Unlabeled Compression Schemes for Maximum Classes

Dima Kuzmin

Manfred K. Warmuth

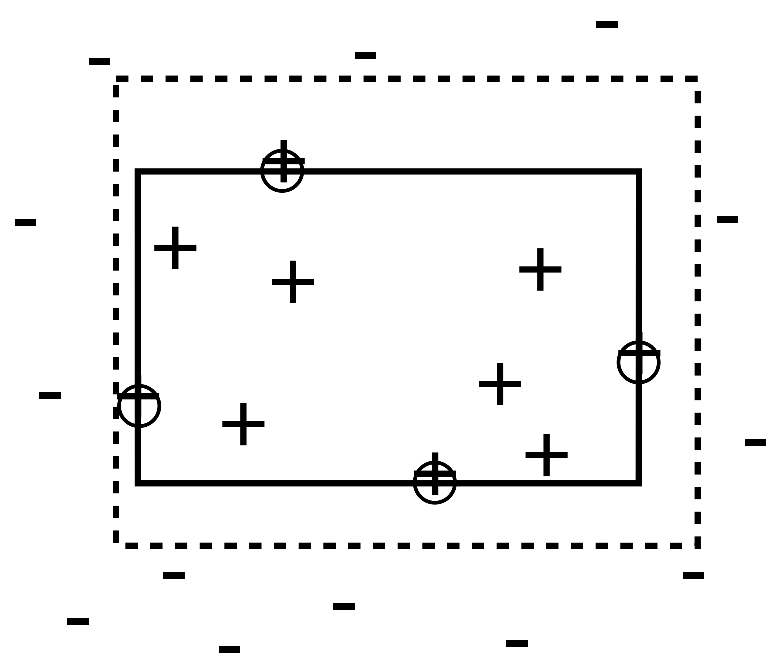
University of California - Santa Cruz, USA

some work done while authors were visiting National ICT Australia

June 29, COLT 2005

Compression Scheme

- Sample consistent with some axis-parallel rectangle
- Compress to: leftmost, rightmost, uppermost and lowermost positive points (circled)
- Reconstruct all labels based on the smallest rectangle containing the compression set (solid line)



Compression Schemes ito 2 functions

- **Compression function** maps samples into subsamples
- **Reconstruction function** maps compression sets into hypotheses consistent with the original sample
- Variations: labeled, unlabeled, lossless, lossy, additional bits, etc.

Why Study Compression Schemes?

- Size of compression scheme is alternate measure of complexity
- Inspires good practical algorithms
- Simple proofs of sample size and generalization bounds ([LW86], [FW95],[Lang05])
- Sample size bound for size k compression scheme:

$$m(\epsilon, \delta) = \Omega\left(\frac{k}{\epsilon} \log \frac{1}{\epsilon} + \frac{\log \frac{1}{\delta}}{\epsilon}\right)$$

Why Study Compression Schemes?

- Size of compression scheme is alternate measure of complexity
- Inspires good practical algorithms
- Simple proofs of sample size and generalization bounds ([LW86], [FW95],[Lang05])
- Sample size bound for class of VC dimension d:

$$m(\epsilon, \delta) = \Omega(\frac{d}{\epsilon} \log \frac{1}{\epsilon} + \frac{\log \frac{1}{\delta}}{\epsilon})$$

Conjecture

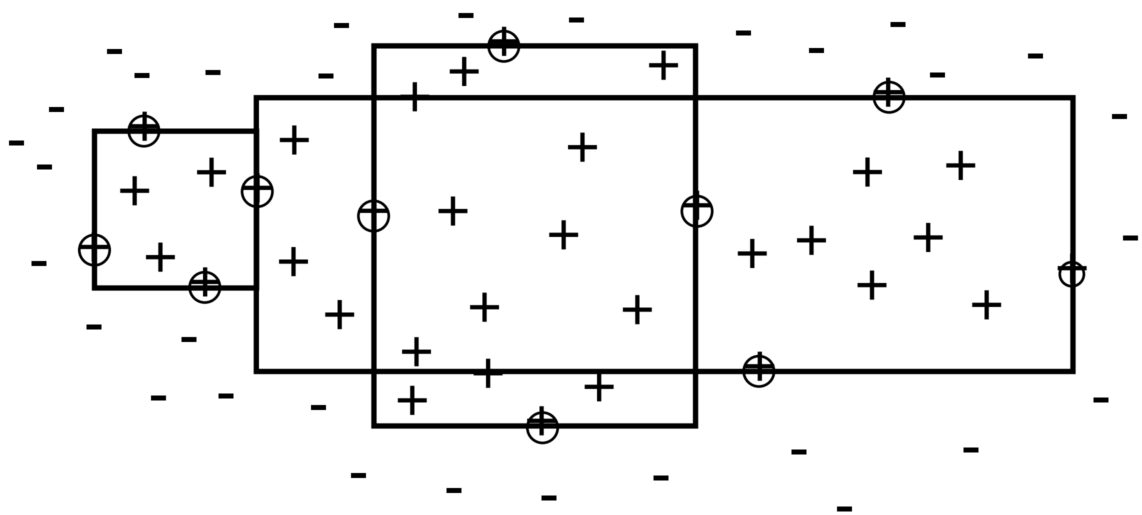
- VC dimension of rectangles is **4**
- We used size $\leq$**4** compression sets
- One of the most tantalizing open problems in learning theory:

For any concept class with VC dimension d there always is a compression scheme that keeps at most d points

[FW95]

Which Practical Algorithms?

- Cover positive points with rectangles



- Compresses to subsample and small number of bits [MST02]
- SVMs: Compression set is set of essential support vectors

[vLBS04]

Maximum Classes

- Sauer's Lemma provides an upper bound on size of a concept class with VCdim d and n instances:

$$|C| \leq \sum_{i=0}^{d} \binom{n}{i} = \binom{n}{\leq d}$$

- Classes are **maximum** when $\leq$ becomes $=$
- Example: class of all subsets of size up to d
- Our results are currently only for such classes

What is known?

- Compression scheme for maximum concept classes that compresses to labeled sets of points of size $= d$ [FW95]

Here:

- We compress maximum classes to unlabeled sets of points of size $\leq d$
- New scheme is tight and new combinatorics
- Connection to the 1-inclusion graph algorithm

Our Scheme

- Each concept represented by subset of size $\leq d$

	x_1	x_2	x_3	x_4	Representative $r(c)$
c_1	0	0	0	0	$\emptyset$
c_2	0	0	$\underline{1}$	0	$\{x_3\}$
c_3	0	0	1	$\underline{1}$	$\{x_4\}$
c_4	0	$\underline{1}$	0	0	$\{x_2\}$
c_5	0	1	$\underline{0}$	$\underline{1}$	$\{x_3, x_4\}$
c_6	0	$\underline{1}$	$\underline{1}$	0	$\{x_2, x_3\}$
c_7	0	$\underline{1}$	1	$\underline{1}$	$\{x_2, x_4\}$
c_8	$\underline{1}$	0	0	0	$\{x_1\}$
c_9	$\underline{1}$	0	$\underline{1}$	0	$\{x_1, x_3\}$
c_{10}	$\underline{1}$	0	1	$\underline{1}$	$\{x_1, x_4\}$
c_{11}	$\underline{1}$	$\underline{1}$	0	0	$\{x_1, x_2\}$

Compressing and Reconstructing

- Sample $(\mathbf{x_3}, \mathbf{1}), (\mathbf{x_4}, \mathbf{0})$

	x_1	x_2	x_3	x_4
c_1	0	0	0	0
c_2	0	0	$\underline{1}$	0
c_3	0	0	1	$\underline{1}$
c_4	0	$\underline{1}$	0	0
c_5	0	1	$\underline{0}$	$\underline{1}$
c_6	0	$\underline{1}$	$\underline{1}$	0
c_7	0	$\underline{1}$	1	$\underline{1}$
c_8	$\underline{1}$	0	0	0
c_9	$\underline{1}$	0	$\underline{1}$	0
c_{10}	$\underline{1}$	0	1	$\underline{1}$
c_{11}	$\underline{1}$	$\underline{1}$	0	0

Compressing and Reconstructing

- Sample $(\mathbf{x_3}, \mathbf{1}), (\mathbf{x_4}, \mathbf{0})$
- Many consistent concepts

	x_1	x_2	x_3	x_4
c_1	0	0	0	0
c_2	0	0	$\underline{1}$	0
c_3	0	0	1	$\underline{1}$
c_4	0	$\underline{1}$	0	0
c_5	0	1	$\underline{0}$	$\underline{1}$
c_6	0	$\underline{1}$	$\underline{1}$	0
c_7	0	$\underline{1}$	1	$\underline{1}$
c_8	$\underline{1}$	0	0	0
c_9	$\underline{1}$	0	$\underline{1}$	0
c_{10}	$\underline{1}$	0	1	$\underline{1}$
c_{11}	$\underline{1}$	$\underline{1}$	0	0

Compressing and Reconstructing

- Sample $(\mathbf{x_3}, \mathbf{1}), (\mathbf{x_4}, \mathbf{0})$
- Many consistent concepts
- Exactly one has its representation set completely inside the sample

	x_1	x_2	x_3	x_4
c_1	0	0	0	0
c_2	0	0	$\underline{1}$	0
c_3	0	0	1	$\underline{1}$
c_4	0	$\underline{1}$	0	0
c_5	0	1	$\underline{0}$	$\underline{1}$
c_6	0	$\underline{1}$	$\underline{1}$	0
c_7	0	$\underline{1}$	1	$\underline{1}$
c_8	$\underline{1}$	0	0	0
c_9	$\underline{1}$	0	$\underline{1}$	0
c_{10}	$\underline{1}$	0	1	$\underline{1}$
c_{11}	$\underline{1}$	$\underline{1}$	0	0

Compressing and Reconstructing

- Sample $(\mathbf{x_3}, \mathbf{1}), (\mathbf{x_4}, \mathbf{0})$
- Many consistent concepts
- Exactly one has its representation set completely inside the sample
- Compress to this representation set - $\{x_3\}$
- Reconstruct based on represented concept

	x_1	x_2	x_3	x_4
c_1	0	0	0	0
c_2	0	0	$\underline{1}$	0
c_3	0	0	1	$\underline{1}$
c_4	0	$\underline{1}$	0	0
c_5	0	1	$\underline{0}$	$\underline{1}$
c_6	0	$\underline{1}$	$\underline{1}$	0
c_7	0	$\underline{1}$	1	$\underline{1}$
c_8	$\underline{1}$	0	0	0
c_9	$\underline{1}$	0	$\underline{1}$	0
c_{10}	$\underline{1}$	0	1	$\underline{1}$
c_{11}	$\underline{1}$	$\underline{1}$	0	0

Condition on Representatives

Two different concepts c, c' clash wrt r if
$c \mid r(c) \cup r(c') = c' \mid r(c) \cup r(c')$

10<u>01</u> and 110<u>1</u> clash since 01 = 01

00<u>01</u> and <u>1</u>00<u>1</u> don't clash since $001 \neq 101$

r must have two properties:

1. r is a bijection between C and sets of points of size at most d
2. no two concepts clash

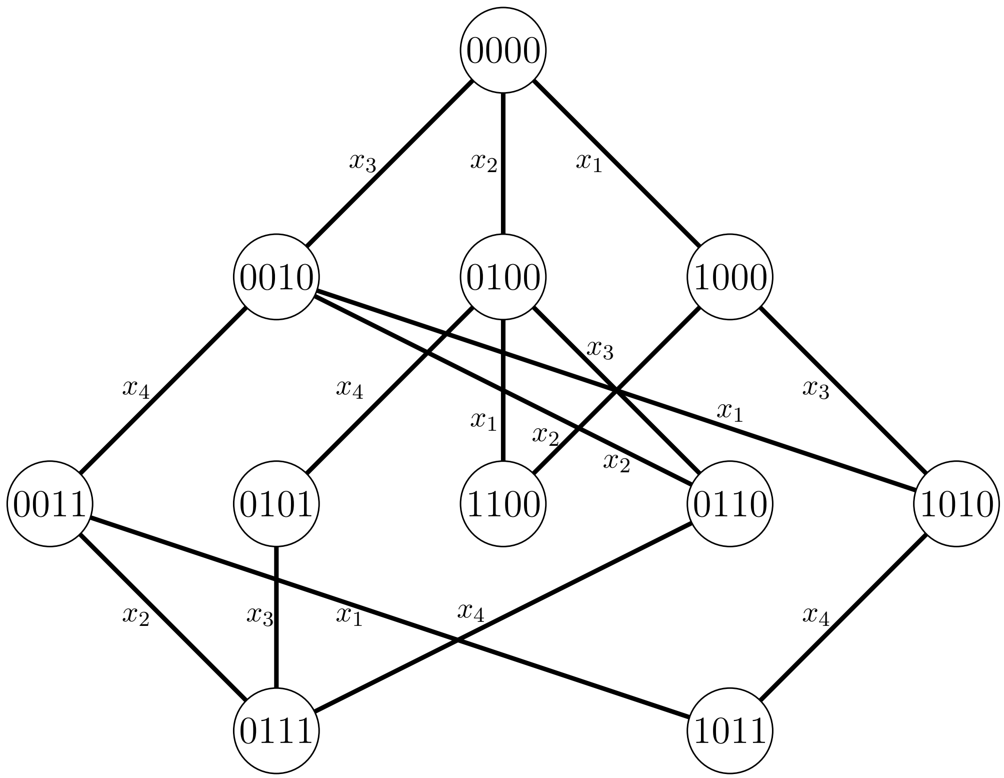
Central Lemma

The following two statements are equivalent:

1. No two concepts clash wrt r
2. For all samples s from C, there is at most one concept $c \in C$ that is consistent with s and $r(c)$ contains only points from s

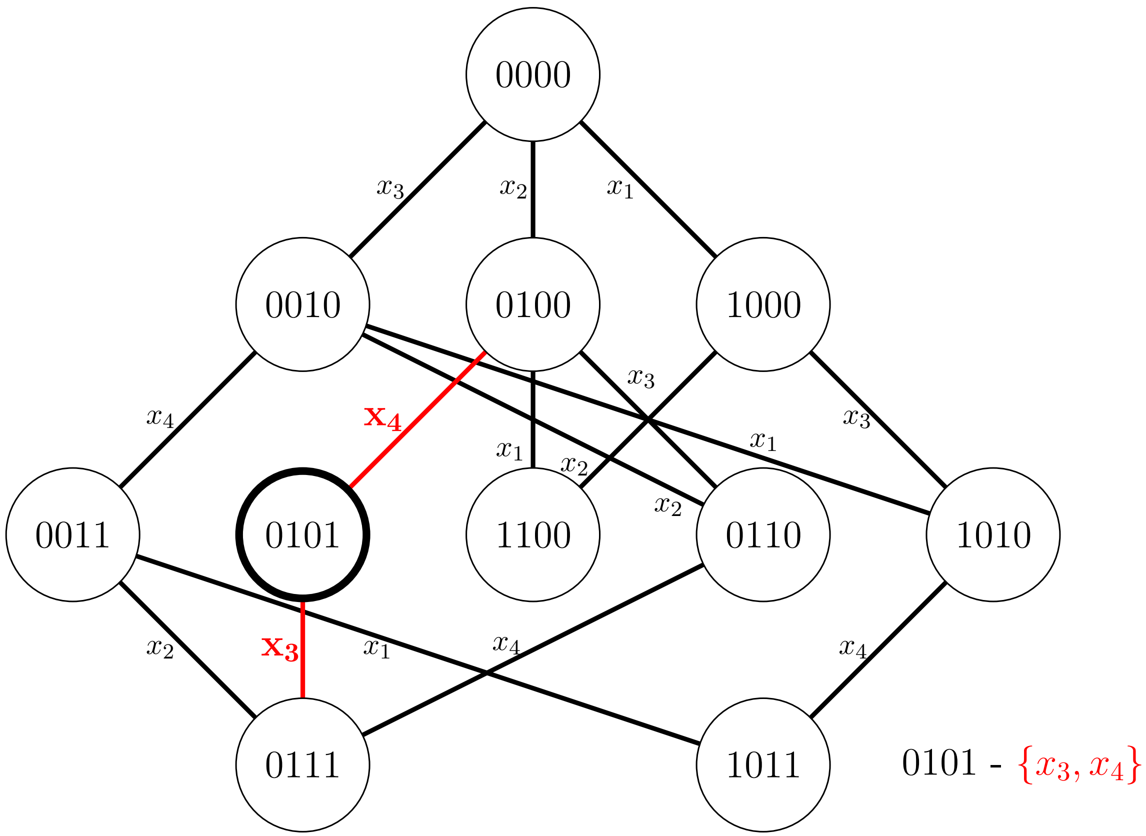
One-inclusion Graph

- Nodes are concepts
- Edge between two concepts differing on one instance



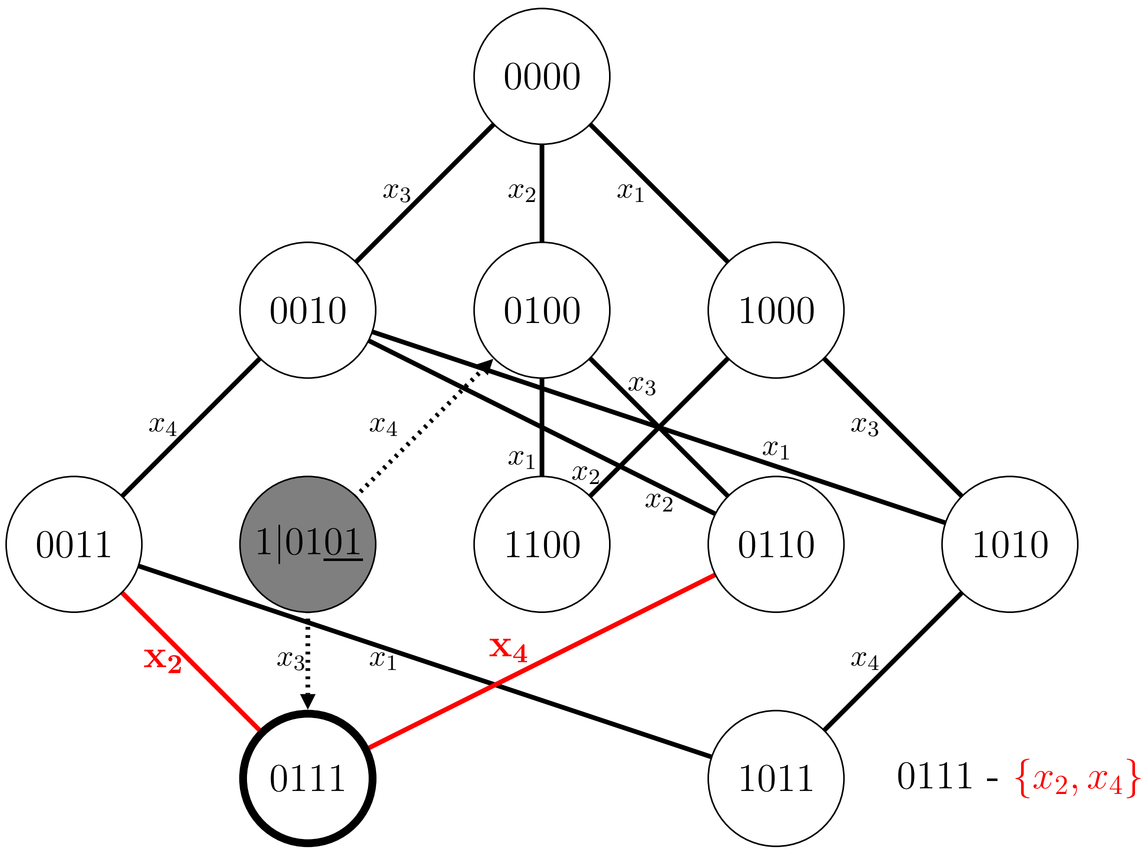
How to Find Representations

- Peel away any lowest degree vertex
- Labels of incident edges become representation sets



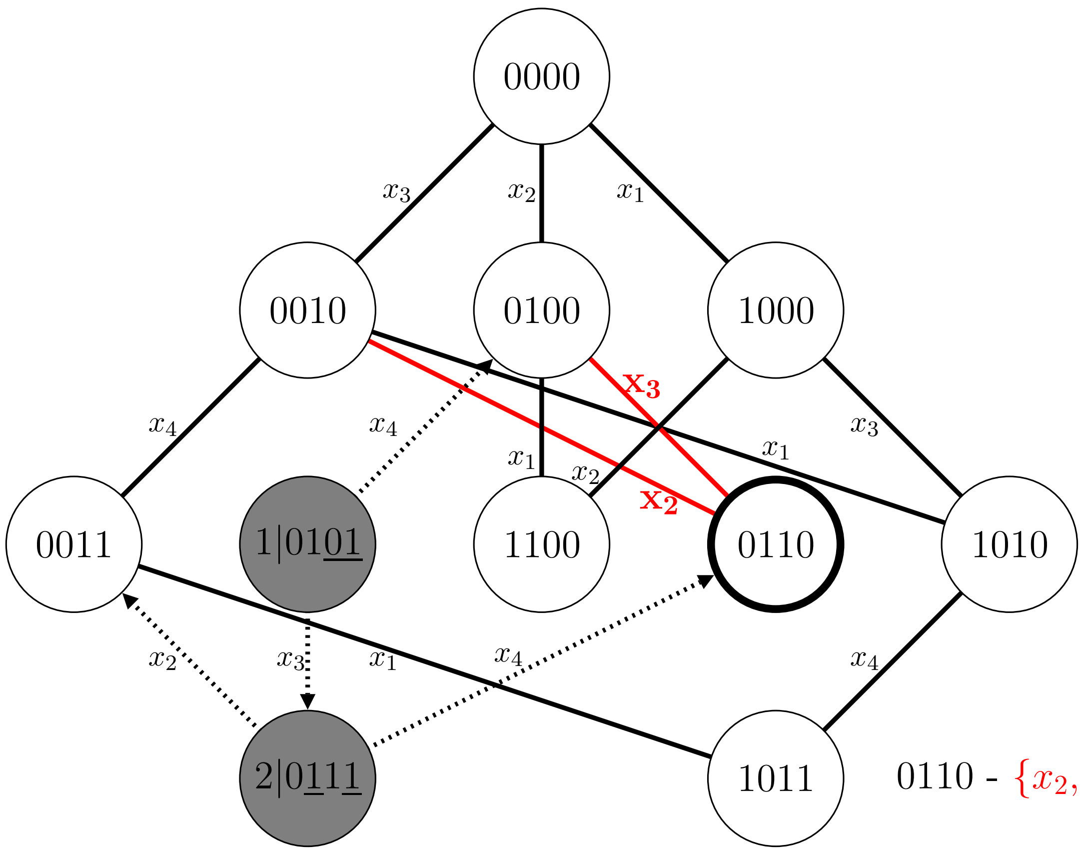
How to Find Representations

- Now peel away next lowest degree vertex
- Labels of incident edges become representation sets



How to Find Representations

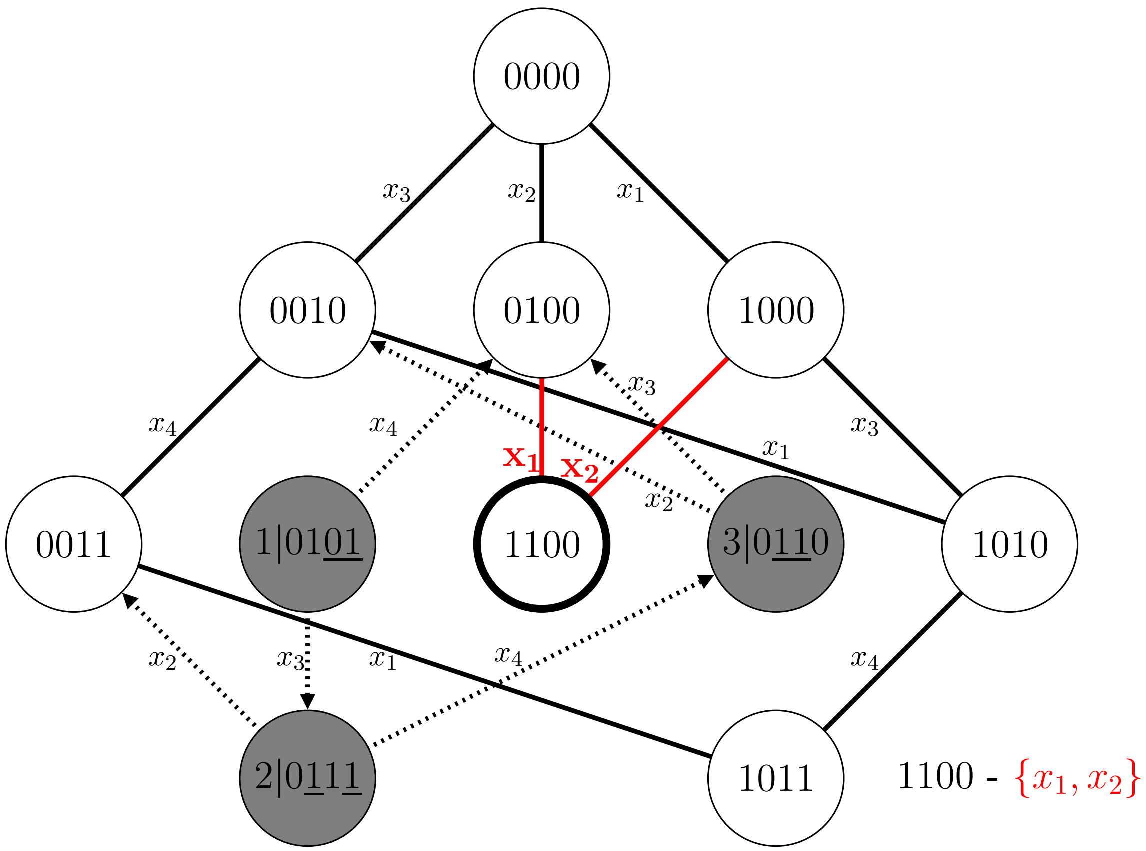
- Now peel away next lowest degree vertex
- Labels of incident edges become representation sets



0110 - $\{x_2, x_3\}$

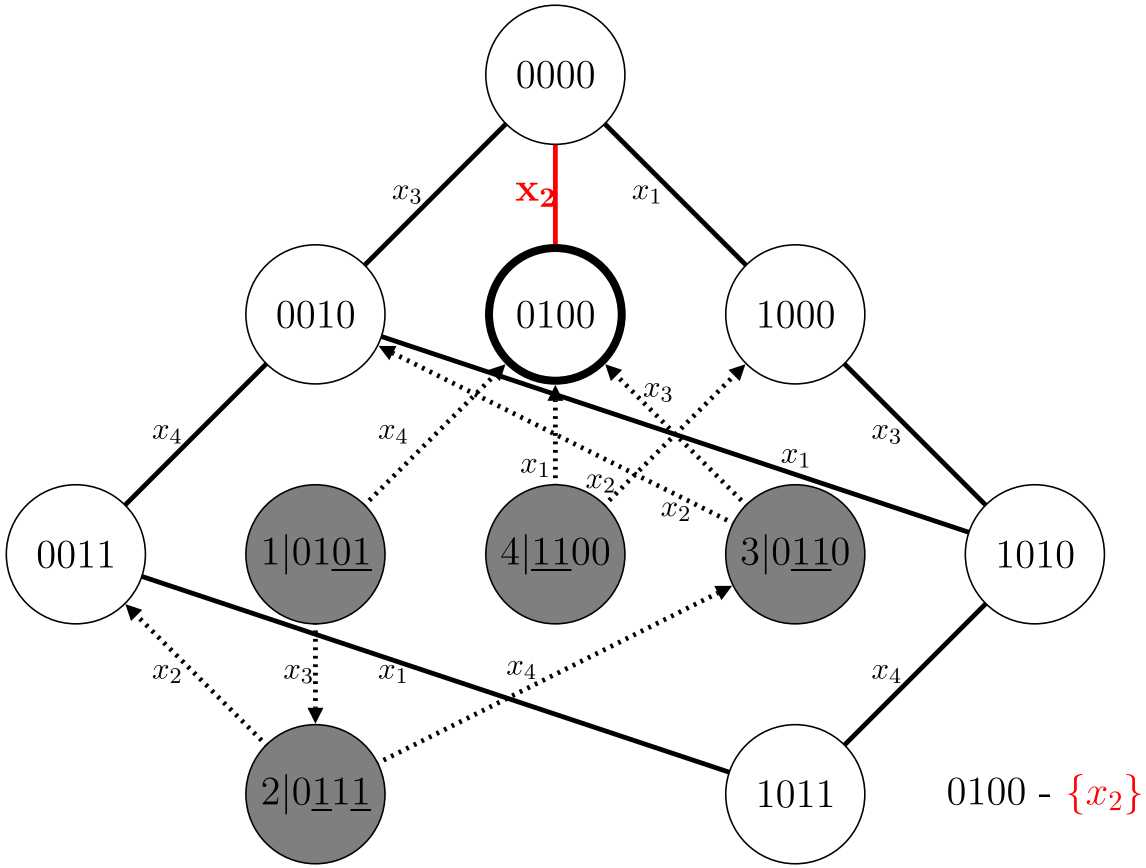
How to Find Representations

- Now peel away next lowest degree vertex
- Labels of incident edges become representation sets



How to Find Representations

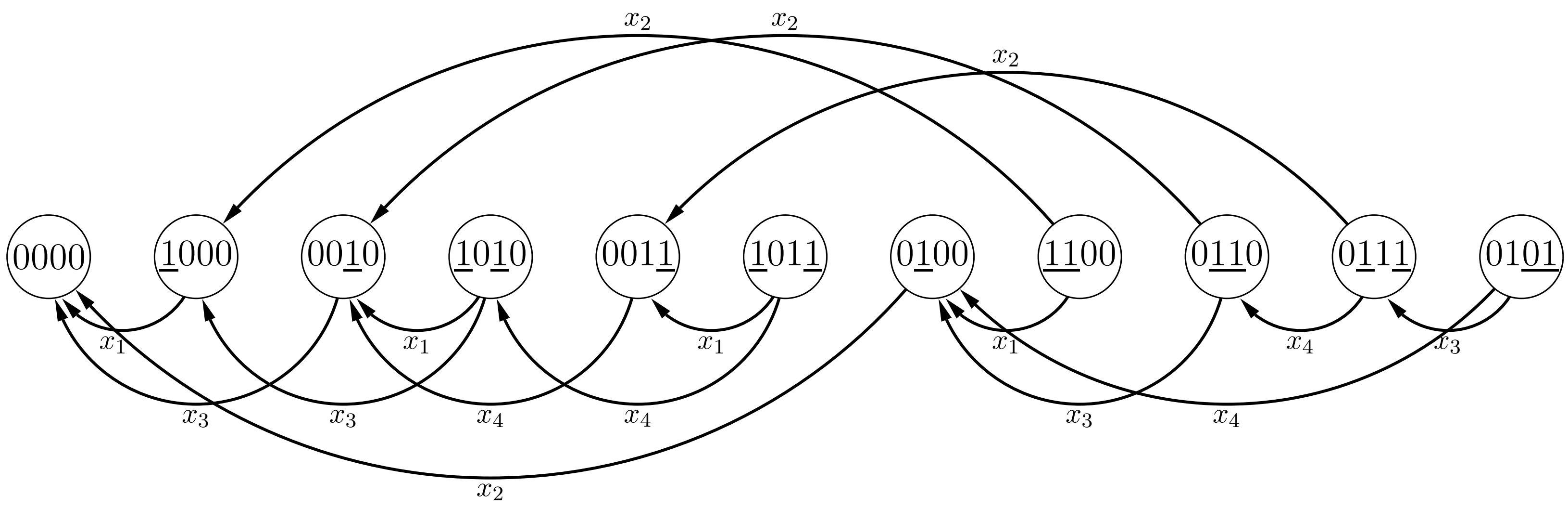
- Now peel away next lowest degree vertex
- Labels of incident edges become representation sets



The Good and the Bad

:-) Algorithm implies a d-orientation of one-inclusion graph

:-) Also provides topological order

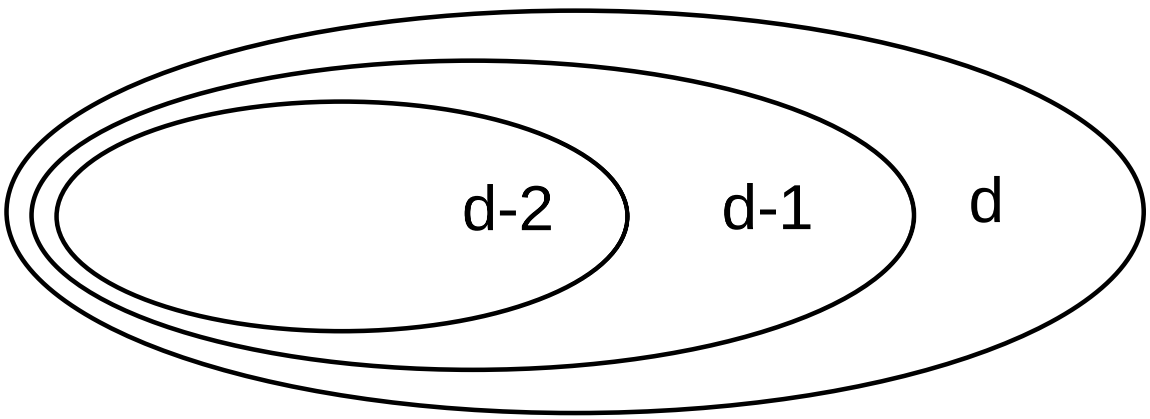


:-(No proof of correctness for peeling algorithm

Alternate Recursive Algorithm

:-) Has correctness proof

:-) Additional property: for $k \leq d$, the concepts that are represented by sets of size up to k form a maximum class of VC dimension k.



Recursive Compression Scheme Algorithm

Goes beyond the scope of this talk! :-(

Any Unlabeled Scheme Implies d-orientation

$$\underset{r(c)}{c} \overset{x}{\text{———}} \underset{r(c')}{c'}$$

If $x \in r(c)$, then $c \xrightarrow{x} c'$

If $x \in r(c')$, then $c \xleftarrow{x} c'$

No other cases

Provides d-orientation of the one-inclusion graph

One-inclusion Graph Algorithm [HLW94]

- Given $\underbrace{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_t, y_t)}_{training}, \underbrace{\mathbf{x}_{t+1}}_{test}$
  Build one-inclusion graph for $x_1, \ldots x_{t+1}$
- d-orient graph and predict with orientation
- Optimal

Wrap Up

- Min-degree conjecture: min-degree of one-inclusion graph for concept class with VCdim d is $\leq d$
- New tight compression scheme for maximum classes
- Intricate combinatorics
- Money: Generalizing to maximal classes
- Acknowledgements: Sally Floyd and Sanjoy Dasgupta for helpful discussions

Restriction, Reduction and the Tail

- Restriction $C - x$ - throw away the x instance from all concepts
- Reduction C^x - contains those concepts that differ only on x
- $Tail_x(C)$ - concepts that don't have an x edge
- $C - x$ is maximum with VCdim d, C^x is maximum with VCdim $d - 1$
- $C - x = C^x \dot{\cup} (Tail_x(C) - x)$

Restriction, Reduction and the Tail

	x_1	x_2	x_3	x_4
c_1	0	0	0	0
c_2	0	0	1	0
c_3	0	0	1	1
c_4	0	1	0	0
c_5	0	1	0	1
c_6	0	1	1	0
c_7	0	1	1	1
c_8	1	0	0	0
c_9	1	0	1	0
c_{10}	1	0	1	1
c_{11}	1	1	0	0

C

x_2	x_3	x_4
0	0	0
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

$C - x_1$

x_2	x_3	x_4
0	0	0
0	1	0
0	1	1
1	0	0

C^{x_1}

Forbidden Patterns and Tail Matching

- A forbidden labeling pattern for a set of instances is the one that is not consistent with any concept in class C
- Every tail concept contains some forbidden pattern of size d for C^x
- Total number of forbidden patterns equal to tail size
- Algorithm idea - assign representatives to the tail that correspond to C^x forbidden sets. Then clashes with C^x are automatically prevented
- A matching problem - match tail concepts to forbidden sets