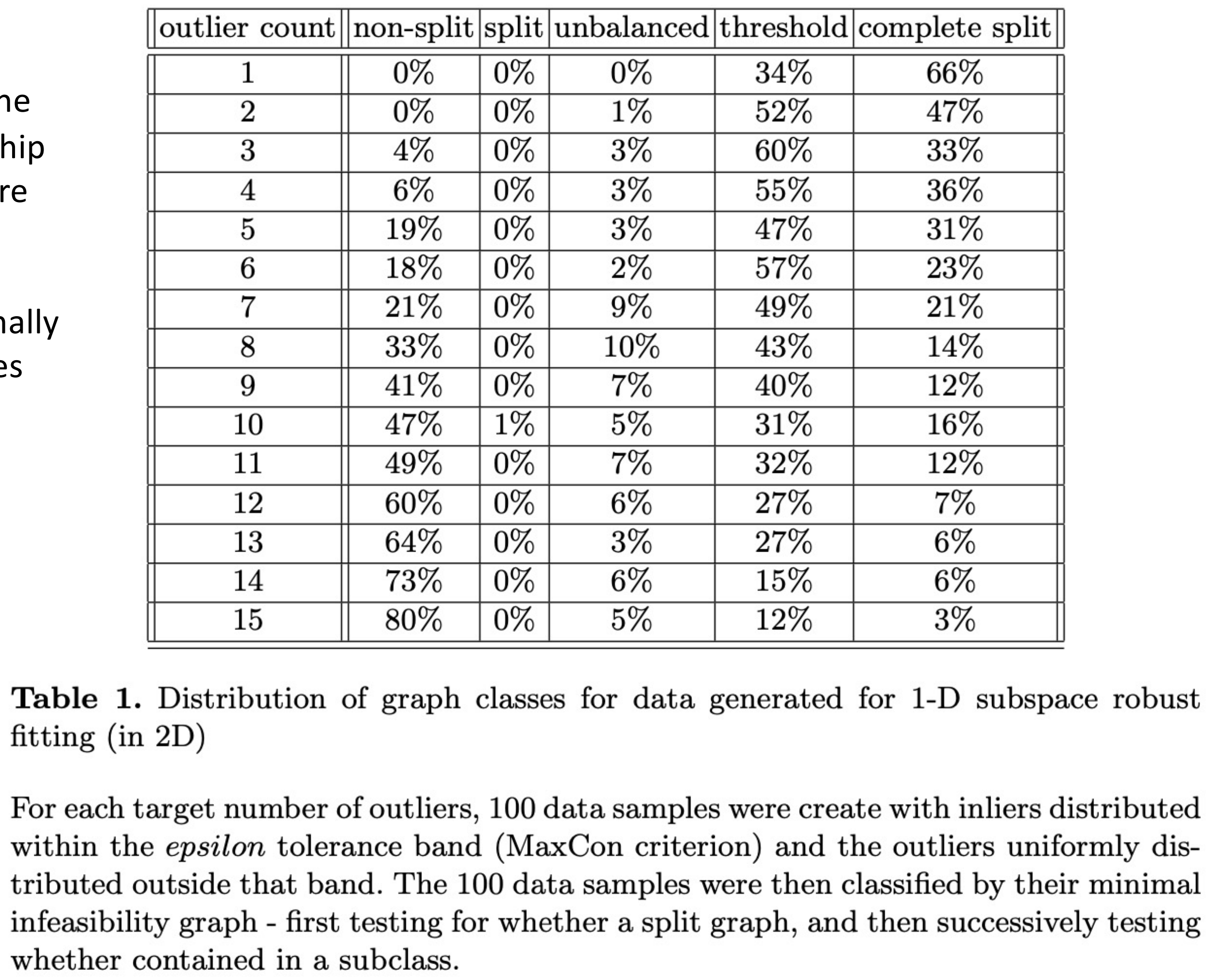
Figure 2.12: (a) Chebyshev approximation for the 1D linear relation $ax = b$ on a set of points $\mathcal{D} = \{(a_i, b_i)\}_{i=1}^{N}$. The equation $ax = b$ defines a line that passes through the origin. The dashed lines indicate the optimised minimax residual value. (b) Illustrating the Chebyshev fit in the parameter space $x \in \mathbb{R}$. Here, each V-shaped curve in blue represents a residual function $|a_i x - b_i|$. The red curve indicates the objective function $\max_i |a_i x - b_i|$.

1D subspace in 2D: y=ax



Ideal single structure – aka complete split graph
(infeasible pairs form that graph)
000000000000000000000111111111

For the *graph* setting many of the test for membership in these classes are cheap – degree tests…or finding isolated or maximally connected vertices and removing…



outlier count	non-split	split	unbalanced	threshold	complete split
1	0%	0%	0%	34%	66%
2	0%	0%	1%	52%	47%
3	4%	0%	3%	60%	33%
4	6%	0%	3%	55%	36%
5	19%	0%	3%	47%	31%
6	18%	0%	2%	57%	23%
7	21%	0%	9%	49%	21%
8	33%	0%	10%	43%	14%
9	41%	0%	7%	40%	12%
10	47%	1%	5%	31%	16%
11	49%	0%	7%	32%	12%
12	60%	0%	6%	27%	7%
13	64%	0%	3%	27%	6%
14	73%	0%	6%	15%	6%
15	80%	0%	5%	12%	3%

Table 1. Distribution of graph classes for data generated for 1-D subspace robust fitting (in 2D)

For each target number of outliers, 100 data samples were create with inliers distributed within the *epsilon* tolerance band (MaxCon criterion) and the outliers uniformly distributed outside that band. The 100 data samples were then classified by their minimal infeasibility graph - first testing for whether a split graph, and then successively testing whether contained in a subclass.

Hypergraphs

All of this is very nice…..**BUT…….**

Solving models of the form y=mx (one parameter) is NOT really interesting.

Trouble is…if you have 2 or more parameters, then the the minimal infeasible set size becomes 3 or more….then you are are talking about HYPERGRAPHS and Hypergraph classes (of which, comparatively, little is known).

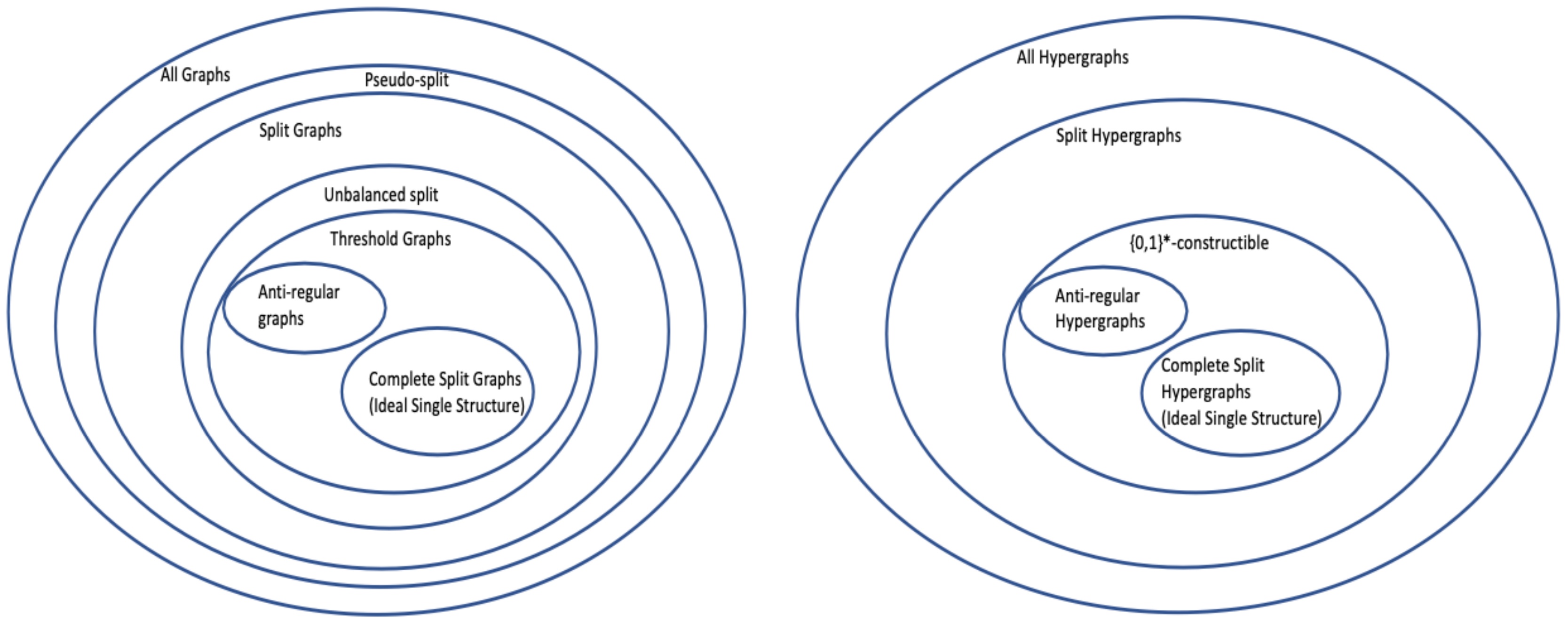


Fig. 1. (a) Threshold and Split Graph Class (b) Split Hypergraph and $\{0,1\}^*$-constructible Classes

Threshold graphs are special cases of split graphs that are themselves special cases of chordal graphs and pseudo-split classes. $\{0,1\}^*$-constructible hypergraphs are special cases of split hypergraphs, which contain the ideal single structure [4,5] as a special case. That subclass has already been studied under the name "complete split" hypergraph.

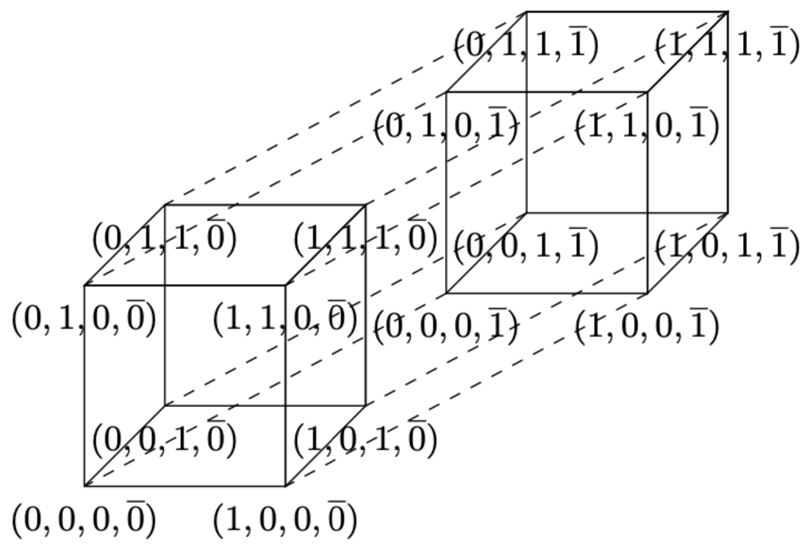
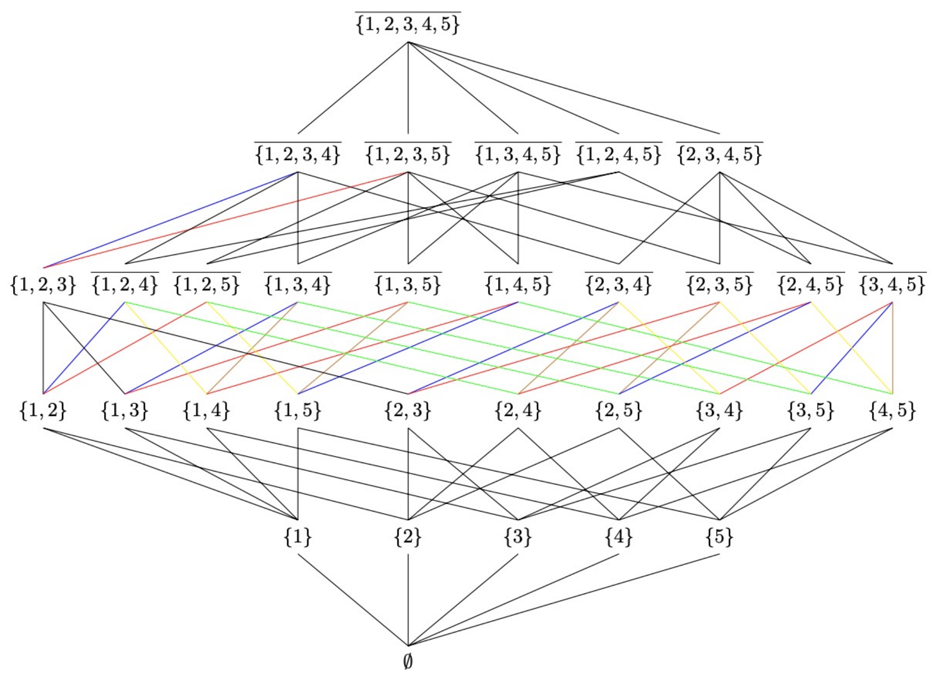


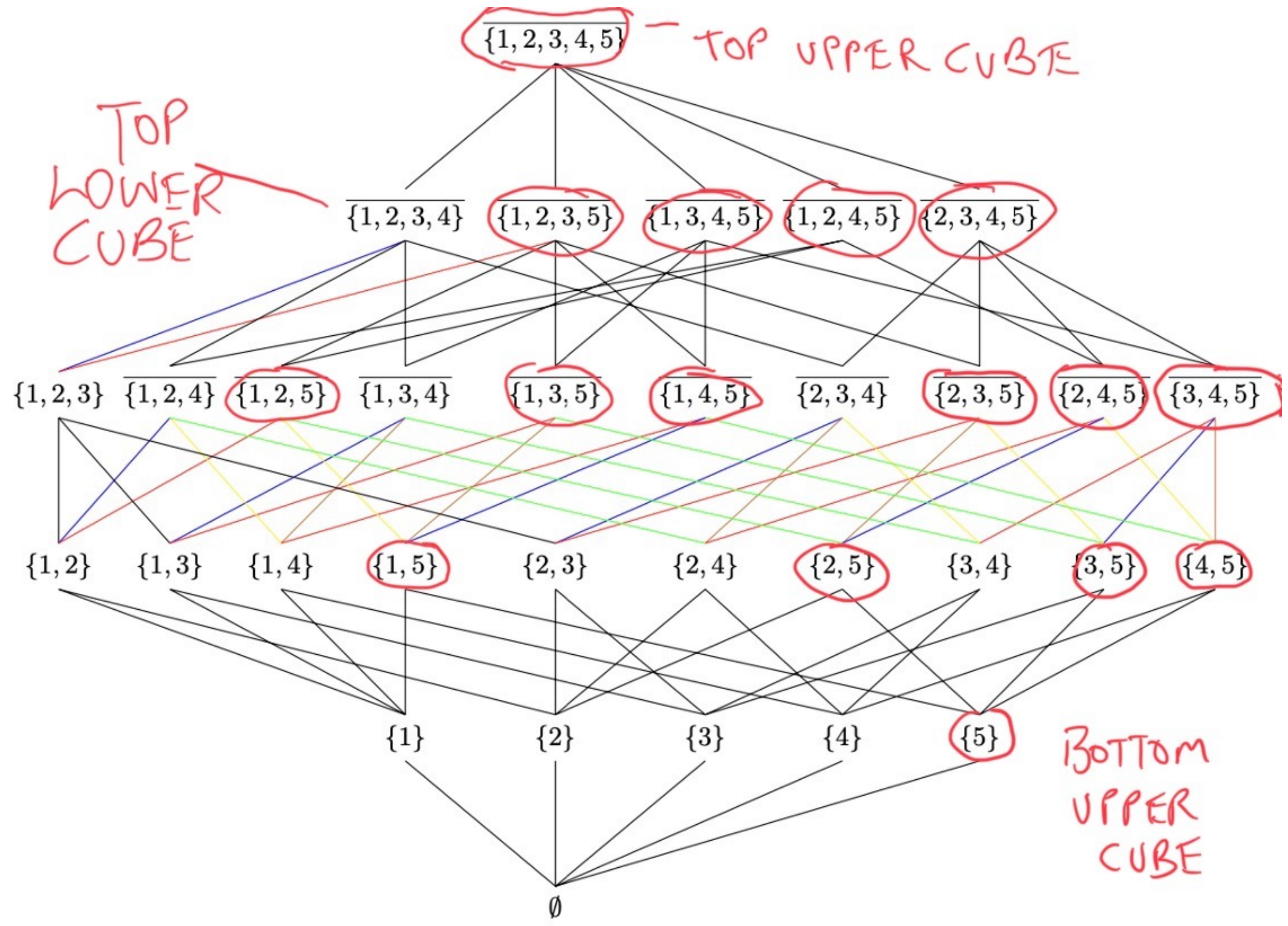
Fig. 2. "Upper" and "Lower" Subcubes
We show a 4-dimensional cube, distinguishing the 4th coordinate by placing an overbar on that coordinate. Also the dashed lines, show the edges running in the 4th direction. In the text, when adding the 4th vertex to the construction string, we would go from a 3-dimensional cube of subsets of vertices to a 4-dimensional cube. As this diagram depicts, we can think of the new cube as composed of an upper (top right) and lower (bottom left) 3-dimensional cubes. In the lower cube, note that since the 4th coordinate is always zero, this is isomorphic to the 3-dimensional cube we had before considering the 4th vertex.

A.3 3-Uniform Hypergraph - Ideal Single Structure

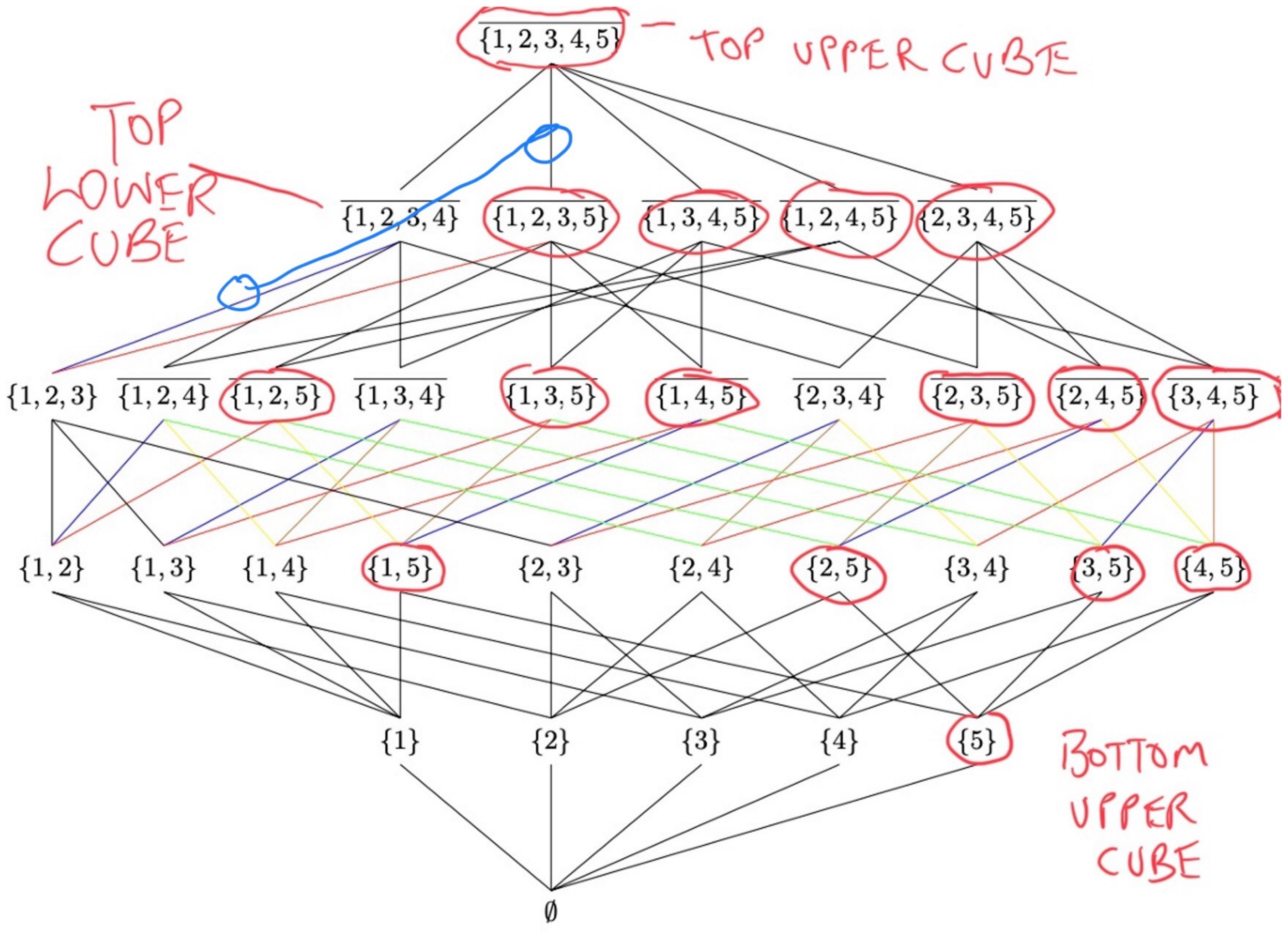


$\overline{\{1,2,3,4,5\}}$
$\overline{\{1,2,3,4\}}$ $\overline{\{1,2,3,5\}}$ $\overline{\{1,3,4,5\}}$ $\overline{\{1,2,4,5\}}$ $\overline{\{2,3,4,5\}}$
$\{1,2,3\}$ $\overline{\{1,2,4\}}$ $\overline{\{1,2,5\}}$ $\overline{\{1,3,4\}}$ $\overline{\{1,3,5\}}$ $\overline{\{1,4,5\}}$ $\overline{\{2,3,4\}}$ $\overline{\{2,3,5\}}$ $\overline{\{2,4,5\}}$ $\overline{\{3,4,5\}}$
$\{1,2\}$ $\{1,3\}$ $\{1,4\}$ $\{1,5\}$ $\{2,3\}$ $\{2,4\}$ $\{2,5\}$ $\{3,4\}$ $\{3,5\}$ $\{4,5\}$
$\{1\}$ $\{2\}$ $\{3\}$ $\{4\}$ $\{5\}$
$\emptyset$

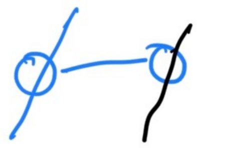
Fig. 4. Boolean Cube on 5 vertices (data points) - Independence Complex of a 3-Hypergraph
In the Boolean Cube we have annotated with an overbar the dependent sets of the 00011 hypergraph (describing the ideal single structure of size 3, and 2 outliers). This defines a Boolean Montone Function (indpendent sets assigned value 0 and dependent sets assigned value 1. For this function, edges that transition from 0 to 1 contribute an increment of 1 to the influence of the vertex (data point), which is added to the feasible subset to make the infeasible subset.
This is depicting example 2. Transitions are colour coded using the following palette v_1 green, v_2 yellow, v_3 brown, v_4 blue, v_5 red. Counting the edges of the approprate colour gives the influence counts of example 2.



Pivotal edges from "flipping membership of 5" are coloured in red – note these are the (only) edges between the lower and upper cubes…

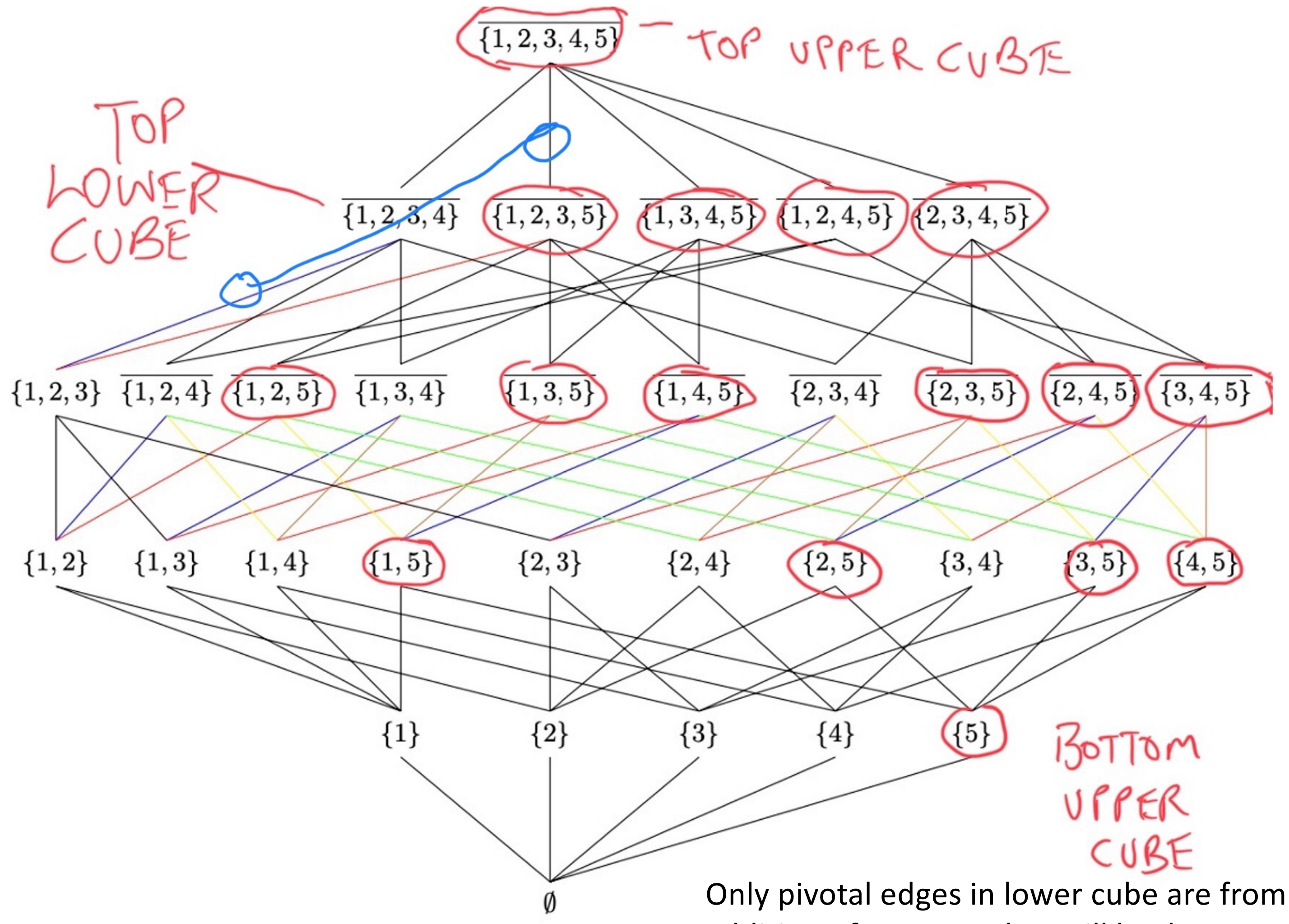


Pivotal edges from "flipping membership of 5" are coloured in red – note these are the (only) edges between the lower and upper cubes…

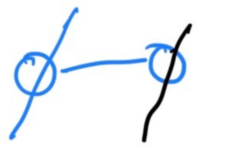


A MATCHING EDGE FROM LOWER CUBE TO UPPER CUBE

No longer a pivotal edge in upper cube ….



Pivotal edges from "flipping membership of 5" are coloured in red – note these are the (only) edges between the lower and upper cubes…



Only pivotal edges in lower cube are from size 2 to 3 nodes – where addition of ANY member will lead to a node of size three containing a 5

Incrementally computing influences of {0,1}* hypergraphs:

Keep track of “non-trivial” independent sets (alpha).
Initialise alpha to 0
Decode the construction string and update influences by…

Isolated: Set the influence of new vertex to zero and double all other influences. α becomes $2\alpha + C_p^{r-1}$.

Dominating: Set the influence of the new vertex to $C_p^{r-1} + \alpha$, and add $C_p^{r-1} - C_p^{r-2}$ to all other influences. α is left as it was.

Example 1. (Refer to figure 3). We take $p = 1$ (i.e., the hypergraph is now the special case of graph) and the construction string 00011, $k_1 = 3$ (the MaxCon solution, which is the size of the single structure, is 3) and there are two outliers. The infeasibility graph (the graph described by the construction string) is

$$\{\{1,4\},\{1,5\},\{2,4\},\{2,5\},\{3,4\},\{3,5\},\{4,5\}\} \tag{3}$$

Theorem 1 yields that the influences of the inliers (vertices 1, 2 and 3) are $C_1^4 - C_1^2 = 2$, and that the influences of the outliers (vertices 4 and 5) are $C_1^4 + \sum_{l=2}^{3} C_l^3 = 4 + 3 + 1 = 8$.

In the Boolean Cube of all subsets of five elements, the only subsets that are non-trivially feasible (do not contain an edge of the hypergraph 3 are

$$\{\{1,2,3\},\{1,2\},\{1,3\},\{2,3\}\} \tag{4}$$

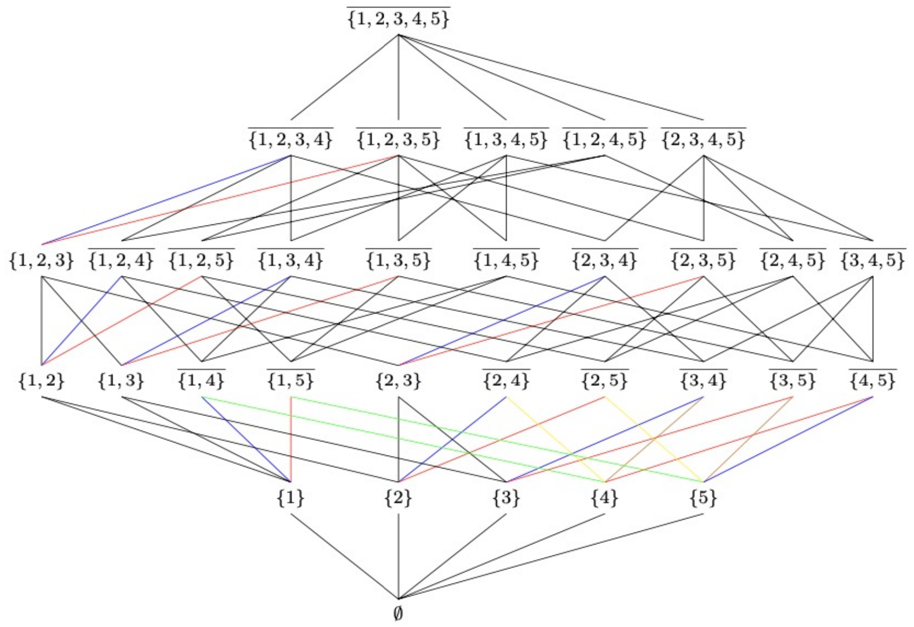
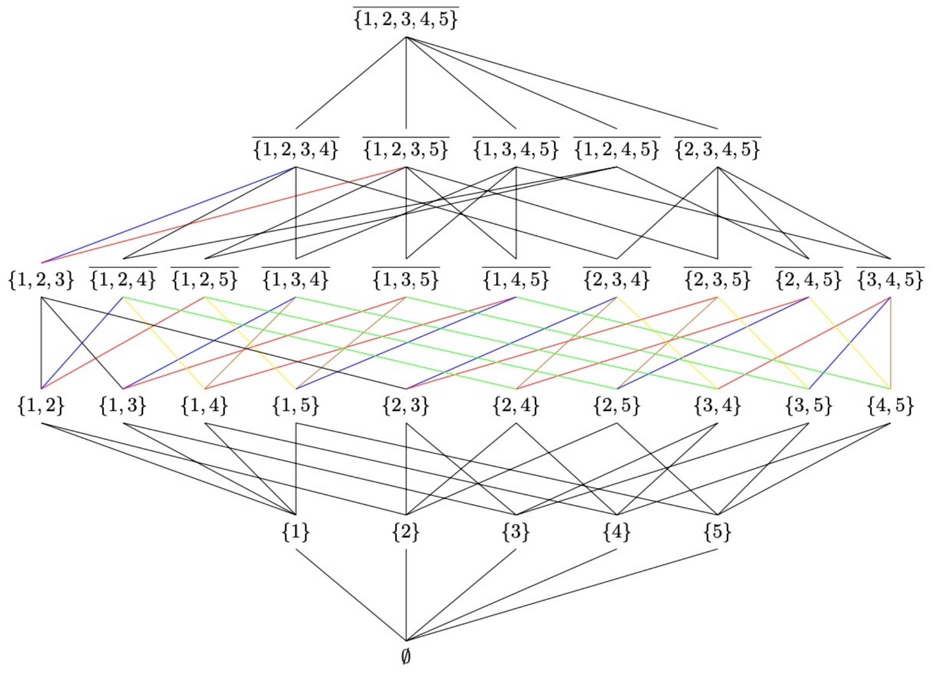


Fig. 3. Boolean Cube on 5 vertices (data points) - Independence Complex of a graph In the Boolean Cube we have annotated with an overbar the dependent sets of the 00011 graph (describing the ideal single structure of size 3, and 2 outliers). This defines a Boolean Monotone Function (independent sets assigned value 0 and dependent sets a value 1. For this function, edges that transition from 0 to 1 contribute an increment of 1 to the influence of the vertex (data point), which is added to the feasible subset to make the infeasible subset.
This is depicting example 1. Transitions are colour coded using the following palette v_1 green, v_2 yellow, v_3 brown, v_4 blue, v_5 red. Counting the edges of the appropriate colour gives the influence counts of example 1.

A.3 3-Uniform Hypergraph - Ideal Single Structure



$\overline{\{1,2,3,4,5\}}$
$\overline{\{1,2,3,4\}}$ $\overline{\{1,2,3,5\}}$ $\overline{\{1,3,4,5\}}$ $\overline{\{1,2,4,5\}}$ $\overline{\{2,3,4,5\}}$
$\{1,2,3\}$ $\overline{\{1,2,4\}}$ $\overline{\{1,2,5\}}$ $\overline{\{1,3,4\}}$ $\overline{\{1,3,5\}}$ $\overline{\{1,4,5\}}$ $\overline{\{2,3,4\}}$ $\overline{\{2,3,5\}}$ $\overline{\{2,4,5\}}$ $\overline{\{3,4,5\}}$
$\{1,2\}$ $\{1,3\}$ $\{1,4\}$ $\{1,5\}$ $\{2,3\}$ $\{2,4\}$ $\{2,5\}$ $\{3,4\}$ $\{3,5\}$ $\{4,5\}$
$\{1\}$ $\{2\}$ $\{3\}$ $\{4\}$ $\{5\}$
$\emptyset$

Fig. 4. Boolean Cube on 5 vertices (data points) - Independence Complex of a 3-Hypergraph
In the Boolean Cube we have annotated with an overbar the dependent sets of the 00011 hypergraph (describing the ideal single structure of size 3, and 2 outliers). This defines a Boolean Montone Function (indpendent sets assigned value 0 and dependent sets assigned value 1. For this function, edges that transition from 0 to 1 contribute an increment of 1 to the influence of the vertex (data point), which is added to the feasible subset to make the infeasible subset.
This is depicting example 2. Transitions are colour coded using the following palette v_1 green, v_2 yellow, v_3 brown, v_4 blue, v_5 red. Counting the edges of the approprate colour gives the influence counts of example 2.

In addition to allowing incremental calculation…

- The above rules show that
  - addition of isolated vertices preserves order of influences, doubles “gaps”
  - addition of dominating vertices preserves order of influences but does not double gaps
- The above are “un-normalized” edge counts. Estimation involves normalised (weighted by inverse of total edges for uniform measure) counts. The total number of edges doubles with adding a new vertex.

So….isolated vertices preserve “normalized gaps”.

Dominating vertices usually diminish gaps…so large numbers of outliers makes estimating the separation inlier/outlier harder…justifies the empirical decision to estimate – remove largest – re-estimate paradigm of CVPR2021/22.

Also degree of hypergraph matters….

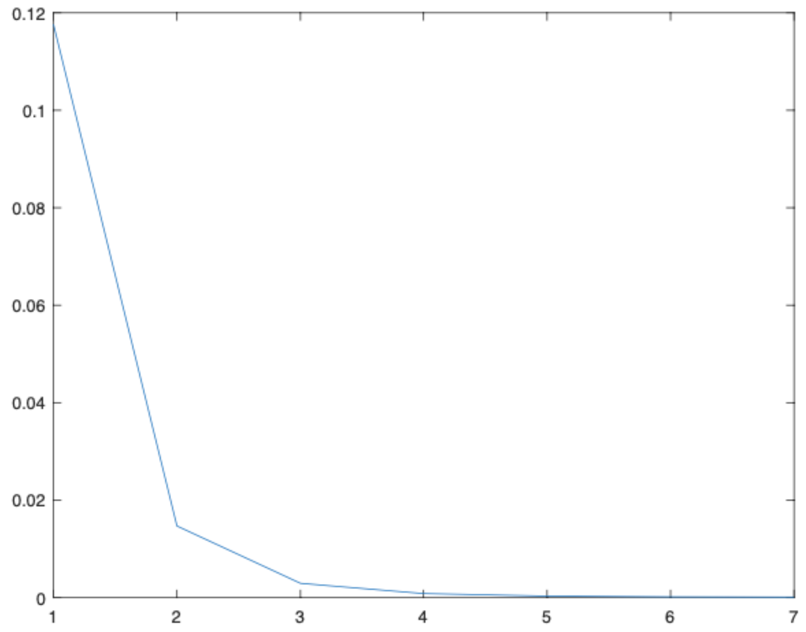


Fig. 7. Minimum inlier/outlier (normalised) influence gap vs degree of hypergraph

For every $\{0,1\}$-constructible graph on 17 vertices, restricted to initial string $p+1$ 0's (so there is a single MaxCon solution), and for $p = 1 \ldots 7$, we found the minimum inlier/outlier influence gap. Clearly, worst case (lowest gap of all hypergraphs at that p) deteriorates exponentially.

Limitations….

- Picture for **hyper**graphs is very incomplete (hence for anything other than toy problems!)
- Even where we know something about the structure of hypergraph classes relevant to our purposes – all computation seems heavy (testing membership of the class for example). In contrast, most of the graph classes that played a part in earlier slides have simple degree sequence tests for membership etc.
- So far, have limited experimental evidence of the relationship between graph class hierarchy and computational hardness (toy problems) or limited (to influence-type approaches) theoretical results.