Graph-based Complexity for Causal Effect by Empirical Plug-in

Rina Dechter (dechter@ics.uci.edu) , Anna K Raichev(araichev@uci.edu),
Jin Tian(Jin.Tian@mbzuai.ac.ae), and Alexander Ihler(ihler@ics.uci.edu)

Overview

We analyze the complexity of the Plug-In method for answering causal queries. While it's often assumed that evaluating high-dimensional probabilistic expressions is expensive, we show that the structure of the causal query (captured by a hierarchy hypergraph) can make evaluation efficient — often even linear in the data size. Our analysis connects the evaluation cost to graph properties like treewidth and hypertree width, providing a new complexity perspective for causal estimation.

Contributions:

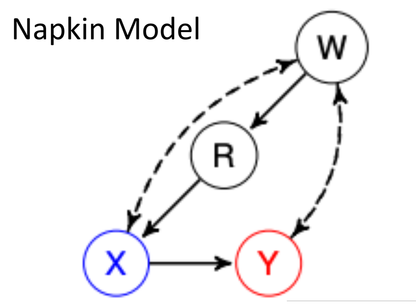
1. Associate an estimand expression with a hierarchy hypergraph and show how the hypertree widths additively determine complexity bounds.
2. Extend these bounds to sum-product queries involving ratios
3. Empirically show hypertree width effectively captures the actual time and memory costs.

Structural Causal Models

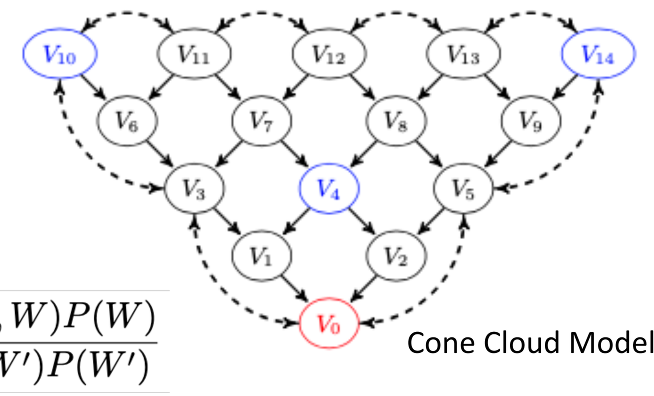
Structural Causal Model: $M = \langle U, V, F, P(U)\rangle$

- $U = \{U_1, \ldots, U_k\}$ set of unmeasurable latent variables
- $V = \{V_1, V_2, \ldots, V_n\}$ set of observable variables
- $F = \{f_i : V_i \in V\}$ $V_i \leftarrow f_i(V_i, PA_i)$; $PA_i \subseteq U \cup V \setminus V_i$
- $P(U)$ is a probability distribution over the exogenous variables

Napkin Model



$$P(Y \mid do(X)) = \frac{\sum_W P(X,Y|R,W)P(W)}{\sum_{W'} P(X|R,W')P(W')}$$



Cone Cloud Model

Problem

Given a causal diagram, an identifiable query $P(Y \mid do(X=x))$ and samples from the observed distribution, the task is to output the distribution of $P(Y \mid do(X=x))$.

Current Practice

1. Apply an algorithm for identifiability [Tian, 2002]
2. Generate an estimand, an algebraic expression involving only probabilistic expressions over the visible variables.
3. Estimate the estimand from the observational data. One possibility is via the *Plug-In method*, in which each term is estimated using the empirical distribution

Motivating Example

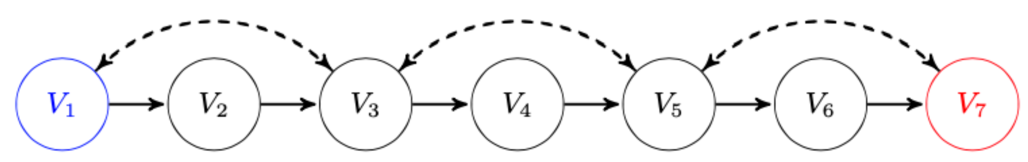


Figure 1: Chain Model with 7 observable variables and 3 latent variables

The identifiability algorithm will generate the expression:

$$P(V_7 \mid do(V_1)) = \sum_{V_2,V_3,V_4,V_5,V_6} P(V_6 \mid V_1, V_2, V_3, V_4, V_5) P(V_4 \mid V_1, V_2, V_3) P(V_2 \mid V_1)$$
$$\times \sum_{V_1'} P(V_7 \mid V_1', V_2, V_3, V_4, V_5, V_6) P(V_5 \mid V_1', V_2, V_3, V_4) P(V_3 \mid V_1', V_2) P(V_1')$$

Limitations

1. Conditional probabilities over a large number of variables can be computationally challenging - tables can be large.
2. Evaluating the estimand is potentially exponential in the number of variables of the largest function.

- Our work shows expressions are estimable in linear time: $hw = 1 \rightarrow O(t^1)$

Hyperwidth & Sparse Factors

The **hyperwidth** exploits the sparseness of the data

Sparse Factors

- **Tightness** of a function f, $t(f)$, is the number of non-zero configurations in its domain.
- **Tightness of a graphical model** with functions **F** is $t = \max_{f \in F} t(f)$.

Combination of Sparse Factors

A	B	f(A,B)
b	b	0.4
b	g	0.1
g	g	0.5

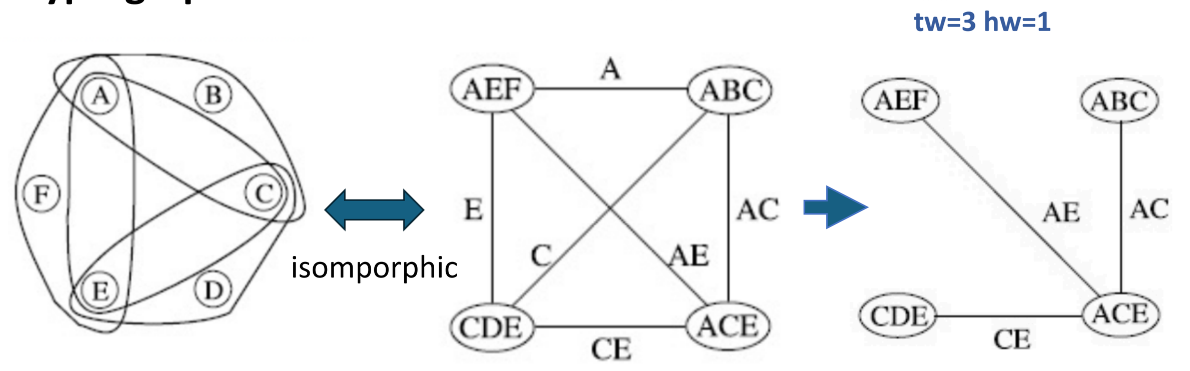
×

B	C	f(B,C)
b	b	0.2
g	g	0.8

=

A	B	C	f(A,B,C)
b	b	b	0.08
b	g	g	0.08
g	g	g	0.4

Hypergraph / Dual Graph / Hypertree (tw=3 hw=1)

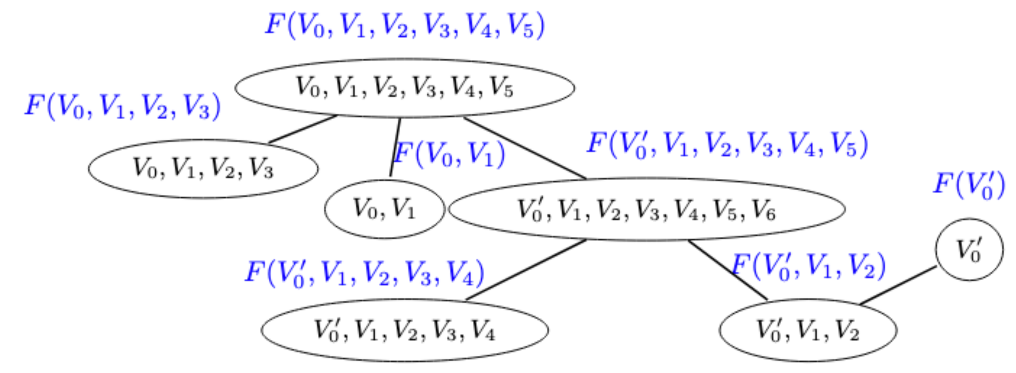


Plug-In hypertree Evaluation (PI-HTE)

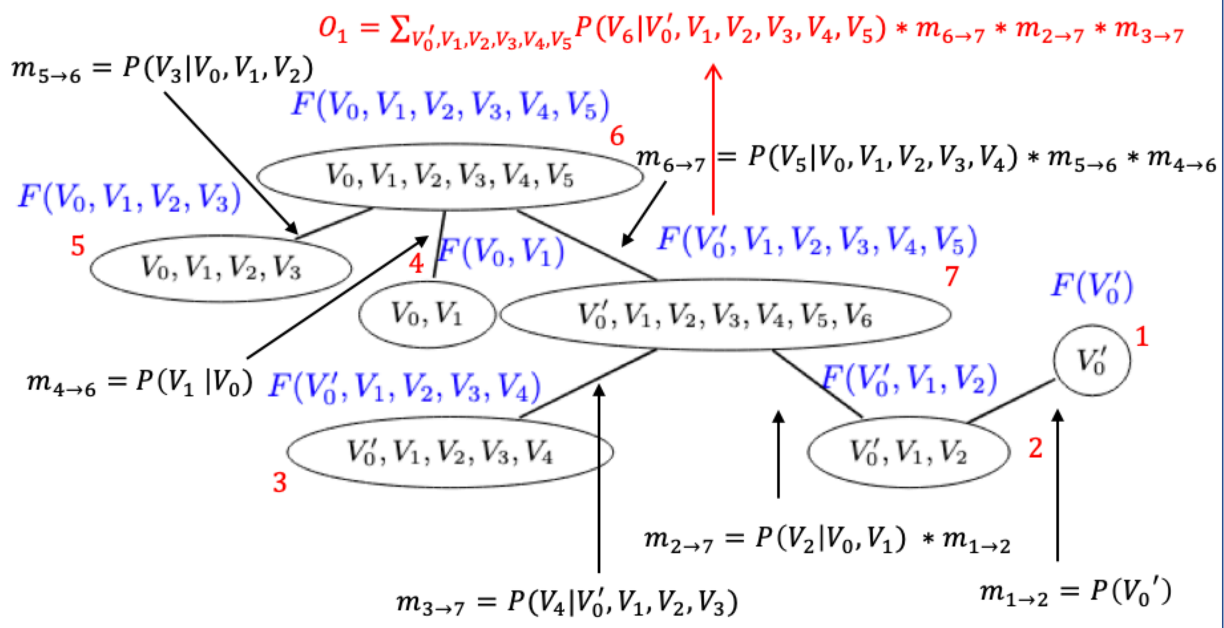
1.Flatten Sum-Products:

$$P(V_6 \mid do(V_0)) = \sum_{V_1,V_2,V_3,V_4,V_5,V_0'} P(V_5|V_0,V_1,V_2,V_3,V_4)$$
$$\times P(V_3|V_0,V_1,V_2)\, P(V_1|V_0) P(V_6|V_0',V_1,V_2,V_3,V_4,V_5)$$
$$\times P(V_4|V_0',V_1,V_2,V_3)\, P(V_2|V_0',V_1)\, P(V_0').$$

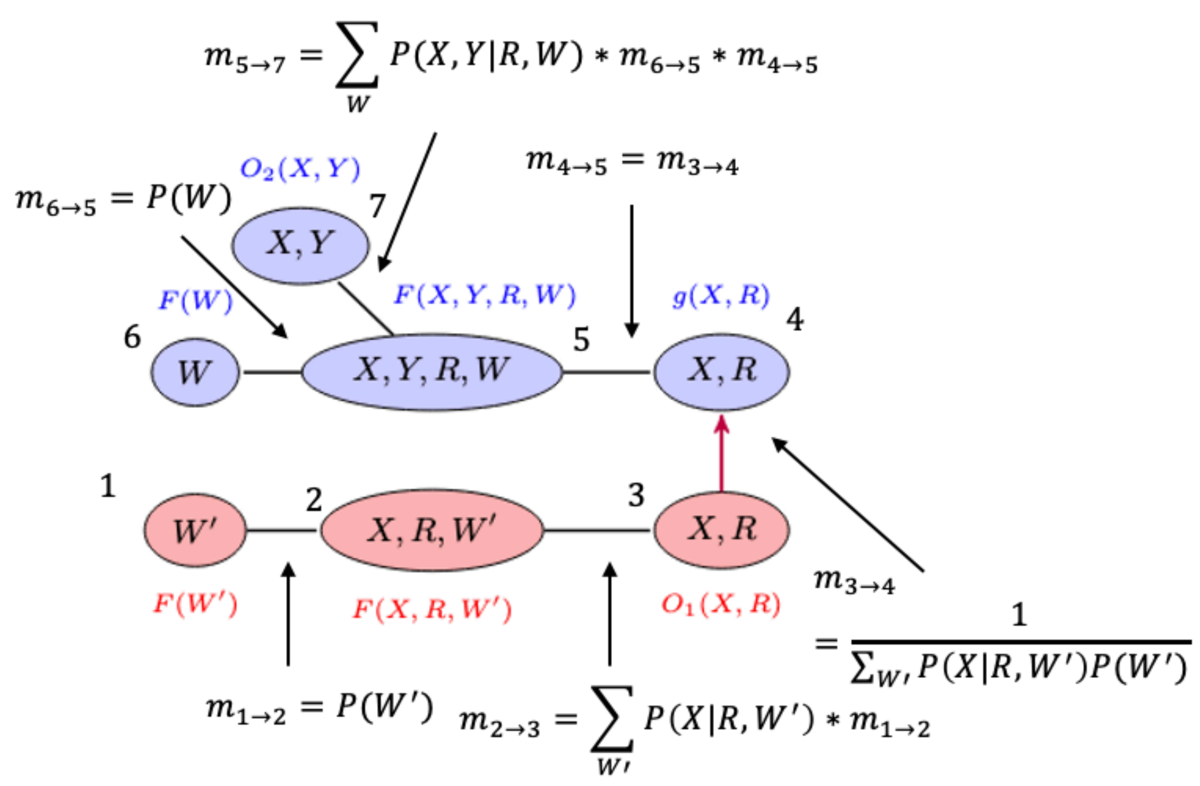
2.Generate Hyper tree:



3.Evaluate with CTE:



✻ **With Ratios we can use a Hierarchy Hypergraph:**

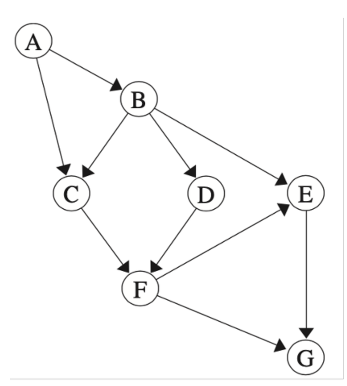


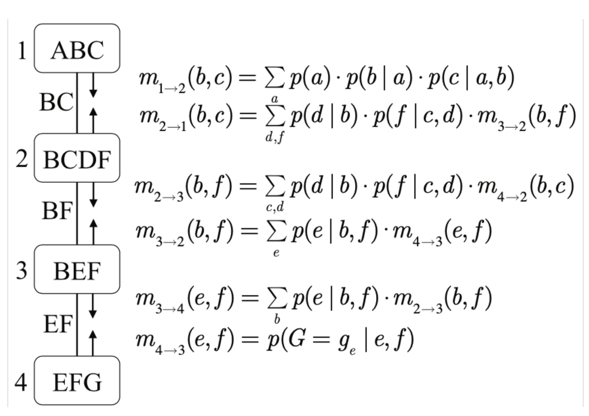
Complexity

- Time and memory are $O(n \cdot t^{\sum_{i=1}^{l} hw_i})$ with depth l

CTE Algorithm

Example





Complexity

- As a function of **hyperwidth**(sparse tables): $O(t^{hw})$ time & space
- As a function of **tree width**: $O(n \cdot k^{w+1})$ time & $O(n \cdot k^{w})$ space

Empirical Analysis

(a) Chain: $P(V_{99}|do(V_0))$
$|V| = 99$; $|U| = 49$ hw = 1, w = 98
$P(V_{98}|V0, \ldots V_{97})$- largest factor

#Samples	t	density	Max table size	time (s)
100	100	9.9×10^{-34}	100	11.2
200	200	2.0×10^{-33}	200	19.7
400	400	3.9×10^{-33}	400	31.4
800	800	7.9×10^{-33}	800	53.7
1,000	1,000	1.0×10^{-32}	1,000	69.5
1,600	1,600	1.6×10^{-32}	1,600	97.5
3,200	3,200	3.2×10^{-32}	3,200	192.4
6,400	6,400	6.4×10^{-32}	6,400	369.2
10,000	10,000	9.9×10^{-32}	10,000	608.8

$$density = \frac{\#entries\ in\ largest\ table}{\#of\ configurations\ in\ joint\ table} \quad k = 4$$

Treewidth bound: $O(k^{98})$ **Hyperwidth bound:** $O(t^1)$

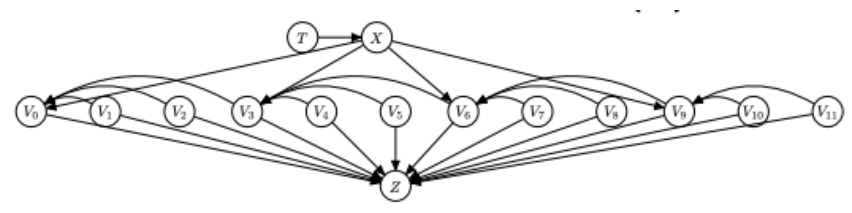
(b) Cone Cloud $P(V_0|do(V_{14}, V_{10}, V_4))$
$|V| = 15$; $|U| = 8$; $k = 10$; hw = 2, w = 14
largest factor $P(V_0|V_1 \ldots V_{14})$

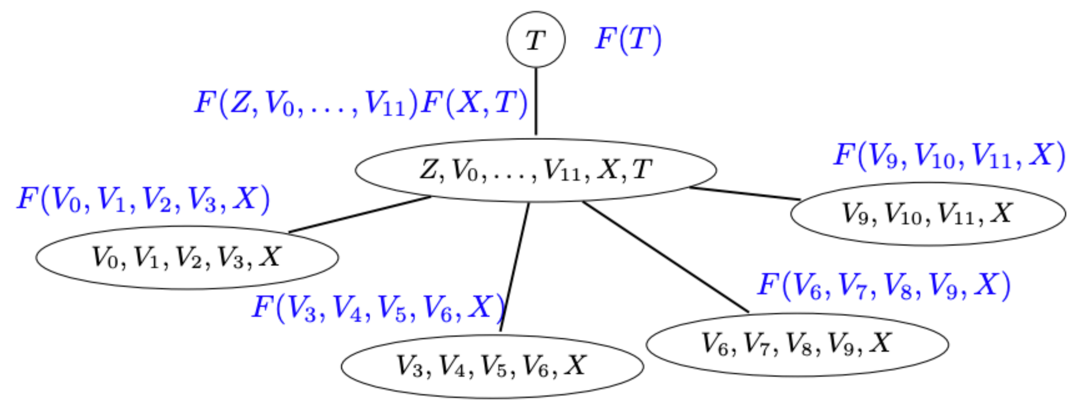
#Samples	t	density	Max table size	time (s)
100	100	1×10^{-13}	9,900	2.2
200	200	2×10^{-13}	39,203	3.3
400	400	4×10^{-13}	152,877	7.3
800	800	8×10^{-13}	589,815	28.8
1,000	1,000	1×10^{-12}	882,604	39.1
1,600	1,600	1.6×10^{-12}	2,206,528	84.8
3,200	3,200	3.2×10^{-12}	7,553,700	280.3
6,400	6,400	6.4×10^{-12}	21,949,125	788.74
10,000	10,000	1×10^{-11}	40,404,630	1,594.2

$k = 10$ **Treewidth bound**: $O(k^{15})$ **Hyperwidth bound:** $O(t^2)$

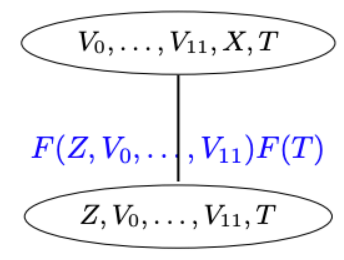
Impact of Different Hypertree Decompositions

2-Layer Model





(a) 2-Layer model hypertree decomposition with hw=2



(b) 2-Layer model hypertree decomposition with hw=5

(a) 2-Layer Model, $P(Z|do(T))$
$|V| = 15$; with hypertree decomposition w/ hw = 2, w = 14

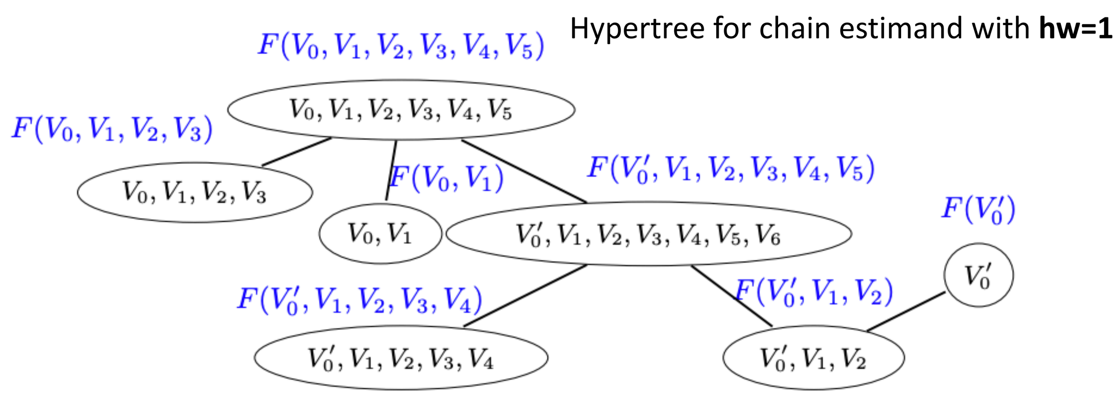
#Samples	t	Max table size	time (s)
100	100	636	1.73
200	200	1,679	1.78
400	400	3,600	2.00
800	800	7,200	2.36
1,000	1,000	9,000	2.71

(b) 2-Layer Model, $P(Z|do(T))$
$|V| = 15$; with hypertree decomposition w/ hw = 5, w = 13

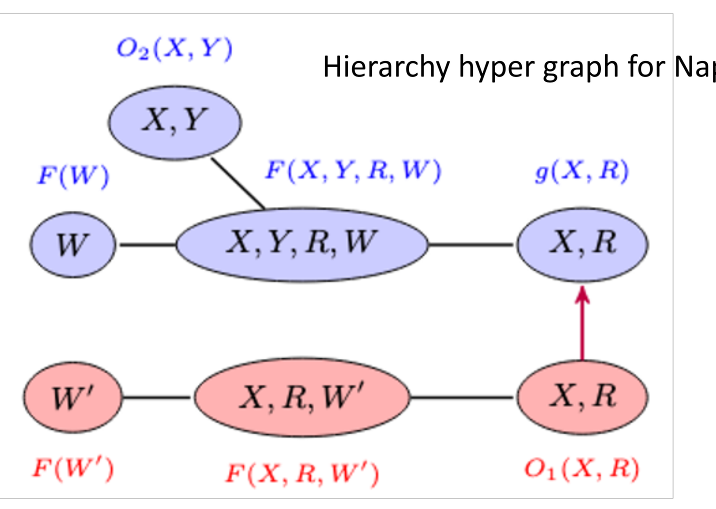
#Samples	t	Max table size	time (s)
100	100	8,489	2.27
200	200	62,447	4.23
400	400	812,926	36.85
800	800	8,949,996	348.84
1,000	1,000	20,188,458	775.93

Hypertrees & Hierarchy Hypergraph

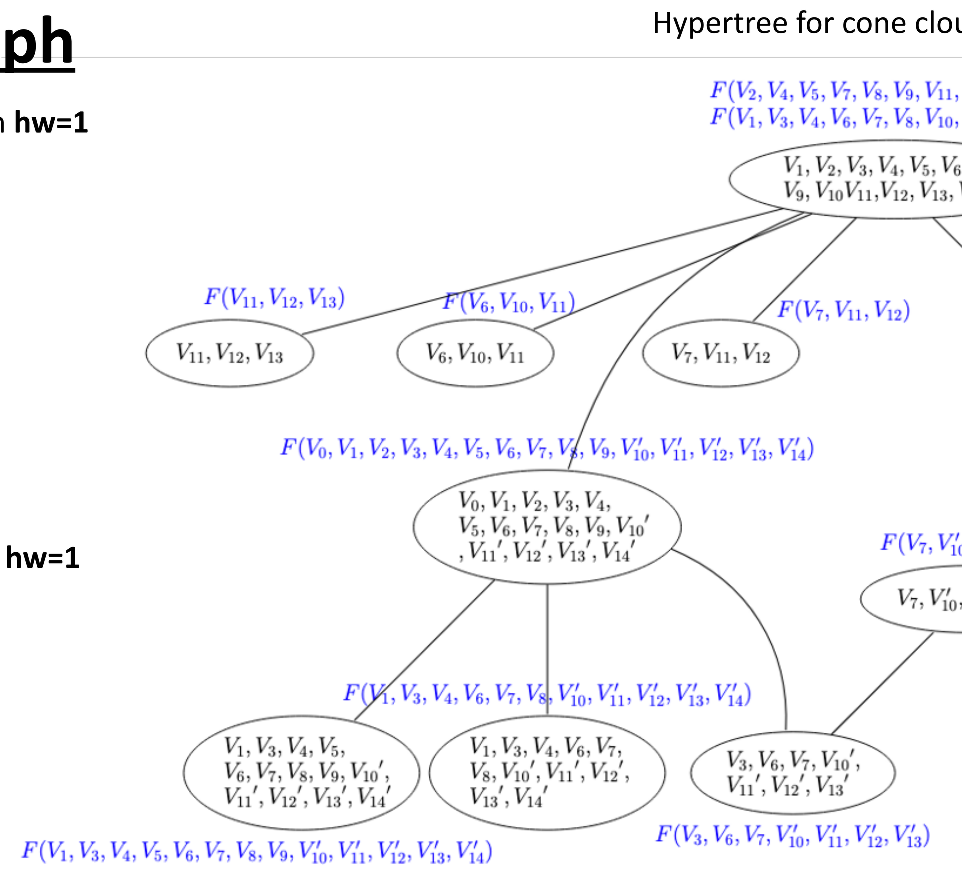
Hypertree for chain estimand with **hw=1**



Hierarchy hyper graph for Napkin estimand with **hw=1**



Hypertree for cone cloud estimand with **hw=2**



Notation

- w - treewidth
- hw- hypertree width
- t- tightness
- k - maximum domain size