Journal of Machine Learning Research ?? (2009) ???–??? Submitted 1/09; Revised 6/09; Published ?/??

Learning Acyclic Probabilistic Circuits Using Test Paths

Dana Angluin DANA.ANGLUIN@YALE.EDU
James Aspnes JAMES.ASPNES@YALE.EDU
Department of Computer Science
Yale University

Jiang Chen CRIVER@GMAIL.COM
Yahoo! Inc.
701 First Avenue
Sunnyvale, CA 94086

David Eisenstat EISENSTATDAVID@GMAIL.COM

Lev Reyzin LEV.REYZIN@YALE.EDU
Department of Computer Science
Yale University

Editor: Rocco Servedio

Abstract

We define a model of learning probabilistic acyclic circuits using value injection queries, in which fixed values are assigned to an arbitrary subset of the wires and the value on the single output wire is observed. We adapt the approach of using test paths from the Circuit Builder algorithm (Angluin et al., 2009) to show that there is a polynomial time algorithm that uses value injection queries to learn acyclic Boolean probabilistic circuits of constant fan-in and log depth. We establish upper and lower bounds on the attenuation factor for general and transitively reduced Boolean probabilistic circuits of test paths versus general experiments. We give computational evidence that a polynomial time learning algorithm using general value injection experiments may not do much better than one using test paths. For probabilistic circuits with alphabets of size three or greater, we show that the test path lemmas (Angluin et al., 2009, 2008b) fail utterly. To overcome this obstacle, we introduce function injection queries, in which the values on a wire may be mapped to other values rather than just to themselves or constants, and prove a generalized test path lemma for this case.

Keywords: nonadaptive learning algorithms, probabilistic circuits, causal Bayesian networks, value injection queries, test paths

1. Introduction

Probabilistic networks are used as models in a variety of domains, for example, gene interaction networks, social networks and causal reasoning. In a binary model of gene interaction, the state of each gene is either active or inactive, and the state of each gene is determined as a function of the states of some number of other genes, its inputs. In a probabilistic variant of the model, the activation function specifies, for each possible combination of the states of the inputs, the probability that the gene will be active (Friedman et al., 2000). In the independent cascade model of social networks, the state of each agent is active or inactive and for each pair (u, v) of agents, there is a probability that the activation of u will cause v to become active. Kempe, Kleinberg, and Tardos (2003, 2005) study the problem of maximizing influence in this and related models of social networks. In a Bayesian network there is an acyclic directed graph and a joint probability distribution over the node values such that the joint distribution is the product of each of the marginal distributions for each node given the values of the parents (in-neighbors) of the node.

A fundamental question is how much we can infer about the properties and structure of such networks from observing and experimenting with their behaviors. Prior research gives evidence from cryptography that there may be no polynomial time algorithm to learn Boolean functions represented by acyclic circuits of constant fan-in and depth $O(\log n)$ when we can set only the inputs of the circuit and observe only the output (Angluin and Kharitonov, 1995). In this paper we consider a different setting, **value injection queries**, in which we can fix the values on any subset of wires in the target circuit, but still only observe the output of the circuit.

The concept of value injection queries was inspired by models of gene suppression and gene overexpression in the study of gene interaction networks (Akutsu et al., 2003; Ideker et al., 2000) and was introduced by Angluin et al. (2009). In a causal Bayesian network there is an additional action $\mathrm{do}(X = x)$ that forces a node X to take on a value x (Pearl, 2000). A value injection query may also be viewed as a set of such actions, one for each wire fixed to a value.

Angluin et al. (2009) investigate the learnability of deterministic circuits using value injection queries and behavioral equivalence queries. Polynomial time learning algorithms using just value injection queries are given for two classes of acyclic circuits. Circuit Builder uses value injection queries to learn acyclic deterministic circuits with constant-size alphabets, constant fan-in and depth $O(\log n)$ up to behavioral equivalence in polynomial time. Another algorithm is given that learns constant-depth acyclic Boolean circuits with NOT gates and unbounded fan-in AND, OR, NAND and NOR gates up to behavioral equivalence in polynomial time using value injection queries. Negative results include an exponential lower bound on the number of value injection queries to learn acyclic Boolean circuits of unbounded depth and unbounded fan-in, and the **NP**-hardness of learning acyclic Boolean circuits of unbounded depth and constant fan-in using value injection queries.

In extending these results to analog circuits, Angluin et al. (2008b) consider circuits with polynomial-size alphabets. They give evidence of the computational hardness of learning acyclic circuits over a polynomial-size alphabet even if the depth is restricted to $O(\log n)$, motivating structural restrictions on the graphs of the circuits to achieve polynomial time learnability. They give the Distinguishing Paths Algorithm, which uses value injection queries and learns acyclic deterministic circuits that are transitively reduced and have polynomial-size alphabets, constant fan-in and unbounded depth up to behavioral equivalence in polynomial time. They also give a generalization to circuits with a constant bound on shortcut width.

In this paper we seek to extend some of these positive learnability results to the case of acyclic probabilistic circuits. The key technique in the previous work has been the idea of a **test path** for an arbitrary wire w in the circuit. Informally speaking, a test path is a directed path of wires from w to the output wire in which each wire is an input of the next wire on the path, and the other (non-path) inputs of wires on the path are fixed to constant values, thus isolating the wires along the path from the rest of the circuit. Ideally, the choice of constant values is made in such a way as to maximize the effect on the output of the circuit of changing w from one value to another. A test path thus functions as a kind of "microscope" for viewing the effects of assigning different values to the wire w. The primary focus of this paper is to understand the properties of test paths in probabilistic circuits, and the extent to which they can be used to give polynomial time algorithms for learning probabilistic acyclic circuits.

In Section 2 we formally define our model of acyclic probabilistic circuits, value injection queries and distribution injection queries, behavioral equivalence, and the learning problem that we consider. In Section 3 we establish some basic results about probabilistic circuits and value and distribution injection experiments. In Section 4 we review the test path lemma used in previous work to establish the ability of a learner to infer circuit behavior from a small subset of experiments and show that it fails utterly in probabilistic circuits with alphabet size greater than two. However, for Boolean probabilistic circuits, we show that the test path lemma holds with an attenuation factor that depends on the structure of the circuit. (Lemma 10 treats general acyclic circuits and Corollary 11 specializes the bound to transitively reduced circuits.) In Section 5 we apply the test path lemma in the Boolean case to adapt the Circuit Builder algorithm (Angluin et al., 2009) to find using value injection queries, with high probability, in time polynomial in n and $1/\varepsilon$, a circuit that is ε-behaviorally equivalent to a target acyclic Boolean probabilistic circuit of size n with constant fan-in and depth bounded by a constant times $\log n$. In Section 6, we consider lower bounds on the attenuation of paths; Theorem 16 shows that our bound is tight for transitively reduced circuits and Theorem 18 gives a lower bound for the case of general acyclic circuits. In Section 7 we give evidence that polynomial time algorithms using general value injection experiments may not do significantly better than algorithms that use test paths. In Section 8 we introduce a stronger kind of query, a **function injection query**, and show that test paths with function injections overcome the limitations of test paths for circuits with alphabets of size greater than two.

2. Model

We extend the circuit learning model (Angluin et al., 2008b, 2009) to probabilistic gates. An unusual feature of this model is that circuits do not have distinguished inputs—since the learning algorithm seeks to predict the output behavior of value injection experiments that override the values on an arbitrary subset of wires, each wire is a potential input.

2.1 Probabilistic Circuits

A **probabilistic circuit** C of **size** $n \geq 1$ has n **wires**, of which one is the distinguished **output wire**. We call the set of C's wires W, and these wires take values in a finite **alphabet** Σ with $|\Sigma| \geq 2$. If $\Sigma = \{0, 1\}$, then C is **Boolean**. The value on a wire is ordinarily determined by the output of an associated probabilistic gate, whose distribution is a function of the values on other wires.

Formally, a **value distribution** D is a probability distribution over Σ, that is, a map from Σ to the real interval $[0, 1]$ such that $\sum_{\sigma \in \Sigma} D(\sigma) = 1$. The probability of σ is $D(\sigma)$. The **support** of D is the set of values $\sigma \in \Sigma$ such that $D(\sigma) > 0$. When the support of D is a singleton $\{\sigma\}$, we say D is **deterministic**. For a nonempty set of values $S \subseteq \Sigma$, the **uniform distribution** $U(S)$ is the distribution such that $U(S)(\sigma) = [\sigma \in S]/|S|$, that is, has value 0 on $\sigma \notin S$ and $1/|S|$ for $\sigma \in S$.

A k-ary **probabilistic gate function** f maps each k-tuple of values $(\sigma_1, \ldots, \sigma_k) \in \Sigma^k$ to a value distribution. When C is Boolean, we can specify f by a truth table giving the expected value for each Boolean vector of inputs. A probabilistic gate function is **deterministic** if it maps k-tuples to deterministic value distributions only.

A **probabilistic gate** g of **fan-in** k pairs a k-ary probabilistic gate function f with a k-tuple $(w_1, \ldots, w_k) \in W^k$ of **input wires**. The gate g is **deterministic** if its gate function f is deterministic. When $k = 0$, the gate g has no inputs, and we can regard it as specifying a value distribution, or, when C is Boolean, a biased coin flip.

A **probabilistic circuit** C maps wires to probabilistic gates. C is **deterministic** if all of its gates are deterministic. The **fan-in** of C is the maximum fan-in over C's gates. The **circuit graph** of C has a node for each wire in W and a directed edge (u, w) if u is one of the input wires of the gate associated with w. It is important to distinguish between wires in the circuit and edges in the circuit graph. For example, if wire u is an input of wires v and w, then there will be two directed edges, (u, v) and (u, w), in the circuit graph.

Wire w is **reachable** from wire u if there is a directed path from u to w in the circuit graph. A wire is **relevant** if the output wire is reachable from it. The **depth** of a wire w is the number of edges in the longest simple path from w to the output wire in the circuit graph. The **depth** of the circuit is the maximum depth of any relevant wire. The circuit is **acyclic** if the circuit graph contains no directed cycles. The circuit is **transitively reduced** if its circuit graph is transitively reduced, that is, if it contains no edge (u, w) such that there is a directed path of length at least two from u to w. In this paper we assume all circuits are acyclic.

2.2 Experiments

In an experiment some wires are constrained to be particular values or value distributions and the other wires are left free to take on values according to their gate functions and the values of their input wires. The behavior of a circuit consists of its responses to all possible experiments. For probabilistic circuits we consider both value injection experiments and distribution injection experiments.

A **distribution injection experiment** e is a function with domain W that maps each wire w to a special symbol $*$ or to a value distribution. A **value injection experiment** e is a distribution injection experiment for which every value distribution assigned is deterministic—that is, always generates the same symbol. To simplify notation, we think of a value injection experiment as a mapping from W to $(\Sigma \cup \{*\})$. If e is either kind of experiment, we say that e leaves w **free** if $e(w) = *$; otherwise we say that e **constrains** w to $e(w)$. If $e(w)$ is a single symbol, then we say e **fixes** w to $e(w)$.

We define a partial ordering $\leq$ on the set containing $*$ and all value distributions D as follows: $D \leq *$ for every value distribution D, and for two value distributions, $D_1 \leq D_2$ if the support of D_1 is a subset of the support of D_2. This ordering is extended to experiments on the same set of wires W as follows: $e_1 \leq e_2$ if for every $w \in W$, $e_1(w) \leq e_2(w)$. The intuitive meaning of $e_1 \leq e_2$ is that e_1 is at least as constraining as e_2 for every wire.

If e is any experiment, w is a wire, and a is $*$ or an element of Σ or a value distribution, then the experiment $e|_{w=a}$ is defined to be the experiment e' such that $e'(w) = a$ and $e'(u) = e(u)$ for all $u \in W$ such that $u \neq w$. If e is any experiment then a **free path** in e is a path in the circuit graph containing only wires w that are free in e.

2.3 Behavior

Let C be a probabilistic circuit. Then a distribution injection experiment e determines a joint distribution over assignments of elements of Σ to all of the wires of the circuit, as follows. If wire w is constrained then w is randomly and independently assigned a value in Σ drawn according to the value distribution $e(w)$; in the case of a value injection experiment, this just assigns a fixed element of Σ to w. If wire w is free and has probabilistic gate function f, and its inputs $u_1, \ldots, u_k$ have been assigned the values $\sigma_1, \ldots, \sigma_k$, then w is randomly and independently assigned a value from Σ according to the value distribution determined by the gate function on these inputs, that is, according to the value distribution $f(\sigma_1, \ldots, \sigma_k)$.

Constrained gates and gates of fan-in zero give the base cases for the above recursive definition, which assigns an element of Σ to every wire because the circuit is acyclic. Let $C(e, w)$ denote the (marginal) value distribution of the assignments of values to w for the above process. The **output distribution** of the circuit, denoted $C(e)$, is the distribution $C(e, z)$, where z is the output wire of the circuit. The **behavior** of a circuit C is the function that maps value injection experiments e to output distributions $C(e)$.

We note that even when the circuit is Boolean and the only non-deterministic gates are uniform coin flips, the problem of exactly computing $C(e)$ is #**P**-hard because we can arrange for $C(e)$ to be the fraction of assignments satisfying a given Boolean formula.

2.4 Example: C_1

We give an example of a simple Boolean probabilistic circuit, which we also refer to later. The 2-input **averaging gate function** $A(b_1, b_2)$ outputs 1 with probability $(b_1 + b_2)/2$. Thus, if both inputs are 0, the output is deterministically 0, if both inputs are 1, the output is deterministically 1, and if its inputs disagree, the output is an unbiased coin flip, $U(\{0, 1\})$. Another characterization of the averaging gate function A is that it randomly and equiprobably selects one of its inputs and copies it to the output.

We define a circuit C_1 of 4 wires as follows: $w_4 = A(w_2, w_3)$, $w_3 = w_1$, $w_2 = w_1$, and $w_1 = U(\{0, 1\})$. The output wire is w_4. C_1 is depicted in Figure 1.

To illustrate the behavior of this circuit, we consider two value injection experiments. Define the experiment e to leave every wire in C_1 free, that is, $e(w_i) = *$ for $1 \leq i \leq 4$. Given e, we construct one random outcome as follows. The wire w_1 is assigned a value as the result of an unbiased coin flip—say it is assigned 0. Then the values assigned to w_2 and w_3 are determined because they are each the output of an identity gate with w_1 as input: both are 0. Finally, because both its input wires have been assigned values, w_4 can be assigned a value according to $A(0, 0)$, which is deterministically 0. It is easy to see that

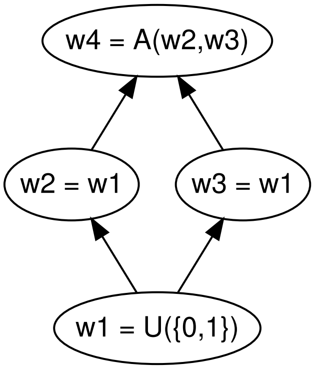


Figure 1: The circuit C_1; w_4 is the output wire.

this is one of two possible outcomes for experiment e; either all wires are assigned 0 or all wires are assigned 1, and these each occur with probability 1/2. The output distribution $C_1(e)$ is just an unbiased coin flip.

Now consider experiment $e' = e|_{w_2=1}$ that fixes w_2 to 1 and leaves the other wires free. Once again, the value of w_1 is determined by a coin flip—say it is assigned 0. Since w_2 is fixed to 1, that is its assignment. Wire w_3 is free, and is therefore assigned the value of w_1, that is 0. Now the inputs of w_4 have been assigned values, so we consider $A(1, 0)$, which randomly and equiprobably selects 0 or 1. If, instead, the coin flip for w_1 had returned 1, all wires would be assigned 1. There are three possible assignments to (w_1, w_2, w_3, w_4) for experiment e': $(1, 1, 1, 1)$ with probability 1/2, $(0, 1, 0, 0)$ with probability 1/4 and $(0, 1, 0, 1)$ with probability 1/4. The output distribution $C_1(e')$ is a biased coin flip that is 1 with probability 3/4.

2.5 Behavioral Equivalence

Two circuits C and C' are **behaviorally equivalent** if they have the same set of wires, the same output wire and the same behavior, that is, for every value injection experiment e, $C(e) = C'(e)$. We also need a concept of approximate equivalence. The **(statistical) distance** between value distributions D and D' is $d(D, D') = (1/2)\sum_{\sigma} |D(\sigma) - D'(\sigma)|$, which takes values in $[0, 1]$. Note that when D and D' are deterministic, $d(D, D')$ is 0 if $D = D'$ and 1 otherwise. For $\varepsilon \geq 0$, C is **ε-behaviorally equivalent** to C' if they contain the same wires and the same output wire, and for every value injection experiment e, $d(C(e), C'(e)) \leq \varepsilon$, where d is the statistical distance between value distributions.

In Lemma 2 we show that the behavioral equivalence of C and C' implies $C(e) = C'(e)$ for all distribution injection experiments as well. However, behavioral equivalence is not sufficient to guarantee that two circuits have the same topology; even when all the gates are Boolean, deterministic and relevant, the circuit graph of the target circuit may not be uniquely determined by its behavior (Angluin et al., 2009).

2.6 Queries

The learning algorithm gets information about the target circuit by specifying a value injection experiment e and observing the element of Σ assigned to the output wire. Such an action is termed a **value injection query**, abbreviated VIQ. A value injection query does not return complete information about the value distribution $C(e)$, but instead returns an element of Σ selected according to the distribution $C(e)$. Thus, in order to approximate the distribution $C(e)$, the learner must repeatedly make value injection queries with experiment e. In this case, the goal of learning is approximate behavioral equivalence.

2.7 The Learning Problem

The learning problem is ε-**approximate learning**: by making value injection queries to a target circuit C drawn from a known class of probabilistic circuits, the goal is to find a circuit C' that is ε-behaviorally equivalent to C. The inputs to the learning algorithm are the names of the wires in C, the name of the output wire and positive numbers ε and δ, where the learning algorithm is required to succeed with probability at least $(1-\delta)$.

We note that acyclic deterministic circuits are a subclass of acyclic probabilistic circuits. If the target circuit C is deterministic and we learn a probabilistic circuit C' that is 1/3-behaviorally equivalent to C, then we can compute the behavior of C on any value-injection experiment e with high probability by sampling the behavior of $C'(e)$. The negative results concerning learning deterministic circuits using value injection queries shown by Angluin et al. (2009) carry over to learning probabilistic circuits. In particular, for $\varepsilon = 1/3$ and $\delta = 1/2$, with no bound on fan-in or depth, the worst-case expected number of value injection queries necessary to learn acyclic probabilistic Boolean circuits is exponential, while with constant fan-in and no bound on depth, no polynomial time algorithm can learn acyclic probabilistic Boolean circuits if **NP** is not equal to **BPP**.

3. Preliminary Results

In this section we establish some basic results about probabilistic circuits, value injection experiments and distribution injection experiments. The reader may choose to skip this section and return to it as needed for proofs in subsequent sections.

We first note that if C is a probabilistic circuit, e is a distribution injection experiment and either $e(w)$ is a value distribution or e deterministically fixes all the input wires of w, then there is a value distribution D such that the value of w in $C(e)$ is determined by a random choice according to D, independent of the values chosen for any other wires. We make systematic use of this observation to reduce the number of experiments under consideration.

We start by considering two circuits C_1 and C_2 over the same wires, and distribution injection experiments e_1 and e_2 that agree on the distribution assigned to a wire w and that show a certain distance between $C_1(e_1)$ and $C_2(e_2)$. The following lemma says that we may modify e_1 and e_2 to fix w to a particular value $\sigma \in \Sigma$ while preserving (or increasing) the distance they show.

Lemma 1 *Let C_1 and C_2 be probabilistic circuits on wires W with the same output wire, let $w \in W$ be a wire, let D be a value distribution, and let e_1 and e_2 be distribution injection* *experiments such that* $e_1(w) = e_2(w) = D$*. Then there exists a value* $\sigma \in \mathrm{support}(D)$ *such that*

$$d(C_1(e_1|_{w=\sigma}), C_2(e_2|_{w=\sigma})) \geq d(C_1(e_1), C_2(e_2)).$$

Proof We have

$$\begin{aligned} d(C_1(e_1), C_2(e_2)) &= \frac{1}{2}\sum_{\tau\in\Sigma}\Big|C_1(e_1)(\tau) - C_2(e_2)(\tau)\Big| \\ &= \frac{1}{2}\sum_{\tau\in\Sigma}\left|\sum_{\rho\in\Sigma} C_1(e_1|_{w=\rho})(\tau)D(\rho) - \sum_{\rho\in\Sigma} C_2(e_2|_{w=\rho})(\tau)D(\rho)\right| \\ &\leq \frac{1}{2}\sum_{\rho\in\Sigma} D(\rho)\sum_{\tau\in\Sigma}\Big|C_1(e_1|_{w=\rho})(\tau) - C_2(e_2|_{w=\rho})(\tau)\Big| \\ &= \sum_{\rho\in\Sigma} D(\rho) d(C(e_1|_{w=\rho}), C(e_2|_{w=\rho})), \end{aligned}$$

by the triangle inequality. Let

$$\sigma = \operatorname*{arg\,max}_{\rho\in\mathrm{support}(D)} d(C(e_1|_{w=\rho}), C(e_2|_{w=\rho})),$$

so that

$$d(C(e_1|_{w=\sigma}), C(e_2|_{w=\sigma})) \geq d(C(e_1), C(e_2))$$

by an averaging argument. ■

By successively replacing each value distribution by a particular value, we may convert a distribution injection experiment that shows a certain distance between two circuits into a value injection experiment that shows at least that distance between the two circuits.

Lemma 2 *Let* C_1 *and* C_2 *be probabilistic circuits on wires* W *with the same output wire and let* e *be a distribution injection experiment. Then there exists a value injection experiment* $e' \leq e$ *such that*

$$d(C_1(e'), C_2(e')) \geq d(C_1(e), C_2(e)).$$

Proof By induction on $|V|$, where $V \subseteq W$ is the set of wires that e constrains to distributions that are not deterministic. If $|V| > 0$, then let $w \in V$. By Lemma 1, there exists a value $\sigma \in \Sigma$ such that

$$d(C_1(e|_{w=\sigma}), C_2(e|_{w=\sigma})) \geq d(C_1(e), C_2(e)).$$

Since $e|_{w=\sigma}$ constrains one fewer wire to a nonconstant distribution, the existence of e' follows from the inductive hypothesis. ■

Thus, value injection experiments suffice to establish approximate behavioral equivalence with respect to distribution injection experiments.

Corollary 3 *If circuits C_1 and C_2 are ε-behaviorally equivalent with respect to value injection experiments, then C_1 and C_2 are ε-behaviorally equivalent with respect to distribution injection experiments.*

Suppose that C is a probabilistic circuit and e_1 and e_2 are distribution injection experiments. For each wire w, we say that e_1 and e_2 **agree** on w if either

- e_1 and e_2 constrain w to the same distribution, or
- w is free in e_1 and e_2, and e_1 and e_2 agree on all of w's inputs.

It is clear that if e_1 and e_2 agree on a wire w, then the marginal distributions of w in e_1 and e_2 are identical, that is, $C(e_1, w) = C(e_2, w)$.

Lemma 4 *Let C be a probabilistic circuit on wires W and let e_1 and e_2 be distribution injection experiments that agree on wires $V \subseteq W$. Then there exist distribution injection experiments $e'_1 \leq e_1$ and $e'_2 \leq e_2$ such that for each wire $w \in V$, there exists a value $\sigma \in \Sigma$ such that $e'_1(w) = e'_2(w) = \sigma$, and*

$$d(C(e'_1), C(e'_2)) \geq d(C(e_1), C(e_2)).$$

Proof By induction on the number of unfixed wires $w \in V$. If there is such a wire, choose v by the acyclicity of the circuit to be one that is not reachable from the others. If $e_1(v) = e_2(v) = *$, then e_1 and e_2 agree on all of v's inputs, and by the choice of v, all of v's inputs are fixed. As such, we may assume without loss of generality that e_1 and e_2 in fact constrain v to the distribution $D = C(e_1, v) = C(e_2, v)$. By Lemma 1, there exists a value $\sigma \in \text{support}(D)$ such that

$$d(C(e_1|_{v=\sigma}), C(e_2|_{v=\sigma})) \geq d(C(e_1), C(e_2)).$$

The existence of e'_1 and e'_2 follows from the inductive hypothesis. ■

The following lemma shows that constraining a wire w does not change the behavior of wires that are not reachable from w.

Lemma 5 *Let C be a probabilistic circuit on wires W, let e be a distribution injection experiment, let $w \in W$ be a wire free in e, and let D be a value distribution. Then e and $e|_{w=D}$ agree on all wires $u \in W$ such that there is no free path from w to u in e.*

Proof If u is constrained, then the conclusion follows. Otherwise, let $u \in W$ be a wire free in e such that there is no free path from w to u in e. Then no input v of u has a free path from w to v in e. We proceed by induction on the length of the longest path to u. If this length is zero, then u does not have any inputs. Otherwise, the inductive hypothesis applies to all of u's inputs, on which e and $e|_{w=D}$ then must agree. It follows that they also agree on u. ■

If we consider the distance between the behavior of a circuit with a wire constrained to two different value distributions, the following lemma allows us to move to a situation in which the wire is constrained to two different value distributions whose supports are disjoint. In the special case of Boolean circuits, the property of disjoint supports means that the resulting value distributions are deterministic. Later we see that this fundamentally distinguishes between alphabet size two and larger alphabets.

Lemma 6 *Let C be a probabilistic circuit on wires W, let $w \in W$ be a wire, and let D_1, D_2 be value distributions. There exist value distributions D_1', D_2' with* $\mathrm{support}(D_1') \cap \mathrm{support}(D_2') = \emptyset$ *such that for all experiments e,*

$$d(C(e|_{w=D_1}), C(e|_{w=D_2})) = d(D_1, D_2) d(C(e|_{w=D_1'}), C(e|_{w=D_2'})).$$

Proof Intuitively, we couple D_1 and D_2 so that $D_1 = D_2$ as often as possible and let $\widehat{D_i}$ be the distribution of D_i given that $D_1 \neq D_2$. It can be shown that $\widehat{D_1}$ and $\widehat{D_2}$ have disjoint support. Formally, we have

$$\begin{aligned} d(C(e|_{w=D_1}), C(e|_{w=D_2})) &= \frac{1}{2} \sum_{\sigma \in \Sigma} \left| C(e|_{w=D_1})(\sigma) - C(e|_{w=D_2})(\sigma) \right| \\ &= \frac{1}{2} \sum_{\sigma \in \Sigma} \left| \sum_{\tau \in \Sigma} C(e|_{w=\tau})(\sigma)(D_1(\tau) - D_2(\tau)) \right|. \end{aligned}$$

If we let

$$\begin{aligned} \widehat{D_1}(\tau) &= D_1(\tau) - \min(D_1(\tau), D_2(\tau)) \\ \widehat{D_2}(\tau) &= D_2(\tau) - \min(D_1(\tau), D_2(\tau)), \end{aligned}$$

then

$$d(C(e|_{w=D_1}), C(e|_{w=D_2})) = \frac{1}{2} \sum_{\sigma \in \Sigma} \left| \sum_{\tau \in \Sigma} C(e|_{w=\tau})(\sigma)(\widehat{D_1}(\tau) - \widehat{D_2}(\tau)) \right|.$$

Since $\sum_{\tau \in \Sigma} \widehat{D_1}(\tau) = 1 - \sum_{\tau \in \Sigma} \min(D_1(\tau), D_2(\tau))$ and likewise for D_2,

$$\begin{aligned} d(D_1, D_2) &= \frac{1}{2} \sum_{\tau \in \Sigma} \left| D_1(\tau) - D_2(\tau) \right| \\ &= \frac{1}{2} \sum_{\tau \in \Sigma} \left| \widehat{D_1}(\tau) - \widehat{D_2}(\tau) \right| \\ &= \sum_{\tau \in \Sigma} \widehat{D_1}(\tau) = \sum_{\tau \in \Sigma} \widehat{D_2}(\tau). \end{aligned}$$

If $d(D_1, D_2) > 0$, then the distributions D_1' and D_2' where

$$\begin{aligned} D_1'(\tau) &= \widehat{D_1}(\tau) / d(D_1, D_2) \\ D_2'(\tau) &= \widehat{D_2}(\tau) / d(D_1, D_2) \end{aligned}$$

satisfy the requisite properties. Otherwise, any two distributions with disjoint support will do. ■

4. Test Paths

The concept of a test path has been central in previous work on learning deterministic circuits by means of value injection queries (Angluin et al., 2008b, 2009). A **test path** for a wire w, or w**-test path**, 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; this includes the inputs of w. A **side wire** with respect to a test path p is a wire fixed by p that is input to a free wire in p.

As an example, suppose that $\Sigma = \{0, 1\}$ and the target circuit has a circuit graph as shown in Figure 2. There are four directed paths from w_1 to the output wire: w_1w_5, $w_1w_3w_5$, $w_1w_2w_4w_5$ and $w_1w_3w_4w_5$. A w_1-test path is a value injection experiment that sets the wires of one of these paths to $*$ and the other wires to 0 or 1, for example, $*011*$ or $**0**$. For the test path $*011*$, the side wires are w_3 and w_4, while for the test path $**0**$ the side wire is w_3. The value injection experiments $*****$ and $*01**$ are not test paths.

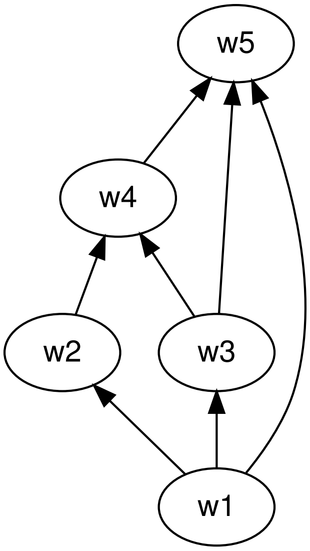


Figure 2: A circuit graph; w_5 is the output wire.

A test path may help the learning algorithm determine the effects of assigning different values to the wire w. The test path lemmas (Angluin et al., 2008b, 2009) may be re-stated as follows.

Lemma 7 *Let C be a deterministic circuit. If for some value injection experiment e, wire w free in e and alphabet symbols σ and τ it is the case that*

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

for every test path $p \leq e$ then also

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

Nontrivial complications arise in attempting to carry over this test path lemma to general probabilistic circuits, as we now show. The following lemma shows that for alphabets of size at least three, there are transitively reduced probabilistic circuits for which the test-path lemma fails completely.

Lemma 8 *If $|\Sigma| \geq 3$, there exists a probabilistic circuit C, value injection experiment e, wire w free in e and alphabet symbols σ and τ such that although for every test path $p \leq e$ for w, $d(C(p|_{w=\sigma}), C(p|_{w=\tau})) = 0$, it is nevertheless the case that $d(C(e|_{w=\sigma}), C(e|_{w=\tau})) = 1/2$.*

Proof Assume that $\Sigma = \{0, 1, 2\}$, and define probabilistic gate functions T and X as follows.

$$\begin{aligned}
T(0) = T(1) &= U(\{0,1\}) \\
T(2) &= 2 \\
X(b_1, b_2) &= b_1 \oplus b_2 \text{ if } b_1, b_2 \in \{0,1\} \\
X(b_1, b_2) &= U(\{0,1\}) \text{ if } b_1 = 2 \text{ or } b_2 = 2,
\end{aligned}$$

where $\oplus$ is sum modulo 2. The gate function T flips a coin on input 0 or 1, and passes 2 through unaltered. The gate function X is exclusive or if neither input is 2, and a coin flip otherwise.

The circuit C has 5 wires, connected as in Figure 3. The output wire is w_5; note that C is transitively reduced.

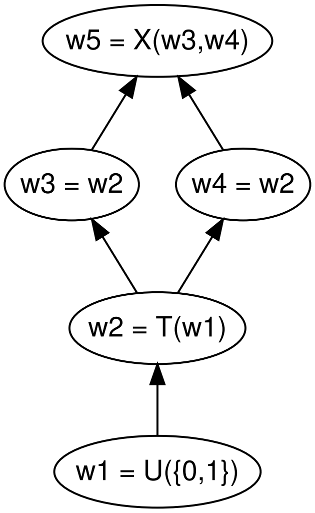


Figure 3: The circuit C; w_5 is the output wire.

Consider the experiment e that leaves all the wires free. In this experiment, we have $C(e|_{w_1=0}) = C(e|_{w_1=1}) = 0$ because w_2 is a coin flip and w_5 is the exclusive or of two copies of the coin flip. On the other hand, $C(e|_{w_1=2}) = U(\{0,1\})$ because $w_4 = w_3 = w_2 = 2$ and w_5 is therefore a coin flip. Thus $d(C(e|_{w_1=0}), C(e|_{w_1=2})) = 1/2$.

However, the only test paths for w_1 fix w_3 and leave all other wires free, or fix w_4 and leave all other wires free, and the two cases are symmetric. If w_3 is fixed to any value and all other wires are free, then w_5 is a coin flip when $w_1 = 2$. If w_3 is fixed to 2 and all other wires are free, then w_5 is also a coin flip. If w_3 is fixed to $b \in \{0,1\}$ and all other wires are free, then when $w_1 \in \{0,1\}$, w_2 is a coin flip, and w_5 is the exclusive or of b and that coin flip, that is, w_5 is also coin flip. Hence, for any test path $p \leq e$ for w_1, we have $C(p|_{w_1=0}) = C(p|_{w_1=2}) = U(\{0,1\})$ and $d(C(p_{w_1=0}), C(p_{w_1=2})) = 0$.

For alphabets Σ of size larger than 3, we can treat three of the symbols as 0, 1 and 2 in the above construction, and the other symbols as "tilt," where each function outputs a tilt value if any of its inputs is a tilt value. ■

4.1 A Bound for Boolean Probabilistic Circuits

Surprisingly, the case of $|\Sigma| = 2$ is different; for Boolean probabilistic circuits there is a useful quantitative relationship between the difference exposed by an arbitrary experiment e and the differences exposed by test paths $p \leq e$. The bound we give depends on the structure of directed paths on free wires in e.

Let e be an experiment and w a wire. Define $\Pi(e,w)$ to be the set of all directed paths from w to the output wire on free wires in e. Let $S(e)$ be the set of wires that originate a free shortcut, that is, the set of free wires w such that there exists a path $p \in \Pi(e,w)$ with two free wires to which w is an input. Define

$$\kappa(e,w) = \sum_{p \in \Pi(e,w)} 2^{|p \cap S(e)|}.$$

Thus, $\kappa(e,w)$ is the sum over paths in $\Pi(e,w)$ of 2 raised to the number of wires on the path that originate free shortcuts in e. If there are no wires that originate free shortcuts in e, then this is just the number of free paths in e. As an example, if the target circuit has the circuit graph shown in Figure 2 and the experiment e leaves all wires free then $\Pi(e,w_1)$ contains the four paths w_1w_5, $w_1w_3w_5$, $w_1w_2w_4w_5$ and $w_1w_3w_4w_5$, $S(e) = \{w_1, w_3\}$, and $\kappa(e,w)$ is $2+4+2+4 = 12$.

The following technical lemma gives a useful recurrence for $\kappa(e,w)$.

Lemma 9 *Let C be a probabilistic circuit, e be a distribution injection experiment, w and u be free wires where w is an input to u, and D_0 be a value distribution. Let $\beta = 2$ if $w \in S(e)$ and $\beta = 1$ otherwise. Then*

$$\kappa(e,w) = \kappa(e|_{u=D_0}, w) + \kappa(e|_{w=1}, u) \cdot \beta.$$

Proof The first term of the sum counts paths that don't contain u, and the second counts paths that do. Let $e' = e|_{u=D_0}$ and $e'' = e|_{w=1}$. We have

$$\begin{aligned}
\kappa(e,w) &= \sum_{p \in \Pi(e,w)} 2^{|p \cap S(e)|} \\
&= \sum_{\substack{p \in \Pi(e,w) \\ u \notin p}} 2^{|p \cap S(e)|} + \sum_{\substack{p \in \Pi(e,w) \\ u \in p}} 2^{|p \cap S(e)|} \\
&= \sum_{p \in \Pi(e',w)} 2^{|p \cap S(e')|} + \sum_{p \in \Pi(e'',u)} 2^{|p \cap S(e'')|}\beta \\
&= \kappa(e',w) + \kappa(e'',u) \cdot \beta,
\end{aligned}$$

since each path $p \ni u$ from w corresponds to the path $p \setminus \{w\}$ from u. ■

Next is the key lemma relating the difference exposed by e to the differences exposed by paths $p \leq e$ for Boolean probabilistic circuits.

Lemma 10 *Let C be a Boolean probabilistic circuit, e be a distribution injection experiment, w be a wire free in e and D_1, D_2 be value distributions. If there exists $\varepsilon \geq 0$ such that for all w-test paths $p \leq e$,*

$$d(C(p|_{w=D_1}), C(p|_{w=D_2})) \leq \varepsilon,$$

then

$$d(C(e|_{w=D_1}), C(e|_{w=D_2})) \leq \kappa(e, w) \cdot \varepsilon.$$

Proof By induction on $\phi(e)$, the number of free wires in e. By Lemma 6, assume that $\text{support}(D_1) \cap \text{support}(D_2) = \emptyset$. The critical feature of the Boolean case is that it follows that $D_1 = 0$ and $D_2 = 1$ without loss of generality—it is important to the following proof that D_1 and D_2 be deterministic.

If $\phi(e) = 1$, then either

$$d(C(e|_{w=0}), C(e|_{w=1})) = 0,$$

or w is the output, e is a w-test path, and $\kappa(e, w) = 1$. Otherwise, the inductive hypothesis is that the lemma holds for all experiments e' with $\phi(e') < \phi(e)$.

Except for w, the experiments $e|_{w=0}$ and $e|_{w=1}$ agree on all constrained wires, so by Lemmas 4 and 5, assume without loss of generality that every wire with no free path from w is in fact fixed. Since C is acyclic, there exists a free wire $u \neq w$ whose only unfixed input is w. Let g be the gate assigned by C to u and let $B_0 = g(e|_{w=0})$ and $B_1 = g(e|_{w=1})$, so that

$$\begin{aligned} C(e|_{w=0}) &= C(e|_{w=0,u=B_0}) \\ C(e|_{w=1}) &= C(e|_{w=1,u=B_1}). \end{aligned}$$

By the triangle inequality,

$$\begin{aligned} d(C(e|_{w=0}), C(e|_{w=1})) \leq\ & d(C(e|_{w=0,u=B_0}), C(e|_{w=1,u=B_0})) \\ & + d(C(e|_{w=1,u=B_0}), C(e|_{w=1,u=B_1})). \end{aligned}$$

Letting $e' = e|_{u=B_0}$, any test path $p \leq e'$ also satisfies $p \leq e$ since $e' \leq e$. The experiment e' has one fewer free wire, as u is free in e, so using the inductive hypothesis, we can bound the first term of the sum by $\kappa(e', w) \cdot \varepsilon$. We now derive a bound on u-test paths so that the inductive hypothesis applies to the second term as well. Let $\beta = 2$ if $w \in S(e)$ and $\beta = 1$ otherwise. Let $e'' = e|_{w=1}$ and suppose $p \leq e''$ is a u-test path. Then

$$\begin{aligned} & d(C(p|_{u=B_0}), C(p|_{u=B_1})) \\ & \leq d(C(p|_{w=1,u=B_0}), C(p|_{w=0,u=B_0})) + d(C(p|_{w=0,u=B_0}), C(p|_{w=1,u=B_1})) \\ & \text{[by the triangle inequality]} \\ & = d(C(p|_{w=1,u=B_0}), C(p|_{w=0,u=B_0})) + d(C(p|_{w=0,u=*}), C(p|_{w=1,u=*})) \\ & \text{[by the definitions of } B_0 \text{ and } B_1\text{]}. \end{aligned}$$

Since w is an input to u, both $p|_{w=*,u=B_0}$ and $p|_{w=*,u=*}$ are w-test paths. Therefore, both terms of the sum are bounded by ε, and the first is nonzero only if w is an input to some free wire in p other than u. It follows that

$$d(C(p|_{u=B_0}), C(p|_{u=B_1})) \leq \beta\varepsilon,$$

and thus that

$$d(C(e''|_{u=0}), C(e''|_{u=1})) \leq \kappa(e'', u) \cdot \beta\varepsilon,$$

so by Lemma 9,

$$\begin{aligned} d(C(e|_{w=0}), C(e|_{w=1})) &\leq \kappa(e', w) \cdot \varepsilon + \kappa(e'', u) \cdot \beta\varepsilon \\ &= \kappa(e, w) \cdot \varepsilon. \end{aligned}$$

■

In the case of transitively reduced circuits, $S(e) = \emptyset$, and $\kappa(e, w) = \pi(e, w)$, where $\pi(e, w) = |\Pi(e, w)|$, the number of directed paths on free wires in e from w to the output wire.

Corollary 11 *Let C be a transitively reduced Boolean probabilistic circuit, e be a distribution injection experiment, and w be a wire free in e. If there exists $\varepsilon \geq 0$ such that for all w-test paths $p \leq e$,*

$$d(C(p|_{w=0}), C(p|_{w=1})) \leq \varepsilon,$$

then

$$d(C(e|_{w=0}), C(e|_{w=1})) \leq \pi(e, w) \cdot \varepsilon.$$

5. Learning Boolean Probabilistic Circuits

The amount of attenuation given by Lemma 10 allows us to adapt the Circuit Builder algorithm (Angluin et al., 2009) to learn Boolean probabilistic circuits with constant fan-in and log depth in polynomial time. For this class of circuits, the attenuation factor $\kappa(e, w)$ is bounded by a polynomial in n.

Theorem 12 *Given constants c and k there is a nonadaptive learning algorithm that with probability at least $(1-\delta)$ successfully ε-approximately learns any Boolean probabilistic circuit with n wires, gates of fan-in at most k and depth at most $c \log n$ using value injection queries in time bounded by a polynomial in n, $1/\varepsilon$ and $\log(1/\delta)$.*

The rest of the section is devoted to proving this theorem. Let the target circuit be C with $\Sigma = \{0, 1\}$ and let positive constants δ, ε, k and c be given such that the fan-in of C is bounded by k and the depth of C is bounded by $c \log n$. For such a circuit, $\pi(e, w)$ is bounded above by $k^{c \log n}$, so the quantity $\kappa(e, w)$ is bounded above by

$$\kappa(n) = k^{c \log n} \cdot 2^{c \log n} = n^{c(\log k + 1)} = n^{O(1)}.$$

We now describe our Probabilistic Circuit Builder algorithm (PCB). PCB is nonadaptive: first it computes a set U of value injection experiments such that every test path is equivalent to some experiment in U. It then repeats each value injection query $e \in U$ enough times that with probability at least $(1-\delta)$, the distribution $C(e)$ is estimated with sufficient accuracy for every $e \in U$. Finally, it uses these estimates to build a circuit C' by repeatedly adding a sufficiently accurate gate all of whose inputs are in the partially constructed circuit. If the estimates of $C(e)$ are all sufficiently accurate, then C' is ε-behaviorally equivalent to C.

5.1 Constructing U

In choosing the experiments U, the goal is that for every potential test path, U includes an equivalent experiment. The structure of the circuit, however, is not known *a priori*, a difficulty that we overcome by the same method as used by Angluin et al. (2009). Let U_* be a universal set of value injection experiments such that for every set of $kc \log n$ wires and every assignment of symbols from $\Sigma \cup \{*\}$ to those wires, some experiment $e \in U_*$ agrees with the values assigned to those wires. There is a deterministic construction of such a set U_* of size

$$2^{O(kc \log n)} \log n = n^{O(kc)}$$

in time polynomial in its size (Angluin et al., 2009). (For intuition, a set of $n^{O(kc)}$ independent random uniform assignments of $*$, 0 and 1 to the wires has this property with high probability.) For every wire w and test path p for w, there is an experiment in U_* that leaves the path wires of p free and fixes the side wires of p to their values in p. Consequently, p and this experiment agree on the output wire. In order to have experiments in which each free wire is also set to 0 and 1, for $b = 0, 1$ let U_b contain every experiment $e|_{w=b}$ such that $e \in U_*$ and w is free in e. The final set of experiments is $U = U_* \cup U_0 \cup U_1$.

5.2 Estimating $C(e)$ for $e \in U$

For each $e \in U$, PCB repeatedly makes a value injection query with e to estimate the value distribution $C(e)$; let $\widehat{C}(e)$ denote this estimate. By Hoeffding's bound, we have that

$$m = O((n\kappa(n)/\varepsilon)^2 \log(|U|/\delta))$$

trials per experiment e suffice to guarantee that with probability at least $1-\delta$, for all $e \in U$,

$$d(C(e), \widehat{C}(e)) \leq \varepsilon/(4n\kappa(n)). \tag{1}$$

Let $e \in U_*$ be a value injection experiment, w be a wire that e leaves free, and D be a value distribution. We define

$$\widehat{C}(e|_{w=D}) = \sum_{\sigma \in \Sigma} D(\sigma)\widehat{C}(e|_{w=\sigma}).$$

Note that this is computed from the values of $\widehat{C}(e|_{w=\sigma})$ and does not require new experiments.

If (1) holds for all $e \in U$, then we have

$$\begin{aligned} d(C(e|_{w=D}), \widehat{C}(e|_{w=D})) &\leq \sum_{\sigma \in \Sigma} D(\sigma) d(C(e|_{w=\sigma}), \widehat{C}(e|_{w=\sigma})) \\ &\leq \varepsilon/(4n\kappa(n)). \end{aligned} \tag{2}$$

5.3 Building the circuit C'

PCB builds the circuit C' one gate at a time. Let W' denote the set of wires of C' that have already been assigned a gate by PCB; initially W' is empty. While $W' \neq W$, PCB attempts to add another gate to C' by searching for a wire $w \in (W - W')$ and a probabilistic gate g' all of whose inputs are in W' such that for each experiment $e \in U_*$ that leaves w free and fixes all inputs of g',

$$d(\widehat{C}(e), \widehat{C}(e|_{w=g'(e)})) \leq 2\varepsilon/(4n\kappa(n)).$$

If no such gate can be found or $W' = W$, PCB outputs C' and halts. We will later show that a gate can be found as long as $W \neq W'$.

The search for g' iterates over every wire $w \in (W - W')$ and every choice of an r-tuple of distinct wires $w_1, \ldots, w_r$ from W' as the inputs of w, where $0 \leq r \leq k$. For each such choice, PCB attempts to define a probabilistic gate function f as follows. For each $(\sigma_1, \ldots, \sigma_r) \in \Sigma^r$, PCB seeks a number $x \in [0, 1]$ such that if D_x is the distribution that is 1 with probability x and 0 with probability $(1 - x)$ then

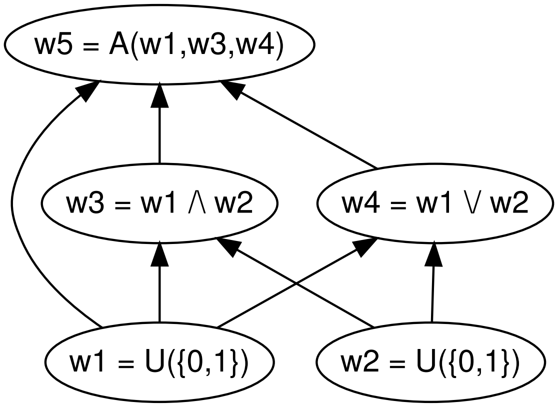
$$d(\widehat{C}(e), \widehat{C}(e|_{w=D_x})) \leq 2\varepsilon/(4n\kappa(n))$$

for all experiments $e \in U_*$ that leave w free and fix w_i to σ_i for $i = 1, \ldots, r$. Since the left hand side is a convex function of x, every such e constrains the possible values of x to an interval, and any x in the intersection of $[0, 1]$ and the intervals for all such e suffices. If the intersection is empty, then the attempt to define f fails; otherwise, $f(\sigma_1, \ldots, \sigma_r)$ is defined to be D_x. If PCB succeeds in defining f for all possible r-tuples $(\sigma_1, \ldots, \sigma_r)$, then the gate g' with inputs $w_1, \ldots, w_r$ and probabilistic gate function f is assigned to w.

5.4 An Illustration

For some intuition about the operation of PCB, consider the probabilistic Boolean circuit shown in Figure 4. Wires w_1 and w_2 are determined by random coin flips, w_3 is the AND of w_1 and w_2, w_4 is the OR of w_1 and w_2, and w_5 is determined by the 3-input averaging gate applied to w_1, w_3 and w_4. The table shows the probability that $w_5 = 1$ for a selected set of value injection experiments.

Suppose that these experiments are contained in U when PCB attempts to add the first gate to C'. Of course, PCB will only have repeated sampling estimates of these probabilities, but suppose for a moment that the exact values were available. Because W' is empty, the first gate added must have no inputs and must be determined by a coin flip that is 1 with some probability x. In this group of experiments, there are two constraints for wire w_1 for the possible values of x. Experiments 1, 2 and 3 give the constraint $(1/6)(1-x)+(5/6)x = 1/2$, which implies $x = 1/2$, and experiments 6, 7 and 8 give the constraint $0(1-x)+(2/3)x = 1/3$,



#	Experiment	$\Pr[w_5 = 1]$
1.	$* * * * *$	1/2
2.	$0 * * * *$	1/6
3.	$1 * * * *$	5/6
4.	$* * 0 * *$	5/12
5.	$* * 1 * *$	3/4
6.	$* 1 * 0 *$	1/3
7.	$0 1 * 0 *$	0
8.	$1 1 * 0 *$	2/3
9.	$* 1 0 0 *$	1/6
10.	$* 1 1 0 *$	1/2

Figure 4: A Boolean circuit with output wire w_5, and some of its behavior.

which also implies $x = 1/2$, consistent with the gate computing w_1 in the target circuit. There are also two constraints on the possible values of x for the wire w_3. Experiments 1, 4 and 5 give the constraint $(5/12)(1 - x) + (3/4)x = 1/2$, which implies $x = 1/4$, and experiments 6, 9 and 10 give the constraint $(1/6)(1 - x) + (1/2)x = 1/3$, which implies $x = 1/2$. Thus there is no consistent value of x that would allow the first gate to be chosen for wire w_3. Rather than exact values, PCB considers intervals determined by error tolerances, but when these are small enough, the constraint intervals for w_3 will not overlap, and PCB will not choose the first gate for wire w_3.

5.5 Correctness

With probability at least $(1 - \delta)$, the estimates $\widehat{C}(e)$ satisfy (1) for all $e \in U$. We now assume that the estimates satisfy these bounds and show that PCB successfully builds a circuit C' that is ε-behaviorally equivalent to C.

We first establish two lemmas connecting gates, paths and experiments. Given a Boolean probabilistic circuit C and a probabilistic gate g, g is **η-correct** for wire w with respect to C if for every value injection experiment e that fixes the input wires for g we have $d(C(e), C(e|_{w=g(e)})) \leq \eta$, where $g(e)$ denotes the value distribution determined by g when its inputs are fixed as in e. Recall that $\phi(e)$ denotes the number of free wires in experiment e, and therefore $\phi(e) \leq n$ for all e.

Lemma 13 *Let C and C' be probabilistic circuits on wires W, and let e be a distribution injection experiment. If for every wire w, the gate for w in C' is η-correct for w with respect to C, then*

$$d(C(e), C'(e)) \leq \phi(e) \cdot \eta.$$

Proof By induction on $\phi(e)$, the number of free wires in e. If $\phi(e) = 0$, then e constrains the output wire, and trivially, $d(C(e), C'(e)) = 0$. Otherwise, the inductive hypothesis is that

$$d(C(e'), C'(e')) \leq \phi(e') \cdot \eta$$

for all experiments e' with fewer than $\phi(e)$ free gates.

By Lemma 2, assume that e is in fact a value injection experiment. Since C' is acyclic, there exists a free wire w in e such that the inputs to w in C' are fixed in e to some k-tuple $(\sigma_1, \ldots, \sigma_k) \in \Sigma^k$. Let f denote the probabilistic gate function for w in C', and let D denote the value distribution $f(\sigma_1, \ldots, \sigma_k)$. Then we have $C'(e) = C'(e|_{w=D})$, and

$$\begin{aligned} d(C(e), C'(e)) &\leq d(C(e), C(e|_{w=D})) + d(C(e|_{w=D}), C'(e|_{w=D})) \\ &\leq \eta + (\phi(e) - 1) \cdot \eta \\ &= \phi(e) \cdot \eta \end{aligned}$$

by the inductive hypothesis and the fact that f is η-correct for w. ■

Corollary 14 *Let C and C' be probabilistic circuits on wires W where $|W| = n$. If for every wire w, the gate g for w in C' is η-correct for w with respect to C, then*

$$d(C(e), C'(e)) \leq n \cdot \eta.$$

Proof By the definition of approximate behavioral equivalence and the bound $\phi(e) \leq n$. ■

Next we show that test paths are sufficient to determine whether a gate is η-correct for a wire in C.

Lemma 15 *Let C be a Boolean probabilistic circuit, w a wire and g' a probabilistic gate. If for every test path p for w that fixes all the inputs of g', $d(C(p), C(p|_{w=g'(p)})) \leq \eta/K_w$, where K_w is the maximum value of $\kappa(e, w)$ for C over all experiments e, then g' is η-correct for w with respect to C.*

Proof Let g be the actual gate that C assigns to w. Let e be a value injection experiment that fixes every input of g'. Then e may not fix all of g's inputs, but because C is acyclic, g's inputs are not reachable from w. By Lemmas 4 and 5, there exists an experiment $e' \leq e$ that fixes g's inputs, with

$$d(C(e'), C(e'|_{w=g'(e')})) \geq d(C(e), C(e|_{w=g'(e)})).$$

Since e' fixes all of g's inputs, $C(e') = C(e'|_{w=g(e')})$. It is given that for all test paths p that fix all inputs of g' that

$$d(C(p|_{w=g(p)}), C(p|_{w=g'(p)})) \leq \eta/K_w,$$

so it follows by Lemma 10 that

$$d(C(e'|_{w=g(e')}), C(e'|_{w=g'(e')})) \leq \kappa(e', w) \cdot \eta/K_w \leq \eta,$$

and g' is η-correct for w. ■

To prove that PCB constructs a circuit C' that is ε-behaviorally equivalent to the target circuit C, we show that for each wire $w \in W$, PCB assigns a gate that is ε/n-correct for w in C.

Assume that $W' \neq W$, that is, that not all wires have been assigned gates, and consider PCB as it attempts to add another gate to C'. PCB looks for a wire $w \in (W - W')$ and probabilistic gate $g' \in G$ with all of its inputs in W' such that for each experiment $e \in U_*$ that leaves w free and fixes all inputs of g',

$$d(\widehat{C}(e), \widehat{C}(e|_{w=g'(e)})) \leq 2\varepsilon/(4n\kappa(n)).$$

If this search succeeds, then by (1),

$$d(C(e), \widehat{C}(e)) \leq \varepsilon/(4n\kappa(n))d(\widehat{C}(e|_{w=g'(e)}), C(e|_{w=g'(e)})) \leq \varepsilon/(4n\kappa(n)),$$

and thus by the triangle inequality we have

$$d(C(e|_{w=g'(e)}), C(e)) \leq \varepsilon/(n\kappa(n)),$$

It follows by Lemma 15 and the choice of $\kappa(n)$ that g' is ε/n-correct for w in C.

To see that the search for a gate will succeed as long as $W' \neq W$, we note that because C is acyclic, there is some wire $w \in (W - W')$ such that all of w's inputs in C are in W'. Let g denote the gate assigned by C to w, with inputs $w_1, \ldots, w_r$ and probabilistic gate function f. By the existence of g, there is at least one feasible gate-wire assignment for PCB to make, ensuring the continued progress of PCB. Consider any experiment $e \in U_*$ that leaves w free and fixes the inputs of g to $(\sigma_1, \ldots, \sigma_r)$. Let D be the value distribution $f(\sigma_1, \ldots, \sigma_r)$. Then $C(e) = C(e|_{w=D})$ and by (1) and (2) we have

$$\begin{aligned} d(\widehat{C}(e), C(e)) &\leq \varepsilon/(4n\kappa(n)) \\ d(C(e|_{w=D}), \widehat{C}(e|_{w=D})) &\leq \varepsilon/(4n\kappa(n)), \end{aligned}$$

so by the triangle inequality,

$$d(\widehat{C}(e), \widehat{C}(e|_{w=D})) \leq 2\varepsilon/(4n\kappa(n)).$$

Therefore, PCB will continue to make progress until it has assigned a gate to every wire in W, and every such gate will be ε/n-correct for its wire in C, which means that C' will be ε-behaviorally equivalent to C.

5.6 Running time

To bound the running time of PCB we argue as follows. The set U of experiments is of cardinality $n^{O(kc)}$ and can be constructed in time polynomial in its size. To estimate $C(e)$, each experiment in U is repeated

$$O((n\kappa(n)/\varepsilon)^2 \log(|U|/\delta))$$

times; recall that $\kappa(n) = O(n^{c(\log k+1)})$. PCB then chooses a gate for a wire n times. For each choice, it must at worst iterate over $O(n)$ wires in $(W - W')$, over all $O(n^k)$ choices of k or fewer input wires from W', over all $|\Sigma|^k$ assignments of values to the input wires, and all experiments in U. Thus the running time of PCB is polynomial in n, $1/\varepsilon$ and $1/\delta$.

6. Lower Bounds on Path Attenuation

The path attenuation bound $\kappa(n)$ is a significant factor in the running time of the PCB algorithm. In this section we consider lower bounds on path attenuation for Boolean probabilistic circuits. The following theorem shows that the bound of $\pi(e, w)$ for transitively reduced Boolean probabilistic circuits in Corollary 11 is tight infinitely often.

Theorem 16 *There is an infinite set of transitively reduced probabilistic Boolean circuits such that for each circuit C in the family, there exists a value injection experiment e and a wire w free in e such that*

$$d(C(e|_{w=0}), C(e_{w=1})) = 1$$

and for every test path p for w we have

$$d(C(p|_{w=0}), C(p|_{w=1})) = 1/\pi(e, w).$$

Proof For each positive integer ℓ, define the circuit C_ℓ to be a chain of ℓ copies of the circuit C_1 in Figure 1 with wire w_4 of one copy identified with wire w_1 of the next copy. More formally, the $3\ell + 1$ wires are $w_{0,4}$ and $w_{i,j}$ for $i = 1, \ldots, \ell$ and $j = 2, 3, 4$. The output wire is $w_{\ell,4}$. The wire $w_{0,4}$ has no inputs and is determined by an unbiased coin flip, that is, $U(\{0, 1\})$. The wires $w_{i,2}$ and $w_{i,3}$ are the outputs of deterministic identity gates with input $w_{i-1,4}$. The wire $w_{i,4} = A(w_{i,2}, w_{i,3})$ is the result of applying the two-input averaging probabilistic gate function A to the wires $w_{i,2}$ and $w_{i,3}$. The circuit C_3 is depicted in Figure 5.

To understand the operation of this circuit in response to a value injection experiment e, we may view each averaging gate as choosing one of its inputs to copy to its output. Starting at the output wire, this determines a path back to the first wire whose value has been fixed, or to the wire $w_{0,4}$ (which has no inputs) and the output of the circuit is the value of the wire so reached.

Define experiment e to leave all of the wires free. Let w denote the wire $w_{0,4}$. Clearly there are 2^ℓ paths on free gates in e from w to the output gate, that is, $\pi(w, e) = 2^\ell$. For experiment e every possible path starts at wire w and we have $C(e|_{w=0}) = 0$ and $C(e|_{w=1}) = 1$, so $d(C(e|_{w=0}), C(e|_{w=1})) = 1$. However, any test path p for w must fix one of the wires $w_{i,2}$ or $w_{i,3}$ for each $i = 1, \ldots, \ell$. Thus, there is exactly one path that leads back to wire w, and this path is the one chosen by the averaging gates with probability $1/2^\ell$. Thus the result for any test path p for w is $d(C(p|_{w=0}), C(p|_{w=1})) = 1/2^\ell = 1/\pi(e, w)$. ■

This lower bound also holds for general transitively reduced circuit topologies, as follows. (Note that this result was incorrectly stated in the preliminary version of this paper (Angluin et al., 2008a).)

Theorem 17 *Let G be a transitively reduced acyclic directed graph with a designated output node z that is reachable from every node. For each node w there exists a Boolean probabilistic circuit C whose circuit graph is G with output wire z such that for every value injection experiment e that leaves w free and for every test path $p \leq e$ for wire w we have*

$$d(C(e|_{w=1}), C(e|_{w=0})) \geq \pi(e, w) \cdot d(C(p|_{w=1}), C(p|_{w=0})).$$

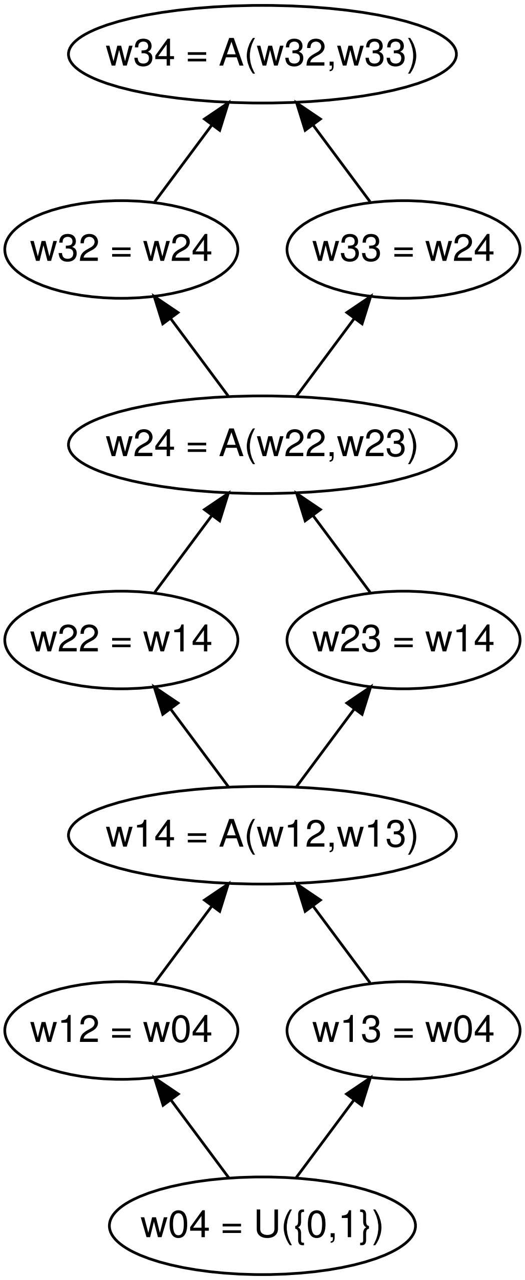


Figure 5: The circuit C_3; $w_{3,4}$ is the output wire.

Proof Let w be given. To construct C, each node v of G is assigned a probabilistic gate whose inputs are the in-neighbors of v in G, as follows. For each node v, let $P(v)$ denote the number of distinct directed paths from w to z that include node v, and for each edge (u, v), let $P(u, v)$ denote the number of distinct directed paths from w to z that include edge (u, v). If there are no paths from w to z through v (that is, $P(v) = 0$) then we let the probabilistic gate function for v be the constant function 0. The probabilistic gate function for w is a coin flip, $U(\{0, 1\})$.

Otherwise, if node v has inputs $u_1, \ldots, u_r$ then it is assigned the probabilistic gate function specified by

$$A_v(b_1, \ldots, b_r) = \sum_{i=1}^{r} b_i \cdot P(u_i, v)/P(v)$$

This generalizes the two-input averaging gate A, weighting input u_i by the fraction of paths from w to z passing through v that also pass through u_i. We may view A_v as performing a random weighted selection of one of its inputs to copy to its output. The weights have been chosen so that each directed path from w to z is selected with probability $1/P(w)$.

Let e be any value injection experiment that leaves w free. If there is no path on free wires in e from w to the output, then $\pi(e, w) = 0$, and the bound in the conclusion of the lemma holds trivially. Otherwise, the output of the circuit in response to e is determined by tracing from the output wire, following the choices of the averaging gates, until either the first wire fixed by e, or w, is reached. Thus

$$d(C(e|_{w=1}), C(e|_{w=0})) = \pi(e, w)/P(w),$$

because there are $\pi(e, w)$ paths from w to the output wire in e. Let $p \leq e$ be any test path for w; now there is just one choice of path that leads back to w, so

$$d(C(p|_{w=1}), C(p|_{w=0})) = 1/P(w),$$

establishing the conclusion of the lemma. ■

Can the general bound in Lemma 10 be improved to the bound for transitively reduced circuits in Corollary 11? The following example shows that the better bound is in general not attainable if the circuit is not transitively reduced. It gives a family of circuits of depth 2ℓ for which the worst-case ratio of the differences shown for w by an experiment e and the best path for w is $(5/4)^\ell \pi(e, w)$.

Theorem 18 *There exists an infinite set of Boolean probabilistic circuits $D_1, D_2, \ldots$ such that for each ℓ there exists a value injection experiment e and a wire w free in e such that $\pi(e, w) = 4^\ell$ and*

$$d(D_\ell(e|_{w=0}), D_\ell(e|_{w=1})) = (5/7)^\ell,$$

but for any test path p for w,

$$d(D_\ell(p|_{w=0}), D_\ell(p|_{w=1})) = (1/7)^\ell.$$

Proof We first define a Boolean probabilistic circuit D_1 and then connect ℓ copies of it in series to get D_ℓ. The wires of D_1 are $w_1, \ldots, w_5$. They are connected as in Figure 6; the output wire is w_5. Note that the edge (w_1, w_5) means that the circuit graph is not transitively reduced. The gate function G is defined by giving its expected value as a function of its inputs:

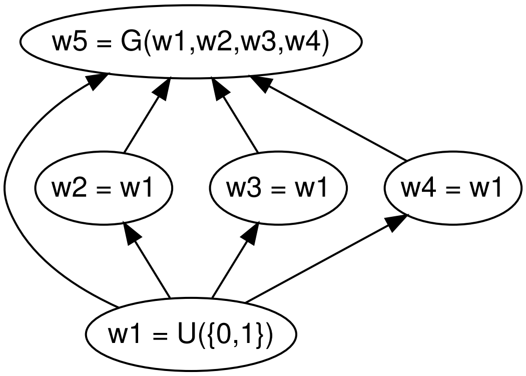


Figure 6: The circuit D_1; w_5 is the output wire.

$$E[G(w_1, w_2, w_3, w_4)] = ((1 - w_1) + 2w_2 + 2w_3 + 2w_4)/7.$$

Let e be the experiment that leaves all five wires free. It is clear that

$$d(D_1(e|_{w_1=0}), D_1(e|_{w_1=1})) = 5/7.$$

We now show that for any test path p for w_1,

$$d(D_1(p|_{w_1=0}), D_1(p|_{w_1=1})) = 1/7.$$

The possible test paths p for w_1 either fix all of w_2, w_3, w_4 or all but one of them. Thus, as we change from $w_1 = 0$ to $w_1 = 1$ in such a test path, the assignments to wires (w_1, w_2, w_3, w_4) change in one of four possible ways:

$(0, b_2, b_3, b_4)$ to $(1, b_2, b_3, b_4)$

$(0, 0, b_3, b_4)$ to $(1, 1, b_3, b_4)$

$(0, b_2, 0, b_4)$ to $(1, b_2, 1, b_4)$

$(0, b_2, b_3, 0)$ to $(1, b_2, b_3, 1)$

Checking each of these possible changes against the definition of G, we see that each change produces a difference of 1/7, as claimed. (This example can be modified to give a difference of 1 versus 1/5.) Thus, setting $w = w_1$, the circuit D_1 gives the base case of the claim in the lemma.

To construct D_ℓ, we take ℓ copies of D_1 and identify wire w_5 in one copy with wire w_1 in the next copy, making the wire w_5 of the final copy the output wire of the whole circuit. Let w denote the wire w_1 in the first such copy. Then $\pi(e, w) = 4^\ell$ and

$$d(D_\ell(e|_{w=0}), D_\ell(e_{w=1})) = (5/7)^\ell.$$

For any test path p, the signal is attenuated by a factor of $1/7$ for each level, and we have

$$d(D_\ell(p|_{w=0}), D_\ell(p|_{w=1})) = 1/7^\ell.$$

■

This construction can be generalized to $k+1$ wires for any odd $k+1$, which increases the attenuation. In the base circuit there are k paths and an attenuation factor of $1/(2k-3)$, and the worst-case ratio of differences for an experiment and its test paths in D_ℓ approaches $2^\ell \pi(e, w)$ as k goes to infinity.

7. Exponential Dependence on Depth

The bounds on path attenuation show that test paths may be much less informative than general value injection experiments, resulting in the exponential dependence of the number of experiments and the running time of PCB on the depth of the target circuit. It is natural to ask whether we might do better by using selected general experiments. In this section, we give computational evidence to the contrary. The following result contrasts with the case of deterministic circuits, where the Distinguishing Paths algorithm uses value injection queries to learn arbitrary transitively reduced acyclic deterministic circuits of constant fan-in over polynomial size alphabets in polynomial time (Angluin et al., 2008b).

Theorem 19 *If* **BPP** $\neq$ **NP** *and* $k \geq 4$ *then there is no polynomial time algorithm using value injection queries that approximately learns all acyclic transitively reduced Boolean probabilistic circuits with fan-in bounded by* k*.*

Proof Suppose L is a polynomial time algorithm that approximately learns the behavior of every transitively reduced acyclic Boolean probabilistic circuit of fan-in bounded by 4 using value injection queries. The hard computational problem we consider is the following: given a satisfiable 3-CNF formula ϕ over the variables $x_1, \ldots, x_n$ with clauses $c_1, \ldots, c_m$, find an assignment to the variables that satisfies significantly more than seven-eights of the clauses of the formula. Finding such an assignment is **NP**-hard by a result of Håstad (2001). We show how to transform the 3-CNF formula ϕ into a pair of transitively reduced circuits C_0 and C_1 with maximum fan-in 4 such that value injection experiments show a difference that is exponentially small in the depth of the circuits unless we can find a variable assignment that satisfies significantly more than seven-eighths of the clauses of the formula.

The efficiency of our construction depends on the existence of a family of graphs with an expansion property. Specifically, there exists a constant $\alpha < 1$ such that for sufficiently large m, there exists a directed graph G_m on m nodes with constant out-degree 3 such that the second largest eigenvalue λ_2 of the transition matrix for a random walk on G_m satisfies $\lambda_2 \leq \alpha$. Such a family can be constructed by the probabilistic method and explicit constructions are also known; these are surveyed by Hoory, Linial, and Wigderson (2006). Let r be the smallest integer such that $\alpha^r \leq 1/40$.

Let ℓ be a positive integer. The two circuits C_0 and C_1 differ only in their default assignments to a subset of their wires, so we describe their common structure as follows. The circuit consists of a stack of ℓ repetitions of a block consisting of r expander layers

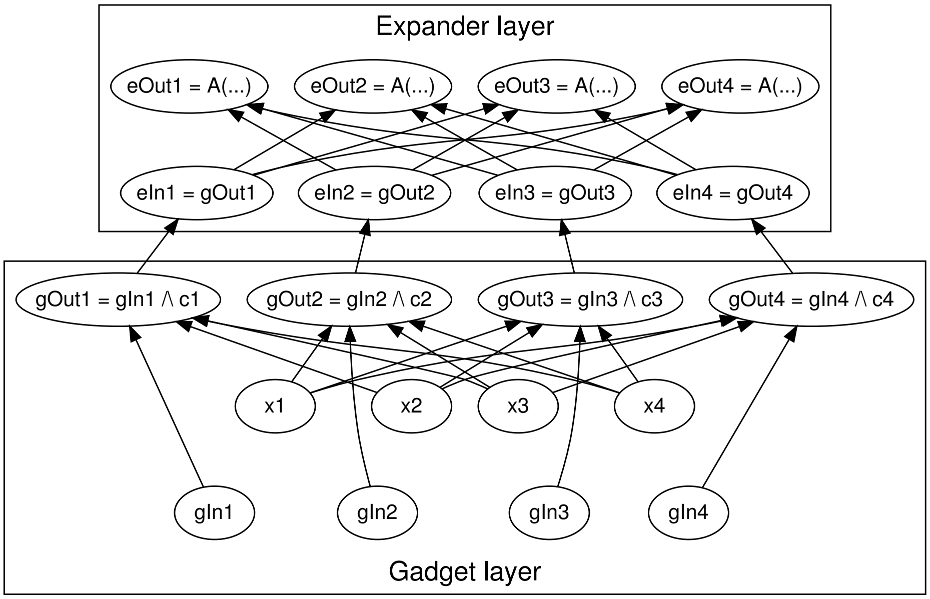


Figure 7: A block with $r = 1$ for the Boolean formula $c_1 \wedge c_2 \wedge c_3 \wedge c_4$, where $c_1 = x_2 \vee \overline{x_3} \vee x_4$ and $c_2 = \overline{x_1} \vee x_3 \vee \overline{x_4}$ and $c_3 = x_1 \vee \overline{x_2} \vee x_4$ and $c_4 = \overline{x_1} \vee x_2 \vee \overline{x_3}$.

above one gadget layer for a total depth of $(2r + 1)\ell$. Figure 7 illustrates a block consisting of one expander layer ($r = 1$) above a gadget layer. Recall that $x_1, \ldots, x_n$ are the variables of ϕ and $c_1, \ldots, c_m$ are the clauses of ϕ.

A **gadget layer** has three types of wires: inputs $\mathrm{gIn}_1, \ldots, \mathrm{gIn}_m$, variables $\mathrm{x}_1, \ldots, \mathrm{x}_n$, and outputs $\mathrm{gOut}_1, \ldots, \mathrm{gOut}_m$. The input wire gIn_j of each gadget layer except the initial one is identified with the corresponding output wire eOut_j of the expander layer just below it. The variable wires x_i of each gadget layer have no inputs and default to the constant 0. Each output wire gOut_j has four inputs: the corresponding gadget input wire gIn_j and the three variable wires for the variables of the clause c_j of ϕ. Its gate function computes the conjunction of gIn_j and the value of the clause c_j given its three variable values.

Thus, if the learner sets the variable wires x_i in a gadget layer according to a satisfying assignment of ϕ, the signals propagate from the gadget inputs gIn_j to their corresponding outputs gOut_j with perfect fidelity. Otherwise, any unsatisfied clause blocks the signal for the corresponding output.

An **expander layer** averages the outputs of the layer below to be the inputs for the layer above, according to the expander graph G_m. Each input eIn_j of an expander layer is set equal to the corresponding output of the gadget or expander layer immediately below it. The three inputs to eOut_j are eIn_k for the three out-neighbors k of j in the expander graph G_m. The gate function for each eOut_j is the three-input averaging gate $A(x, y, z)$, which is 1 with probability $(x + y + z)/3$. The output of the whole circuit is the first output wire of the final (topmost) expander layer.

The **initial inputs** are the input wires gIn_j of the initial gadget layer. They have no inputs; for the circuit C_0 they are all assigned the default value 0, and for the circuit C_1 they are all assigned the default value 1. Note that C_0 and C_1 are transitively reduced and have a maximum fan-in of 4.

The challenge for the learner is to determine which of C_0 and C_1 is the target circuit. If a value injection experiment succeeds in setting the variable wires in every gadget layer to (possibly different) satisfying assignments for the formula ϕ and leaves all other wires free, then the output of C_0 is 0 and the output of C_1 is 1. If not all the clauses of ϕ are satisfied, then this distance is reduced.

Intuitively, the learner's strategy must be to fix the variable wires in each gadget layer to prevent the signal from the initial inputs from getting blocked; fixing the input or output wires of gadget or expander layers would not help, because they would then have the same value regardless of their inputs. Without a good variable assignment, however, the signal strength drops by a constant factor for each layer, as we now show.

Let e be a value injection experiment. The experiment e induces an assignment to the variables of ϕ for each gadget layer, either by fixing the value of each variable wire or letting it default to 0. The effect of an averaging gate is to select one of its inputs at random and copy the value of that input to the output. Thus, the output of the circuit for experiment e is in effect determined by a random walk backward from the output wire until the walk reaches a wire whose value is fixed by e (and the output is the fixed value), or a gadget layer output wire corresponding to an unsatisfied clause (and the output is 0), or an initial input wire (and the output is the value of that wire.) Suppose that for each gadget layer e encodes a variable assignment that satisfies at most $(9/10)m$ of the clauses of ϕ. We show that the probability that the random walk hits an initial input wire is bounded above by $(39/40)^{\ell}\sqrt{m}$.

Without loss of generality we may assume that e fixes no wires other than variable wires and initial input wires, because any other fixed wires reduce the probability of reaching an initial input. For $i = 1, \ldots, \ell$, let W_i be the $m \times m$ diagonal matrix with 1s for each satisfied clause in the ith gadget layer and 0s for each unsatisfied clause. Let B be the transition matrix for an r-step random walk on G_m and let $e_1 = (1, 0, \ldots, 0)$. The probabilities of the random walk hitting the initial inputs are given by the vector $e_1 B W_\ell B W_{\ell-1} \cdot B W_2 B W_1$. By the following argument, for all i and vectors v, we have $\|vBW_i\| \leq (39/40)\|v\|$.

Write $v = cu + w$, where c is a scalar and $u = (1, \ldots, 1)$ and w is a vector such that $u \perp w$. Then u is an eigenvector of B with eigenvalue 1 and multiplying w by B shrinks its length to at most the second eigenvalue of B times its original length. By Pythagoras, $\|cu\| \leq \|v\|$ and $\|w\| \leq \|v\|$. We have $vBW_i = (cu + w)BW_i$. On one hand, $\|cuBW_i\| = \|cuW_i\| \leq \sqrt{9/10}\|cu\| \leq (19/20)\|v\|$. On the other hand, $\|wB\| \leq (1/40)\|w\| \leq (1/40)\|v\|$, because the second eigenvalue of B is no larger than 1/40, and $\|wBW_i\| \leq (1/40)\|v\|$, because W_i does not increase the L_2 norm. The resulting $(39/40)^{\ell}$ bound on the L_2 norm of the probability vector gives a bound of $(39/40)^{\ell}\sqrt{m}$ on the L_1 norm, which is an upper bound on the probability that any initial input is reached.

Suppose the learning algorithm L runs in time $f(N, 1/\varepsilon, 1/\delta)$, for some nondecreasing polynomial f, where N is the number of wires in the target circuit. Let $N(\ell)$ denote the number of wires in C_0 (or C_1) as a function of the number ℓ of blocks in the stack. Then

$N(\ell) = O(\ell(n + rm))$. We choose ℓ sufficiently large that

$$((39/40)^{\ell}\sqrt{m})f(N(\ell), 4, 4) < 1/4,$$

clearly $N(\ell)$ is bounded by a polynomial in m and n.

We randomly and equiprobably choose the target circuit C to be C_0 or C_1 and simulate L with target circuit C and $\varepsilon = \delta = 1/4$. When L makes a value injection experiment e, we check whether any of the induced variable assignments of e satisfies more than $(9/10)m$ clauses of ϕ. If so, we output the assignment and halt. Otherwise, we use a random walk from the output wire in the circuit C to give an output for e. If no experiment e satisfies more than $(9/10)m$ of the clauses of ϕ, then the probability that any of them reaches an initial input in C is less than $1/4$. If none of them reaches an initial input, then L cannot distinguish between C_0 and C_1, and must output a circuit that is not $1/4$-approximately behaviorally equivalent to C with probability at least $3/8 > 1/4$, violating the requirements of approximate learning. ■

We conclude that if **BPP** $\neq$ **NP**, any polynomial time learning algorithm requires in expectation exponentially many queries in ℓ to learn the default settings of the initial inputs and therefore, PCB is within a polynomial of optimal.

8. Non-Boolean Circuits Revisited

The sharp contrast in results for transitively reduced circuits with alphabet size at least three, for which test paths may show no difference (Lemma 8) and those with alphabet size two, for which test paths must show a significant difference (Lemma 10) motivate us to consider a generalization of the kinds of experiments we consider, to function injection experiments. This generalization allows us to extend the results of Lemma 10 to non-Boolean alphabets.

In a value injection experiment, each wire is either fixed to a constant value or left free. In a function injection experiment for a wire w, these possibilities are expanded to permit a transformation of the value that the wire w would take if it were left free. As an example, consider a transformation in which the values that w could attain are linearly ordered and all values below a certain threshold are mapped to the minimum value and all other values are mapped to the maximum value. It is conceivable that this kind of transformation could be feasible in some domains; in any case, the theoretical consequences are quite interesting. We first give a general definition of function injection, but in the results below we are primarily concerned with 2-partitions, that is, transformations that are like the above example in that they partition the values into at most two blocks and map each block to a fixed element of the block.

An **alphabet transformation** is a function f that maps symbols to distributions over symbols. An alphabet transformation is **deterministic** if it assigns only deterministic distributions, in which case we think of it as a map from symbols to symbols. A deterministic alphabet transformation f is a k-**partition** if there exists a partition of Σ into at most k disjoint nonempty sets Σ_i such that for each i there exists $\sigma_i \in \Sigma_i$ such that $f(\Sigma_i) = \{\sigma_i\}$. Note that if $k_1 \leq k_2$, every k_1-partition is also a k_2-partition.

A 1-partition is a constant function, achieving the same result as fixing the wire to a value in a value injection experiment. We use 2-partitions to reduce the case of larger alphabets to the binary case. Note that the 2-partitions of a binary alphabet include the identity and the two constant functions, but not the negation function.

If D is a value distribution and f is an alphabet transformation, then $f(D)$ is the value distribution in which

$$(f(D))(\sigma) = \sum_{\tau \in \Sigma} D(\tau)(f(\tau))(\sigma).$$

A **function injection experiment** is a mapping e with domain W that assigns to each wire the symbol $*$ or a symbol from Σ or an alphabet transformation f. Then e leaves w **free** if $e(w) = *$, **fixes** w if $e(w) \in \Sigma$, and **transforms** w if $e(w)$ is an alphabet transformation f. We extend the ordering $\leq$ on experiments by stipulating that each alphabet transformation $f \leq *$. A **2-partition experiment** is a function injection experiment in which every alphabet transformation is a 2-partition.

We now define the joint probability distribution on assignments of symbols from Σ to wires determined by a function injection experiment e. If e fixes w, then w is just assigned $e(w)$. Otherwise, if the inputs of w have been assigned the values $\sigma_1, \ldots, \sigma_k$ and f is the gate function for w, we randomly and independently choose a symbol σ according to the value distribution $f(\sigma_1, \ldots, \sigma_k)$. If w is free in e, then σ is the symbol assigned to w; however, if $e(w)$ is an alphabet transformation, then a symbol τ is chosen randomly and independently according to the value distribution $e(\sigma)$ and assigned to w. That is, when $e(w)$ is an alphabet transformation, we generate the symbol for w as though it were free, and then use the distribution $e(w)$ to transform that symbol. Because C is acyclic, this process assigns a symbol to every wire of C.

In a **function injection query** (FIQ), the learning algorithm gives a function injection experiment e and receives a symbol σ assigned to the output wire of C by the probability distribution defined above. A **functional test path** for a wire w is a function injection experiment in which the free and transformed wires are a directed path in the circuit graph from w to the output wire, and all other wires are fixed.

As an example of how functional test paths help in learning non-Boolean probabilistic circuits, consider again the circuit in the proof of Lemma 8, depicted in Figure 3. We specify a functional test path p by $p(w_1) = p(w_4) = p(w_5) = *$, $p(w_3) = 0$ and $p(w_2)$ is the alphabet transformation $0 \to 0$, $1 \to 0$, and $2 \to 2$. Note that the alphabet transformation is a 2-partition. Then $C(p|_{w_1=0}) = 0$ but $C(p|_{w_1=2}) = U(\{0,1\})$, so this functional test path witnesses a difference of $1/2$, as large as the experiment that leaves all the wires free. Test paths with functions allow us to carry over the results of Lemma 10 to non-Boolean alphabets.

Lemma 20 *Let C be a probabilistic circuit, e be a function injection experiment, w be a wire free in e and D_1, D_2 be value distributions. If there exists $\varepsilon \geq 0$ such that for all functional w-test paths $p \leq e$ that are 2-partitions,*

$$d(C(p|_{w=D_1}), C(p|_{w=D_2})) \leq \varepsilon,$$

then

$$d(C(e|_{w=D_1}), C(e|_{w=D_2})) \leq \kappa(e, w) \cdot \varepsilon.$$

Proof The obstacle in Lemma 10 is that when the alphabet is non-Boolean, we may assume only that D_1 and D_2 have disjoint support, not that they are deterministic. This obstacle can be overcome by injecting a 2-partition at w. Let $\Sigma_1 = \text{support}(D_1)$ and $\Sigma_2 = \text{support}(D_2)$ and assume $\Sigma_1 \cap \Sigma_2 = \emptyset$. Then

$$d(C(e|_{w=D_1}), C(e|_{w=D_2})) \leq \sum_{\substack{\rho_1 \in \Sigma_1 \\ \rho_2 \in \Sigma_2}} D_1(\rho_1) D_2(\rho_2) d(C(e|_{w=\rho_1}), C(e|_{w=\rho_2}))$$

by the triangle inequality. Let

$$(\sigma, \tau) = \arg\max_{\substack{\rho_1 \in \Sigma_1 \\ \rho_2 \in \Sigma_2}} d(C(e|_{w=\rho_1}), C(e|_{w=\rho_2}))$$

so that

$$d(C(e|_{w=D_1}), C(e|_{w=D_2})) \leq d(D_1, D_2) d(C(e|_{w=\sigma}), C(e|_{w=\tau})).$$

Let f be an alphabet transformation that maps Σ_1 to σ and Σ_2 to τ and all other symbols to either σ or τ. Then f is a 2-partition, and

$$d(C(e|_{w=D_1}), C(e|_{w=D_2})) \leq d(C(e|_{w=f(D_1)}), C(e|_{w=f(D_2)})).$$

Since $f(D_1) = \sigma$ and $f(D_2) = \tau$, the rest of the proof goes through. ■

Corollary 21 *Let C be a transitively reduced probabilistic circuit, e be a function injection experiment, w be a wire, and D_1, D_2 be value distributions. If there exists $\varepsilon \geq 0$ such that for all functional w-test paths $p \leq e$,*

$$d(C(p|_{w=D_1}), C(p|_{w=D_2})) \leq \varepsilon,$$

then

$$d(C(e|_{w=D_1}), C(e|_{w=D_2})) \leq \pi(e, w) \cdot \varepsilon.$$

We expect that a further generalization of the Probabilistic Circuit Builder algorithm to use function injection experiments can learn non-Boolean circuits of logarithmic depth and constant fan in in polynomial time. The universal set would map wires to the set containing all alphabet symbols from Σ and all 2-partitions of Σ, of which there are fewer than $|\Sigma|^2 2^{|\Sigma|}$. Thus, the universal set will still be of size $n^{O(1)}$, suggesting that a polynomial time algorithm may be attainable in this case.

Certain other natural questions arise in response to the idea of function injection experiments. We can define circuits C and C' to be **strongly behaviorally equivalent** if $C(e) = C'(e)$ for every function injection query e. Does behavioral equivalence imply strong behavioral equivalence? Once again, alphabet size determines the answer: no for alphabet size greater than two, yes for alphabet size two.

Lemma 22 *For $\Sigma = \{0, 1, 2\}$, there exist deterministic circuits C_1 and C_2 that are behaviorally equivalent but not strongly behaviorally equivalent.*

Proof In both C_1 and C_2 there are two wires w_1 and w_2, where w_2 is the output wire. In both circuits the gate for w_2 has input w_1 and deterministically maps 0 to 0 and maps 1 and 2 to 1. In C_1, w_1 is the constant 1 and C_2 it is the constant 2.

Then if e is the value injection experiment that leaves both wires free, $C_1(e) = 1 = C_2(e)$. If e fixes either w_1 or w_2, then also $C_1(e) = C_2(e)$. Thus C_1 is behaviorally equivalent to C_2.

However, the 2-partition function injection experiment e that leaves w_2 free and maps the output of w_1 according to the transformation $0 \to 0$, $1 \to 0$, $2 \to 2$ yields $C_1(e) = 0$ and $C_2(e) = 1$. Thus C_1 is not strongly behaviorally equivalent to C_2. ■

However, 2-partition function experiments suffice to establish strong behavioral equivalence.

Lemma 23 *Let C and C' be probabilistic circuits with the same alphabet Σ, the same set of wires and the same output wire. If $C(e) = C'(e)$ for every 2-partition function experiment e then C and C' are strongly behaviorally equivalent.*

Proof By a generalization of the Probabilistic Circuit Builder algorithm to functional test paths. ■

Because in the Boolean case every 2-partition function injection query is a value injection query, we then have the following.

Corollary 24 *For Boolean probabilistic circuits C and C', if C is behaviorally equivalent to C' then C is strongly behaviorally equivalent to C'.*

9. Discussion and Open Problems

These results concern general probabilistic acyclic gates, with no restriction other than fan-in on the kinds of probabilistic gate functions considered. Particular domains may warrant specific assumptions about the gate functions, which may make the learning problems more tractable. For example, for the problem of learning the structure of an independent cascade social network using exact value injection queries, a query-optimal algorithm is presented by Angluin et al. (2008c). Note that social networks may in general contain cycles, which complicates their analysis.

The Distinguishing Paths algorithm (Angluin et al., 2008b) learns transitively reduced acyclic deterministic circuits over polynomial size alphabets with constant fan-in and no depth bound using value injection queries in polynomial time, and relies on a version of the test path lemma. Theorem 19 shows that if $\mathbf{BPP} \neq \mathbf{NP}$ then this algorithm does not generalize to arbitrary transitively reduced Boolean probabilistic circuits, but there is a possibility that it might generalize to transitively reduced Boolean probabilistic circuits with a polynomial bound on the total number of directed paths in the circuit graph. A somewhat technical open question is whether in the case of general Boolean probabilistic circuits, the ability to inject the NOT function might reduce the maximum path attenuation to just the number of paths, as it does in the case of the circuit in Figure 6.

Acknowledgments

James Aspnes acknowledges the support of NSF grant CNS-0435201. This work was done while Jiang Chen was a member of the Center for Computational Learning Systems, Columbia University; he acknowledges the support of a research contract from Consolidated Edison. Lev Reyzin acknowledges that this material is based upon work supported under a National Science Foundation Graduate Research Fellowship. A preliminary version of this paper was presented at COLT 2008 (Angluin et al., 2008a). The authors thank the reviewers of the COLT 2008 version of the paper and the referees of the present paper for their thoughtful comments.

References

Tatsuya Akutsu, Satoru Kuhara, Osamu Maruyama, and Satoru Miyano. Identification of genetic networks by strategic gene disruptions and gene overexpressions under a boolean model. *Theor. Comput. Sci.*, 298(1):235–251, 2003.

Dana Angluin and Michael Kharitonov. When won't membership queries help? *J. Comput. Syst. Sci.*, 50(2):336–355, 1995.

Dana Angluin, James Aspnes, Jiang Chen, David Eisenstat, and Lev Reyzin. Learning acyclic probabilistic circuits using test paths. In *Twenty-First Annual Conference on Learning Theory*, pages 169–179. Omicron, July 2008a.

Dana Angluin, James Aspnes, Jiang Chen, and Lev Reyzin. Learning large-alphabet and analog circuits with value injection queries. *Machine Learning*, 72(1-2):113–138, 2008b.

Dana Angluin, James Aspnes, and Lev Reyzin. Optimally learning social networks with activations and suppressions. In *Nineteenth International Conference on Algorithmic Learning Theory*, volume 5254 of *Lecture Notes in Computer Science*, pages 272–286, October 2008c.

Dana Angluin, James Aspnes, Jiang Chen, and Yinghua Wu. Learning a circuit by injecting values. *J. Comput. Syst. Sci.*, 75(1):60–77, 2009.

Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe´er. Using Bayesian networks to analyze expression data. *J. Comput. Biol.*, 7(3–4):601–620, 2000.

Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc. (N.S.)*, 43(4):439–561 (electronic), 2006.

Trey E. Ideker, Vesteinn Thorsson, and Richard M. Karp. Discovery of regulatory interactions through perturbation: Inference and experimental design. In *Pacific Symposium on Biocomputing 5*, pages 302–313, 2000.

David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD '03: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146, New York, NY, USA, 2003. ACM.

David Kempe, Jon M. Kleinberg, and Éva Tardos. Influential nodes in a diffusion model for social networks. In *ICALP*, pages 1127–1138, 2005.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.