Learning Acyclic Probabilistic Circuits Using Test Paths

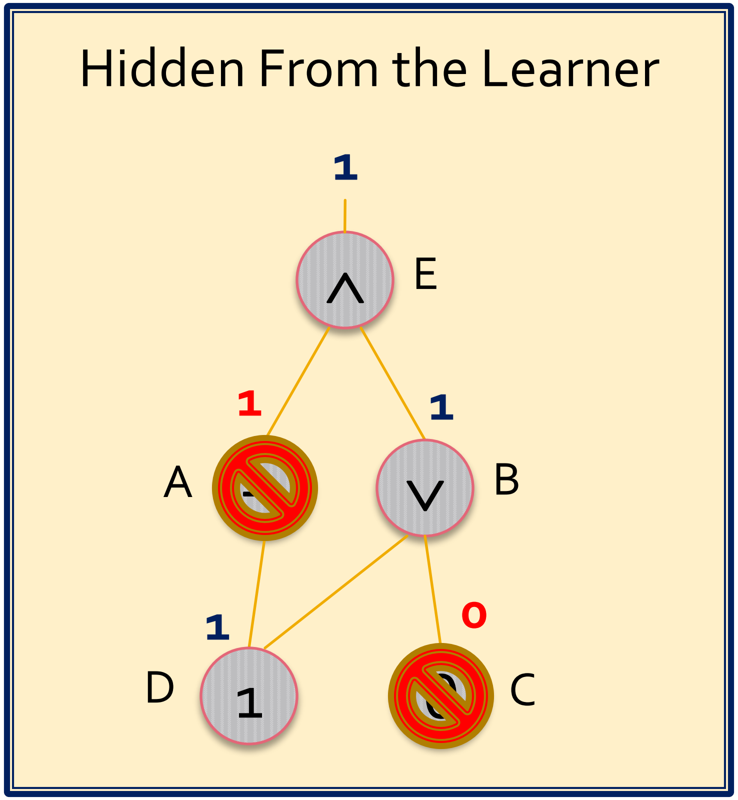
Dana Angluin[1], James Aspnes[1], Jiang Chen[2], David Eisenstat, Lev Reyzin[1]

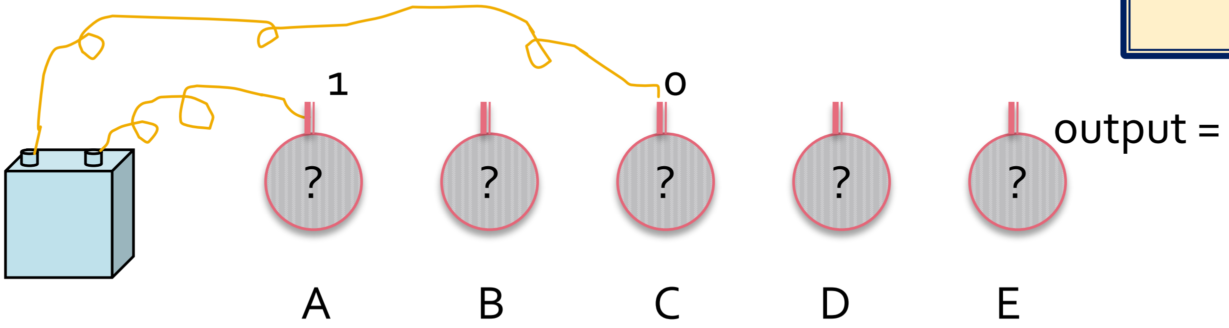
[1]Yale University [2]Yahoo! Inc.

COLT 2008

Value Injection Queries (VIQs)

- Introduced by [AACW ’06]
- Experiments on a hidden Circuit.
  - a gate output may be fixed
  - a gate may be left free
- Query
  - given an experiment, we can observe its output
- Example:





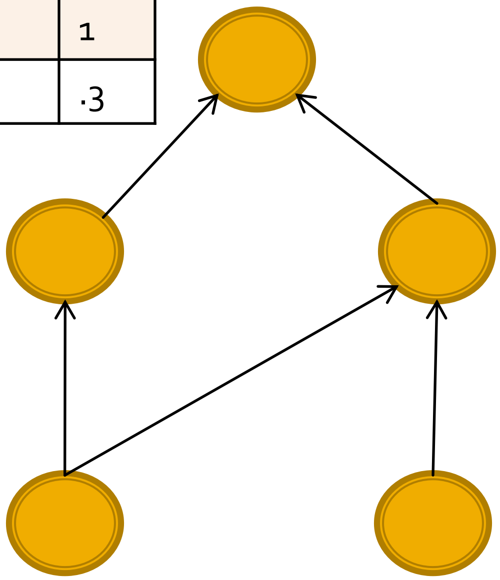
Motivation for the Model

- Model proposed in AACW '06 to study gene disruption and overexpression in gene interaction networks – Boolean circuits.

- Genes have more than two states: AACR '07 studied large alphabet circuits, interested in sensitivity on various parameters

- Real life circuits are often probabilistic – brings us to this paper AACER '08

(Acyclic) Probabilistic Circuits



Input 1	Input 2	P(1)
0	0	.15
0	1	.5
1	0	1
1	1	.3

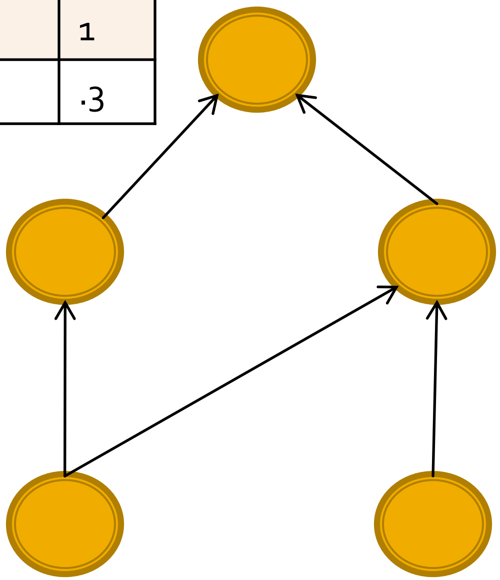
Input 1	Input 2	P(1)
0	0	.7
0	1	1
1	0	.5
1	1	0

Input	P(1)
0	.4
1	.8

P(1)
.9

P(1)
.9

(Acyclic) Probabilistic Circuits



Input 1	Input 2	P(1)
0	0	.15
0	1	.5
1	0	1
1	1	.3

Input 1	Input 2	P(1)
0	0	.7
0	1	1
1	0	.5
1	1	0

Input	P(1)
0	.4
1	.8

P(1)
.9

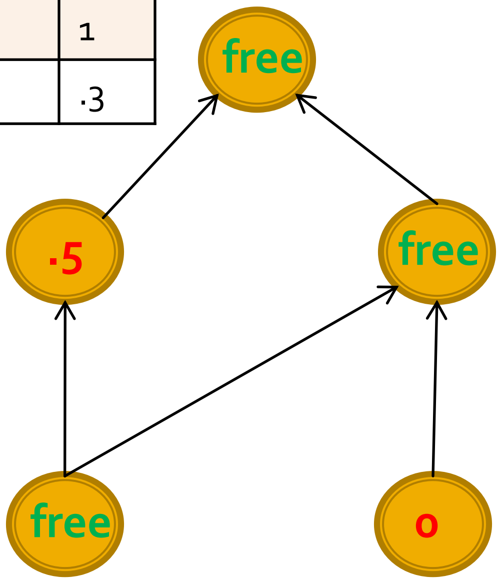
P(1)
.9

VIQs on Probabilistic Circuits

Exact VIQs

(Acyclic) Probabilistic Circuits

Input 1	Input 2	P(1)
0	0	.15
0	1	.5
1	0	1
1	1	.3



Input 1	Input 2	P(1)
0	0	.7
0	1	1
1	0	.5
1	1	0

~~Input~~	~~P(1)~~
~~0~~	~~4~~
~~1~~	~~.8~~

P(1)
.9

~~P(1)~~
~~9~~

VIQs on Probabilistic Circuits

Exact VIQs

The Learning Problems

- **ε-Approximate Learning**
  - **ε-behavioral equivalence:** Circuits C and C’ are ε-behaviorally equivalent if for any experiment s, $d(C(s)-C'(s)) < \varepsilon$.
    - $d(C(s)-C'(s))$ is a notion of statistical distance
  - **The problem:** Given query access to a hidden circuit C^*, find a circuit C ε-behaviorally equivalent to C^* by making value-injection queries.

- **Exact Learning**
  - **behavioral equivalence:** Two circuits C and C’ are behaviorally equivalent if for any experiment s, $C(s)=C'(s)$.
  - **The problem:** Given query access to a hidden circuit C^*, find a circuit C behaviorally equivalent to C^* by making exact value-injection queries.

Previous Work

Circuit	Fan-in	Topology	Gates	VIQ Learnability
Boolean	2	arbitrary	AND/OR	NP-Hard
Boolean	unbounded	constant depth	AND/OR/Θ_2	NP-Hard
Boolean	constant	log depth	arbitrary	Poly-time
Large ∑	constant	log depth	arbitrary	W(1) Hard in shortcut width
Large ∑	constant	bounded shortcut width	arbitrary	Poly-time
Analog	constant	bounded shortcut width	arbitrary	Poly-time approximate

Talk Outline / Main Results

- The Test Path Lemma

- Boolean Probabilistic Circuits
  - Approximately Learnable

- Larger Alphabet Probabilistic Circuits
  - Not Learnable Using Test Paths
  - Learnable with Function Injection Queries

Talk Outline / Main Results

- The Test Path Lemma
- Boolean Probabilistic Circuits
  - Approximately Learnable
- Larger Alphabet Probabilistic Circuits
  - Not Learnable Using Test Paths
  - Learnable with Function Injection Queries

If nothing else, I want to show you how probabilistic circuits behave differently than you might expect

The Test Path Lemma

- A test path for a wire w is a value injection experiment in which the free gates form a directed path in the circuit graph from w to the output wire. All the other wires in the circuit are fixed, including the inputs of w.

- The test path lemma: Let C be a deterministic circuit.

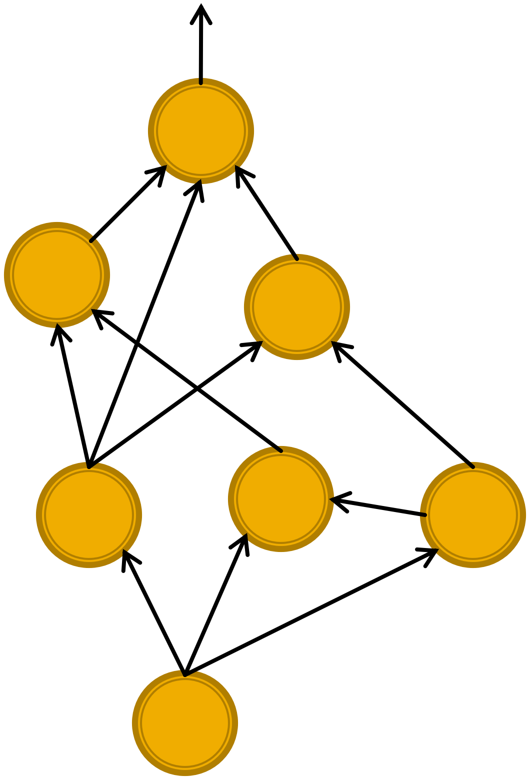
  If for some value injection experiment e, wire w and alphabet symbols σ and τ it is the case that

  $C(p|_{w=\sigma}) = C(p|_{w=\tau})$

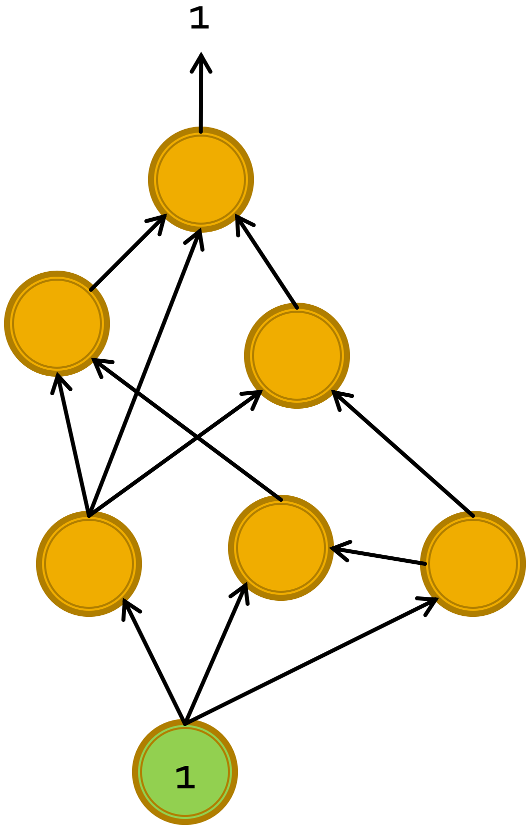
  Then for every test path p < e, then also

  $C(e|_{w=\sigma}) = C(e|_{w=\tau})$.

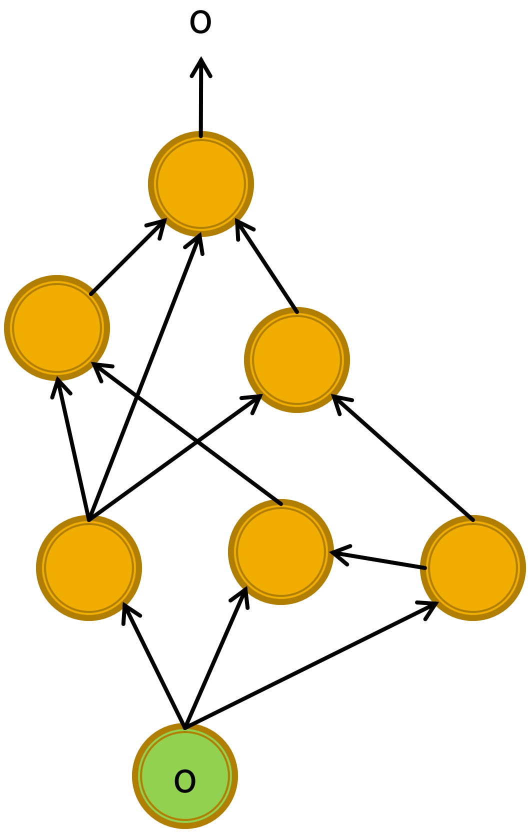
Test Path Lemma Illustrated



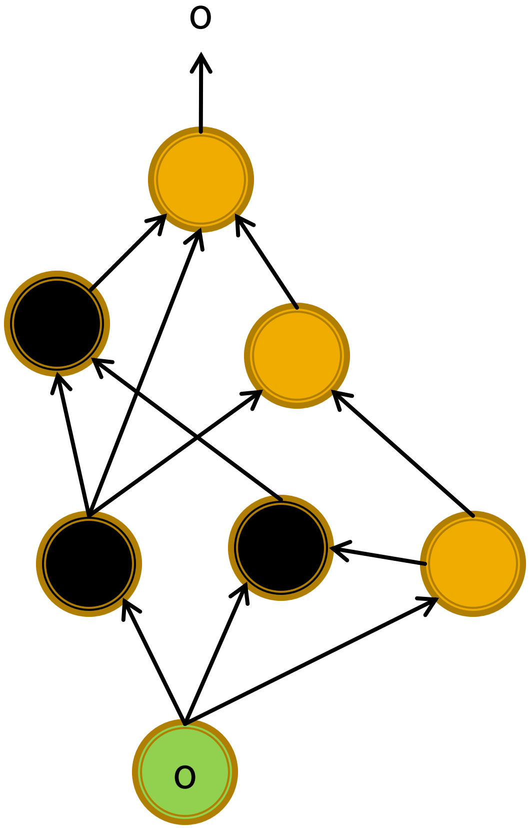
Test Path Lemma Illustrated



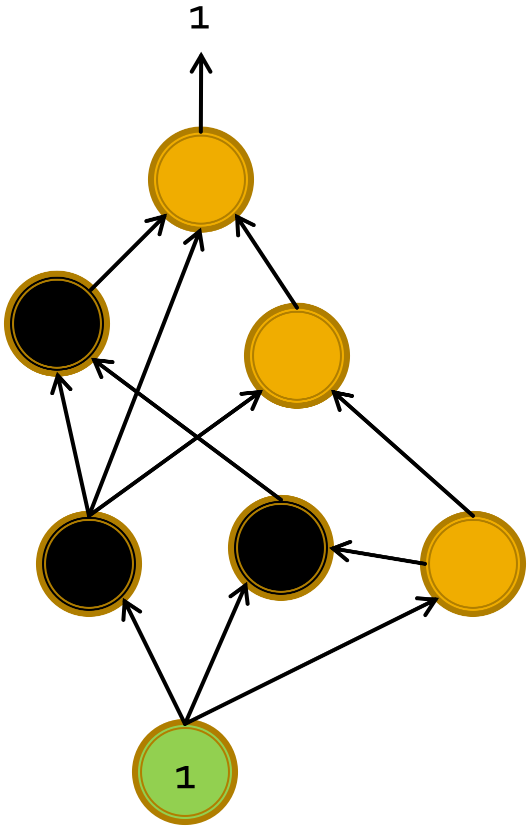
Test Path Lemma Illustrated



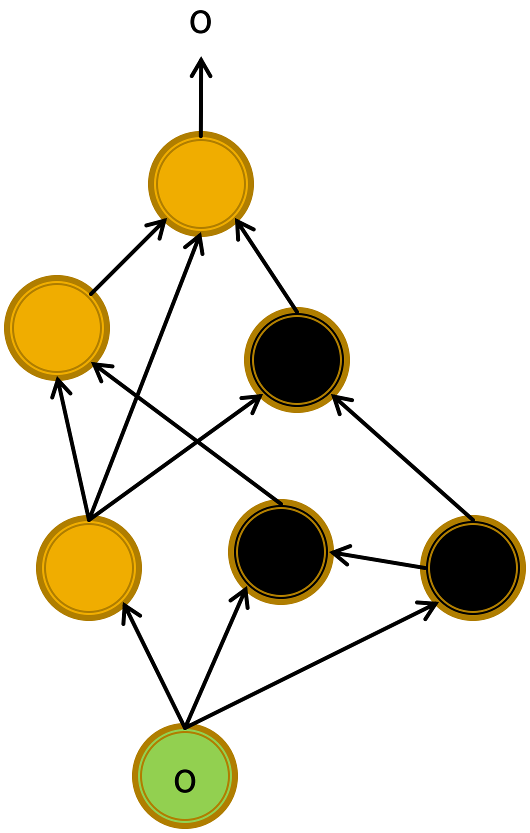
Test Path Lemma Illustrated



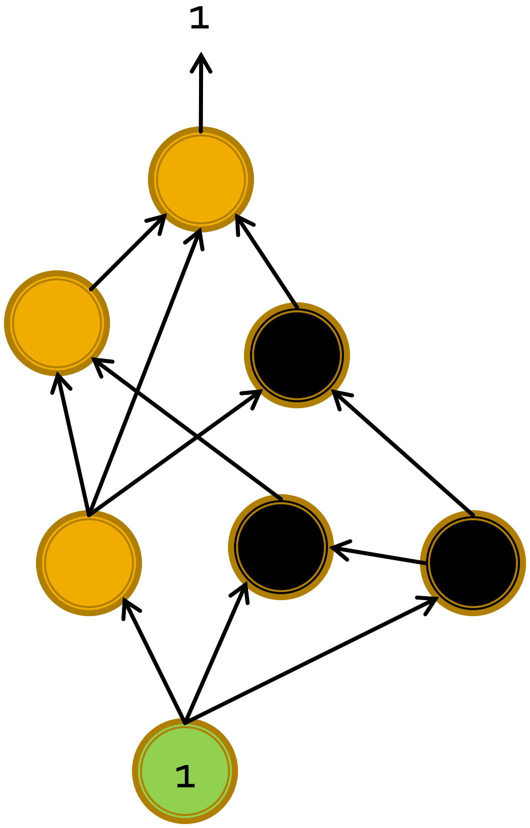
Test Path Lemma Illustrated



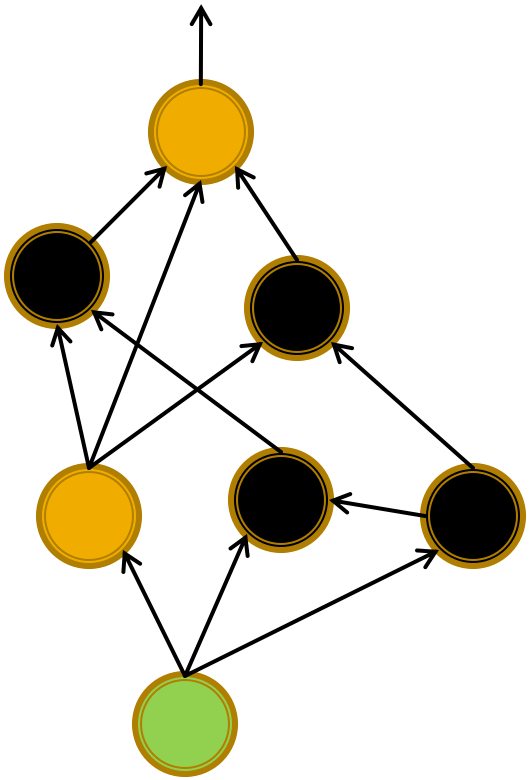
Test Path Lemma Illustrated



Test Path Lemma Illustrated

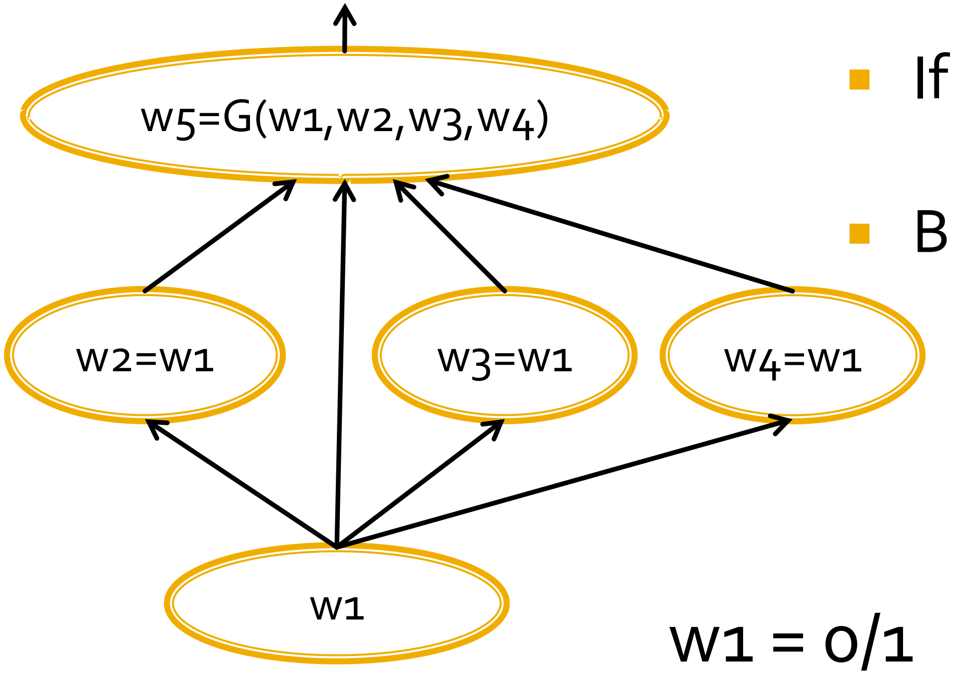


Test Path Lemma Illustrated



Attenuation of Signal in Test Paths

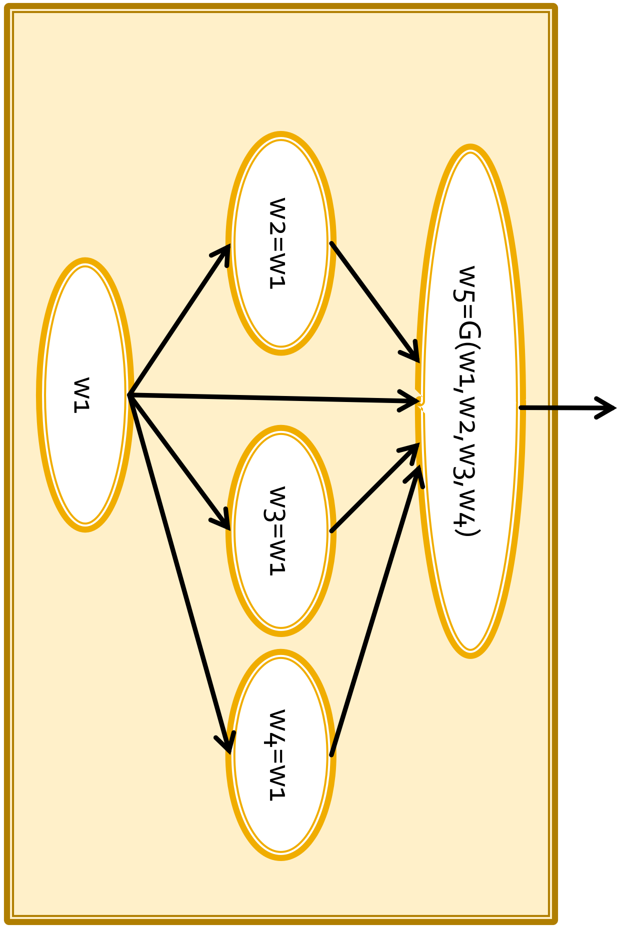
Let $G(w1,w2,w3,w4) = ((1-w1)+2w2+2w3+2w4)/7$



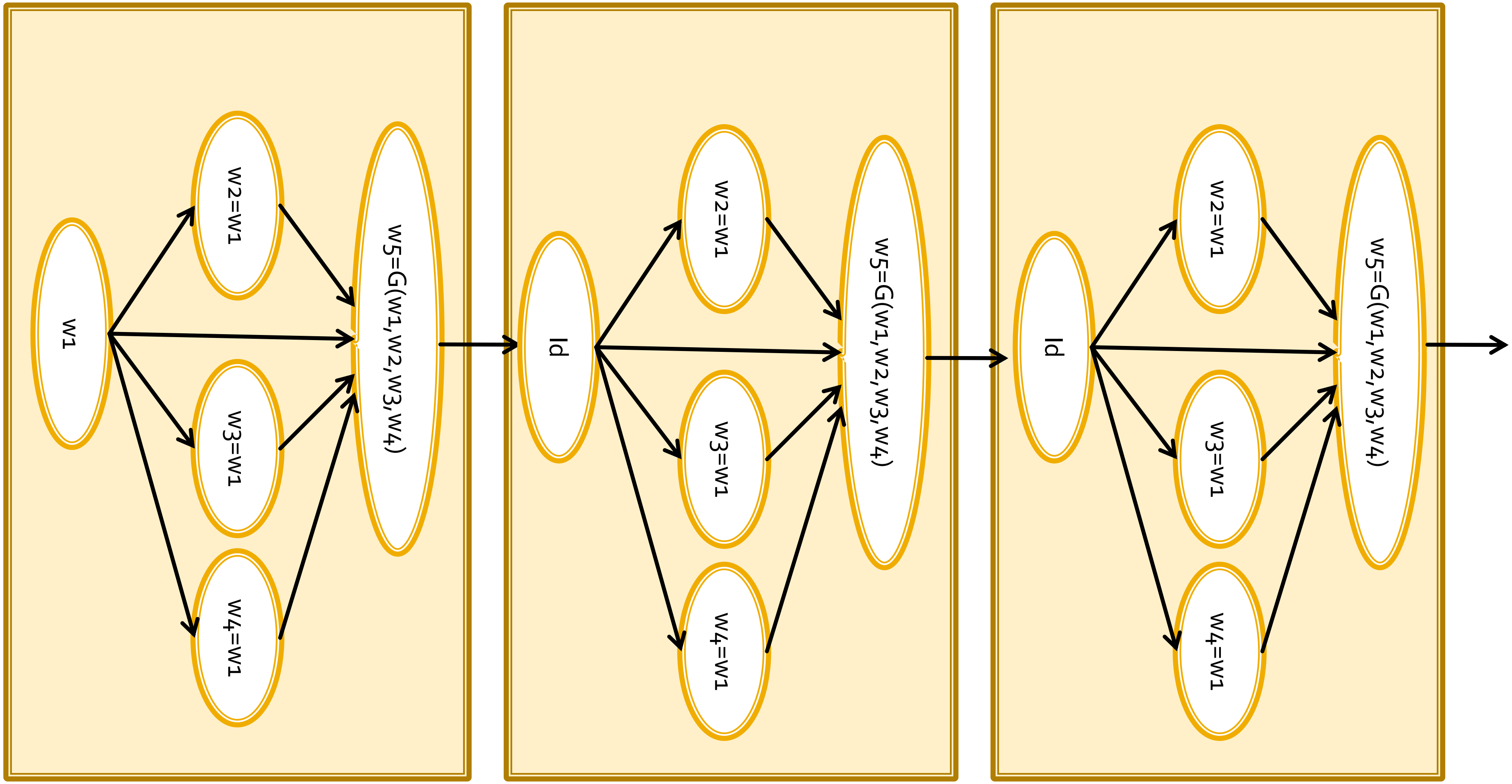
$w1 = 0/1$

- If e sets all wires to be free, then $d(D1(e|_{w=0}),D1(e|_{w=1})) = 5/7$.
- But for any test path p for w1 $d(D1(p|_{w=0}),D1(p|_{w=1})) = 1/7$.

Exponential Attenuation



Exponential Attenuation



Boolean Probabilistic Circuits

- Positive Result for Boolean Probabilistic Circuits:
  - There is a nonadaptive learning algorithm that with probability at least (1 – δ) ε-approximately learns any Boolean probabilistic circuit w/ n wires, constant fan-in and depth c log n using value injection queries in time bounded by a **polynomial in n, 1/ ε and log(1/ δ).**

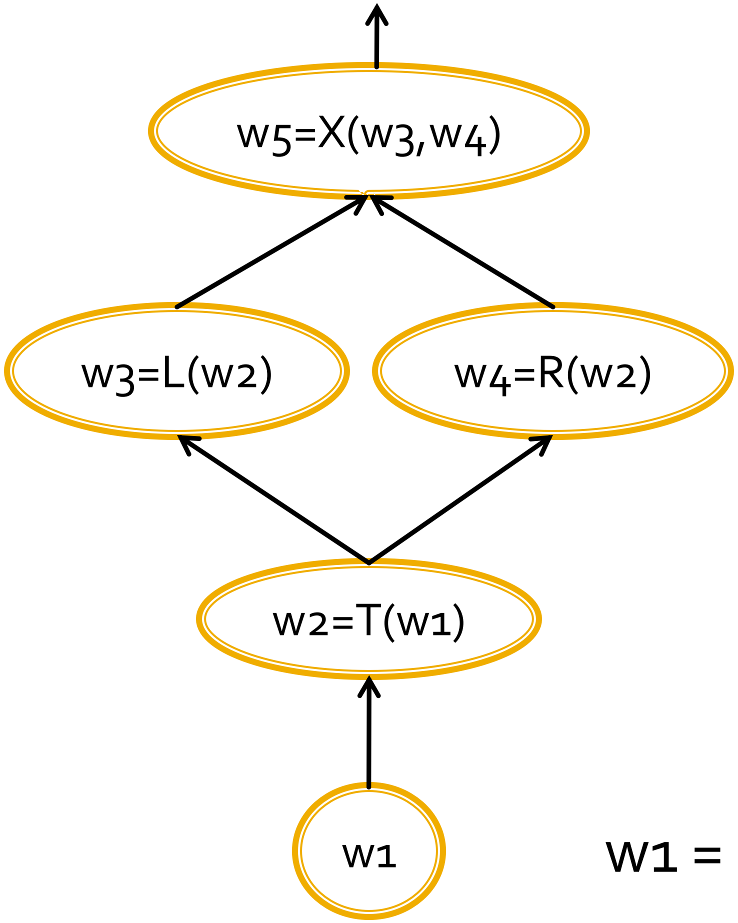
Boolean Probabilistic Circuits

- Positive Result for Boolean Probabilistic Circuits:
  - There is a nonadaptive learning algorithm that with probability at least (1 – δ) ε-approximately learns any Boolean probabilistic circuit w/ n wires, constant fan-in and log n depth using value injection queries in time bounded by a **polynomial in n, 1/ ε and log(1/ δ).**
- Proof idea:
  - The test paths lemma still holds (with some attenuation)
  - The test paths approximately determine the function of the circuit. Collect all possible test paths (**of log depth**) and put them into CircuitBuilder (AACW '06)

Larger Alphabet Probabilistic Circuits

- Lets consider probabilistic circuits that have gates that operate on more than two alphabet symbols.
- What happens to the test path lemma in the large alphabet, probabilistic case?

Test Paths Fail (Completely) for |∑|>2



w5=X(w3,w4)

w3=L(w2)

w4=R(w2)

w2=T(w1)

w1

T(00)=T(11) =	U({00,11})
T(01)=T(10) =	U({01,10})
L(00)=L(01)=	00
L(10)=L(11)=	01
R(00)=R(10)=	00
R(01)=R(11)=	01
X(ab,cd)=	0(b⊗d)

w1 = 00 / 01

Function Injection Queries

- An alphabet transformation is a function f that maps symbols to distribution over symbols.
- A function injection experiment is a mapping that for each wire either leaves it free, assigns it an alphabet symbol, or assigns a transformation f.
- A function injection query (FIQ) takes a function injection experiment and returns the symbol assigned to the output wire.

Learning Large Alphabet Circuits

- A 2-partition experiment is a function injection experiment in which every alphabet transformation is a 2-partition.

- By using 2-partition experiments, we can “smash” the large alphabet circuits back to the Boolean case.
  - We get same positive learnability results for probabilistic large alphabet circuits using FIQs as we have for probabilistic Boolean circuits using VIQs.

Open Problems

- Can we learn large alphabet probabilistic circuits without using test paths? This seems very hard to approach…
- Can “NOT injections” help?
- We can study other variants of this model, for example social networks (AAR ’ 08)

Summary and Discussion

- Test paths are a useful tool for learning circuits!
- Learnability (once again) surprisingly sensitive to small changes in parameters.
- We have an interesting model in which we made progress in learning probabilistic circuits/ Bayesian Networks

Results Table

Circuit	Fan-in	Topology	Gates	VIQ Learnability
Boolean	2	arbitrary	AND/OR	NP-Hard
Boolean	unbounded	constant depth	AND/OR/Θ_2	NP-Hard
Boolean	constant	log depth	arbitrary	Poly-time
Large ∑	constant	log depth	arbitrary	W(1) Hard in shortcut width
Large ∑	constant	bounded shortcut width	arbitrary	Poly-time
Analog	constant	bounded shortcut width	arbitrary	Poly-time approximate
Probabilistic Boolean	constant	log depth	arbitrary	Poly-time approximate
Probabilistic Large ∑	constant	log depth	arbitrary	Poly-time w/ FIQs ??? w/ VIQs
Probabilistic cyclic!	Unbounded	arbitrary	independent cascade	Poly-time w/ exact VIQs