Hitting sets derandomize BPL (lecture notes)

Course: Derandomizing Space-Bounded Computation, Winter 2025, University of Chicago
Instructor: William Hoza (williamhoza@uchicago.edu)

A *hitting set* is a weak, barebones version of a PRG. The definition is as follows.

Definition 0.1 (Hitting sets)**.** Let $\mathcal{F}$ be a class of functions $f\colon \{0,1\}^n \to \{0,1\}$. An ε-hitting set for $\mathcal{F}$ is a set $H \subseteq \{0,1\}^n$ such that for every $f \in \mathcal{F}$, if $\mathbb{E}[f] > \varepsilon$, then there is some $x \in H$ such that $f(x) = 1$.

Observe that if G is a PRG that fools $\mathcal{F}$ with error ε, then the image of G is an ε-HSG for $\mathcal{F}$. In fact, you can double check that the same is true if (G, ρ) is a *weighted* PRG that fools $\mathcal{F}$ with error ε.

$$\text{PRG} \implies \text{WPRG} \implies \text{Hitting Set}.$$

Hitting sets are potentially much easier to construct than weighted or unweighted PRGs. This raises the question: If and when we can construct hitting sets, what are they good for?

It is a simple exercise to show that if there exists a $\frac{1}{2}$-hitting set for width-n length-n standard-order ROBPs that can be computed using $O(\log n)$ bits of space,[1] then $\mathsf{L} = \mathsf{RL}$.[2] In these lecture notes, we prove that $\mathsf{L} = \mathsf{BPL}$ under the same assumption.

Theorem 0.2 (Hitting sets derandomize BPL [CH22])**.** *Assume that for every $n \in \mathbb{N}$, there exists a $\frac{1}{2}$-hitting set $H \subseteq \{0,1\}^n$ for width-n length-n standard-order ROBPs that can be computed using $O(\log n)$ bits of space given n. Then $\mathsf{L} = \mathsf{BPL}$.*

1 Proof idea

Suppose we are given the description of a width-n length-n standard-order ROBP f. Our goal is to estimate $\mathbb{E}[f]$ to within ± 0.1 using $O(\log n)$ bits of space. How is a hitting set H helpful? The first idea is that we can interpret each $x \in H \subseteq \{0,1\}^{\text{poly}(n)}$ as the truth table of a candidate PRG, which we denote $G^{(x)}\colon \{0,1\}^{O(\log n)} \to \{0,1\}^n$. Roughly speaking, our plan is to use the hitting property of H to argue that there is at least one $x \in H$ such that $G^{(x)}$ is a good PRG. Then we will use $G^{(x)}$ to estimate $\mathbb{E}[f]$ by trying all seeds.

One difficulty with this approach is that we need a way to figure out which candidate PRG $G^{(x)}$ is "good." The key is the inverse Laplacian perspective. Let M be the transition probability matrix of f and let $L = I - M$ be the Laplacian matrix. As discussed in the previous lecture notes, L is invertible, and the entries of L^{-1} are given by $L^{-1}_{u,v} = \mathbb{E}[f_{u \to v}]$. Each candidate PRG $G^{(x)}$ can be used to compute a candidate approximation $A^{(x)}$ to L^{-1}. To judge whether $G^{(x)}$ is a good PRG, we multiply $A^{(x)} \cdot L$ and compare it to the identity matrix.

2 The details

2.1 The algorithm

Let f be a given width-w length-n standard-order ROBP with start vertex v_{start} and (wlog unique) accepting vertex v_{acc}. Let $N = w \cdot (n+1)$ be the number of vertices in f. For a large enough $t = O(\log(wn))$, let $H \subseteq \{0,1\}^{N^2 \cdot 2^t \cdot n}$ be a $\frac{1}{2}$-hitting set for ROBPs of width $w \cdot (2^t + 1) + 1$ and length $N^2 \cdot t \cdot n$. By assumption, such a hitting set can be computed in $O(\log(wn))$ bits of space.

In Section 1, we suggested that we would interpret each string in H as the truth table of a candidate PRG. Actually, we interpret each string $x \in H$ as a *list* of truth tables of N^2 many candidate PRGs

[1] I.e., there is an $O(\log n)$-space algorithm that prints out the list of all strings in H, given the input n.
[2] Recall that RL is defined like BPL, except that false positives are prohibited.

$G^{(x)}_{u\to v}\colon \{0,1\}^t \to \{0,1\}^n$; there is one candidate PRG for each pair of vertices u, v in f. Define a matrix $A^{(x)} \in [0,1]^{N\times N}$ by the formula

$$A^{(x)}_{u,v} = 2^{-t} \cdot \sum_{y\in\{0,1\}^t} f_{u\to v}(G^{(x)}_{u\to v}(y)).$$

To estimate $\mathbb{E}[f]$, we output $A^{(x)}_{v_{\text{start}},v_{\text{acc}}}$, where $x \in H$ is chosen to minimize the error $\|I - LA^{(x)}\|_1$. Here L is the Laplacian matrix $I - M$, where M is the transition probability matrix of f. You can double check that this algorithm only uses $O(\log(wn))$ bits of space.

2.2 The correctness proof

The proof of correctness is all about the interplay between two measures of error: $\|L^{-1} - A\|_{\max}$, which is a measure of "accuracy," and $\|I - LA\|_1$, which is a measure of "local consistency." Accuracy is what we ultimately care about, but local consistency has the crucial property that it can be efficiently computed.

We begin by showing that if we pick x (the list of truth tables of candidate PRGs) uniformly at random, then the resulting matrix $A^{(x)}$ has good accuracy with high probability.

Lemma 2.1 (A random x has good accuracy)**.** *If we pick $x \in \{0,1\}^{N^2\cdot 2^t\cdot n}$ uniformly at random, then*

$$\Pr_x\left[\|L^{-1} - A^{(x)}\|_{\max} \le 2^{-t/2+1}\cdot\sqrt{\ln(2N)}\right] > \frac{1}{2}.$$

Proof. For each pair of vertices u, v, by Hoeffding's inequality, we have

$$\Pr_x[|A^{(x)}_{u,v} - \mathbb{E}[f_{u\to v}]| \ge \delta] \le 2\exp(-\delta^2\cdot 2^t).$$

Therefore, by the union bound,

$$\Pr_x[\|L^{-1} - A^{(x)}\|_{\max} \ge \delta] \le 2N^2\exp(-\delta^2\cdot 2^t). \qquad \square$$

Next, we show that accuracy can be verified by a polynomial-width ROBP g. Note: We are not claiming that it is possible to *efficiently compute* the description of g given f. We are merely claiming that g *exists.*

Lemma 2.2 (Accuracy can be checked by an ROBP)**.** *For every $\gamma \in (0,1)$, there exists a standard-order ROBP $g\colon \{0,1\}^{N^2\cdot 2^t\cdot n} \to \{0,1\}$ of width $w\cdot(2^t+1)+1$ such that*

$$g(x) = 1 \iff \|L^{-1} - A^{(x)}\|_{\max} \le \gamma.$$

Proof sketch. The program g computes each entry $A^{(x)}_{u,v}$ and compares it to suitable thresholds before moving on to the next entry. To do this, we need to store a state in the simulation of f, and we need to store a counter from 0 to 2^t indicating the number of seeds y we have found such that $f_{u\to v}(G^{(x)}_{u\to v}(y)) = 1$, and we need one "fail" state indicating that we have already found an entry (u,v) such that $|A^{(x)}_{u,v} - L^{-1}_{u,v}| > \gamma$. This is a total width of $w\cdot(2^t+1)+1$. $\square$

Taken together, Definition 0.1 and Lemmas 2.1 and 2.2 imply that there is some $x \in H$ such that $\|L^{-1} - A^{(x)}\|_{\max} \le 2^{-t/2+1}\cdot\sqrt{\ln(2N)}$. To complete the proof, we relate accuracy to local consistency:

Lemma 2.3 (Accurate $\iff$ locally consistent)**.** *For any matrix $A \in [0,1]^{N\times N}$, we have*

$$\frac{1}{2N}\cdot\|I - LA\|_1 \le \|L^{-1} - A\|_{\max} \le (n+1)\cdot\|I - LA\|_1.$$

Proof.

$$\begin{aligned}\frac{1}{2N} \cdot \|I - LA\|_1 = \frac{1}{2N} \cdot \|L \cdot (L^{-1} - A)\|_1 \leq \frac{1}{2N} \cdot \|L\|_1 \cdot \|L^{-1} - A\|_1 &\leq \|L^{-1} - A\|_{\max} \\ &\leq \|L^{-1} - A\|_1 \\ &= \|L^{-1} \cdot (I - LA)\|_1 \\ &\leq \|L^{-1}\|_1 \cdot \|I - LA\|_1 \\ &\leq (n+1) \cdot \|I - LA\|_1. \qquad \square\end{aligned}$$

Recall that our algorithm picks x to minimize $\|I - LA^{(x)}\|_1$. By Lemma 2.3, the error of our algorithm is bounded by

$$\begin{aligned}(n+1) \cdot \min_{x \in H}\{\|I - LA^{(x)}\|_1\} &\leq (n+1) \cdot \min_{x \in H}\{2N \cdot \|L^{-1} - A^{(x)}\|_{\max}\} \\ &\leq (n+1) \cdot 2N \cdot 2^{-t/2+1} \cdot \sqrt{\ln(2N)}.\end{aligned}$$

Choosing a suitable value $t = O(\log(wn))$ completes the proof of Theorem 0.2.

References

[CH22] Kuan Cheng and William M. Hoza. "Hitting Sets Give Two-Sided Derandomization of Small Space". In: *Theory of Computing* 18.21 (2022), pp. 1–32. DOI: 10.4086/toc.2022.v018a021.