PRINCETON UNIVERSITY F'02 COS 597D: A THEORIST'S TOOLKIT

Lecture 5: Talk on "Linear Hashing" Paper

Lecturer: *Sanjeev Arora* Scribe:*Nir Ailon & Iannis Tourlakis*

This talk is based on a paper by Alon, Dietzfelbinger, Miltersen, Petrank and Tardos. It deals with the problem of finding a "good" hash family, in the sense that it is small w.r.t. the entire hash family but still preserves good random properties. A natural family to test is the linear hash family, and it turns out that if $\mathbf{Z}_2$ is the ground field, then the expected largest bucket size property is not very much far from the full hash family case.

1 Covering the bucket space

To prove this result, a preliminary tool is developed, called the "covering property". This relates to the question of whether or not the image a randomly chosen hash function is the entire bucket space. More precisely, we denote by $U = \mathbf{Z}_2^m$ the "universe" space, by $S \subset U$ the set of elements that we want to hash, and by $W = \mathbf{Z}_2^t$ the bucket space. $\mathcal{H} := \{h : U \to W \mid h \text{ linear}\}$ is the linear hash family from which we choose a hash function uniformly at random.

THEOREM 1
(Existential covering property) If $|S| \geq t2^t \log_2 e$, *then* $\exists h \in \mathcal{H}$ *s.t.* $h(S) = W$

Note: This bound is tight in the sense that there is a construction of a slightly smaller set S for which there is no covering linear hash function.

THEOREM 2
(Probabilistic covering property) For any $\epsilon > 0$ *there exists a constant* c_ϵ *s.t. If* $|S| \geq c_\epsilon t2^t$, *then* $Pr_{h\in\mathcal{H}}(h(S) = \mathbf{Z}_2^t) \geq 1 - \epsilon$.

Theorem 1 is not used in the proof of the main result, but it gives a first intuition for the ideas that follow. Its proof is as follows. PROOF: Let $s := m - t$. We construct a growing sequence of sets as follows: $A_0 = S$, $A_1 = A_0 \cup (A_0 + v_1)$, $A_2 = A_1 \cup (A_1 + v_2)$, ..., $A_s = A_{s-1} \cup (A_{s-1} + v_s)$. $v_1 ... v_s$ are chosen from U in a way that will be explained shortly. We also define $\alpha_i = 1 - \frac{|A_i|}{|U|}$, the fraction of elements left outside A_i. Note that $A_i = A_0 + span\{v_1 ... v_i\}$ for all $i = 1..s$. Our goal is to have $A_s = U$ because then we can simply choose an onto h s.t. $\ker h \supset \{v_1..v_s\}$. We show that to achieve this it is enough to have

$$\alpha_i \leq \alpha_{i-1}^2 \tag{1}$$

for $i = 1..s$. Because then, $\alpha_s \leq \alpha_0^{(2^s)}$, which gives $\alpha_s < 2^{-t}$ from our choice of the parameters. This means that $|A_s| > |U| - 2^s$. But if a single element $u \in U$ is left outside A_s, then 2^s elements $u + \text{span}\{v_1..v_s\}$ are outside A_s (Note: $v_1..v_s$ are linearly independent if (1) holds, so there are actually 2^s elements in the span). Therefore there cannot be any element outside A_s, as required. It remains to show that we can choose $v_1..v_s$ to achieve (1). The following lemma does this:

Lemma 3
Assume that $v_1...v_{i-1}$ are already chosen. Then if v_i is chosen uniformly at random from U, then

$$E(\alpha_i) = \alpha_{i-1}^2.$$

Using a probabilistic method argument, it follows that there exists a choice for v_i that achieves inequality (1). The proof of lemma 3 is by observing that if $u, v_i \in U$ are chosen uniformly at random, then the events $u \notin A_{i-1}$ and $u \notin (A_{i-1} + v)$ are independent. □

A first idea for the proof of theorem 2 is to choose $v_1...v_s$ at random, and then choose a random element $h \in \mathcal{H}$ s.t. $\forall i\ h(v_i) = 0$. Then we keep our fingers crossed and hope that with high probability we get both that $A_s = U$ and that h is *onto* W. This naive scheme seems like a promising idea, because we know from lemma 3 that α_i is good in expectation. But unfortunately this is not true : There is an adversarial choice of a set S for which the probability that α_s is small enough is not high enough (take S a linear subspace of codimension 2 in U). To overcome this problem, we split our choice of h to three parts: we construct an intermediate vector space V of dimension l such that $|V| = \mathcal{O}(|S|/\epsilon)$.

1. Choose a linear transformation $T_0 : U \to V$ uniformly at random.
2. Choose $s = l - t$ random vectors $v_1...v_s$ from V.
3. Choose an *onto* linear transformation $T_1 : V \to W$ such that $T_1(v_i) = 0\ \forall i$, uniformly at random.

Then we take $h = T_0 \circ T_1$. It can be easily verified that this induces a uniformly random choice of $h \in \mathcal{H}$. This also prevents the adversarial choice of S. The proof makes an extensive use of the probabilistic method, and its details can be found in the original article.

2 Bounds on the size of largest bucket

In this section we use the results from the previous section to bound the expected size of the largest bucket when hashing using random linear maps between vector spaces over $\mathbf{Z}_2$.

To that end, let $\mathcal{F}$ be a collection of functions mapping U to W and define the following quantity:

$$L_s^n(\mathcal{F}) = \max_{S \subseteq U, |S|=s} E_{h \in \mathcal{F}}[\max_{y \in W} |\{x \in S : h(x) = y\}|\,].$$

That is, $L_s^n(\mathcal{F})$ is the expected size of the largest bucket for an adversarially chosen input set S of size s. The theorem we will prove is the following.

Theorem 4
Let $\mathcal{H}$ be the set of all linear maps from $U = \mathbf{Z}_2^m$ to $W = \mathbf{Z}_2^{\log n}$. Then $L_{n \log n}^n(\mathcal{H}) = O(\log n \log\log n)$.

That is, when mapping an adversarially chosen set of size $n \log n$ into n buckets with a randomly chosen linear map between vector spaces over $\mathbf{Z}_2$, the expected size of the largest bucket is $O(\log n \log\log n)$.

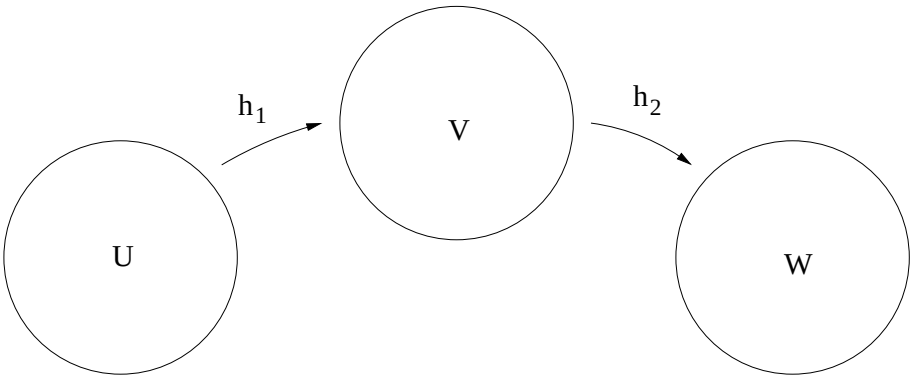

FIGURE 1: Randomly choosing $h = h_1 \circ h_2$

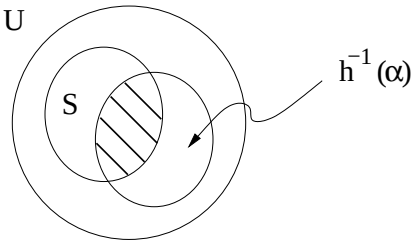

FIGURE 2: Shaded area is $h^{-1}(\alpha) \cap S$

REMARK 1 Suppose we distribute $n \log n$ balls into n buckets uniformly at random. Then the expected size of the largest bucket will be $O(\log n)$. That is, if $\mathcal{U}$ is the set of *all* maps from $U = \mathbf{Z}_2^m$ to $W = \mathbf{Z}_2^{\log n}$, then $L^n_{n \log n}(\mathcal{U}) = O(\log n)$. So hashing using a random element of $\mathcal{H}$ is nearly as good as hashing using a completely random map.

REMARK 2 Let F be a finite field with $|F| = s$. For the class $\mathcal{H}$ of all linear maps from F^2 to F, the paper shows that $L^s_s(\mathcal{H}) = \Omega(s^{1/3})$. Indeed, if $|F|$ is a perfect square, then $L^s_s(\mathcal{H}) > \sqrt{s}$.

We now prove Theorem 4. Fix a set $S \subseteq U$ of size $n \log n$ and let $V = \mathbf{Z}_2^\ell$, where $\ell > \log n$ will be chosen below. Make a random uniform choice of $h \in \mathcal{H}$ by first picking a random linear map $h_1 : U \to V$, then picking a random *onto* linear map $h_2 : V \to W$, and then letting $h = h_1 \circ h_2$ (figure 1).

We want to bound the probability of the following event (also see figure 2):

- $\mathcal{E}_1$: There exists $\alpha \in W$ such that $|h^{-1}(\alpha) \cap S| > t$, where $t = O(\log n \log \log n)$ will be chosen below.

We will bound the probability of $\mathcal{E}_1$ as follows. We will define an event $\mathcal{E}_2$ and show that whereas $\Pr[\mathcal{E}_2]$ is small, $\Pr[\mathcal{E}_2|\mathcal{E}_1]$ is large. Hence, $\Pr[\mathcal{E}_1]$ must also be small. In particular, we will prove the following two lemmas:

LEMMA 5
If $2^\ell \geq (c_\epsilon + 1) n \log n$, *then* $\Pr[\mathcal{E}_2] \leq \epsilon$.

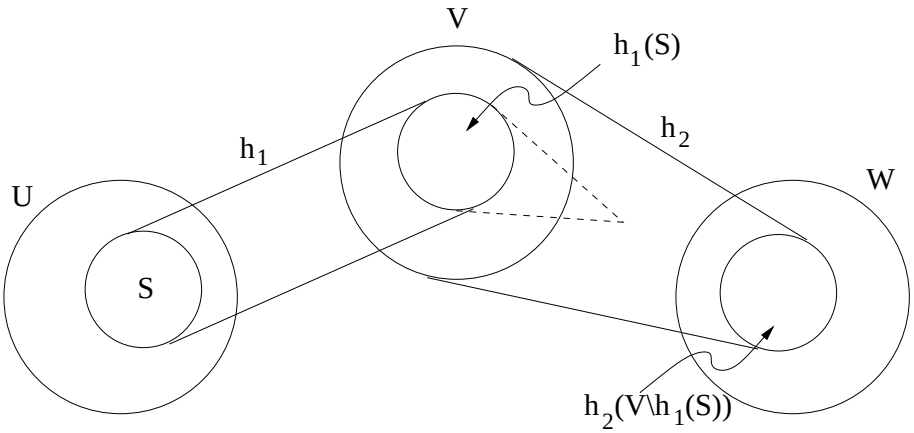

FIGURE 3: Does $h_2(V\backslash h_1(S))$ cover W?

LEMMA 6
If $t > c_{1/2}(2^\ell/n)\log(2^\ell/n)$, *then* $\Pr[\mathcal{E}_2|\mathcal{E}_1] \geq 1/2$.

With the lemmas in hand, picking

$$\ell = \lceil \log n + \log\log n + \log(c_\epsilon + 1) \rceil, \text{and}$$
$$t(n) = \lceil c_{1/2}(2^\ell/n)\log(2^\ell/n) \rceil = O(\log n \log\log n),$$

implies the existence of a constant d_ϵ depending on ϵ such that

$$\Pr[\exists \alpha \in W : |h^{-1}(\alpha) \cap U| \geq d_\epsilon \log n \log\log n] < \epsilon, \tag{2}$$

proving Theorem 4.

It remains to prove the lemmas and to define $\mathcal{E}_2$:

- $\mathcal{E}_2$: $h_2(V\backslash h_1(S)) \neq W$.

An equivalent formulation for $\mathcal{E}_2$ is:

- $\mathcal{E}_2$: There exists $\alpha \in W$ such that $h_2^{-1}(\alpha) \subseteq h_1(S)$.

PROOF:[Lemma 5] Intuitively, since Theorem 2 implies it is very unlikely for $h_2(V\backslash h_1(S))$ not to cover W, Lemma 5 then follows.

More formally, fix h_1 (if the lemma holds for a fixed h_1, it holds for a randomly chosen h_1). Note that $|h_1(S)| \leq |S| = n\log n$. Hence,

$$|V\backslash h_1(S)| \geq 2^\ell - n\log n \geq c_\epsilon n \log n.$$

Applying Theorem 2 yields the lemma (also see figure 3). □

PROOF:[Lemma 6] Assume $\mathcal{E}_1$ holds. Hence, for some $\alpha \in \mathbf{Z}_2^{\log n}$, there exists $S' \subseteq S$ such that $|S'| \geq t$ and $S' \subseteq h^{-1}(\alpha)$ (figure 4). Our goal is to show that $h_2^{-1}(\alpha) \subseteq h_1(S)$ with high probability.

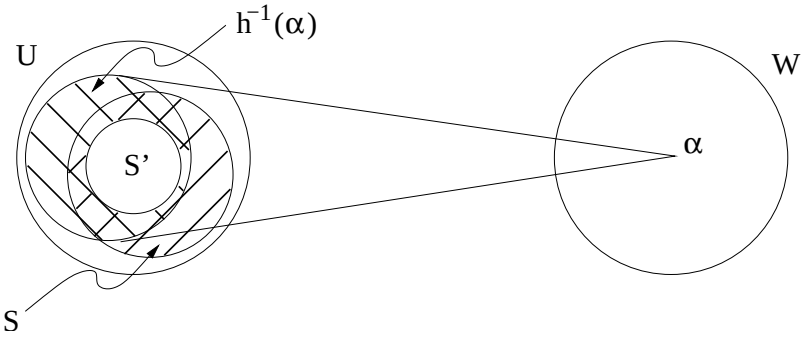

FIGURE 4: Case where α has a large inverse set under h

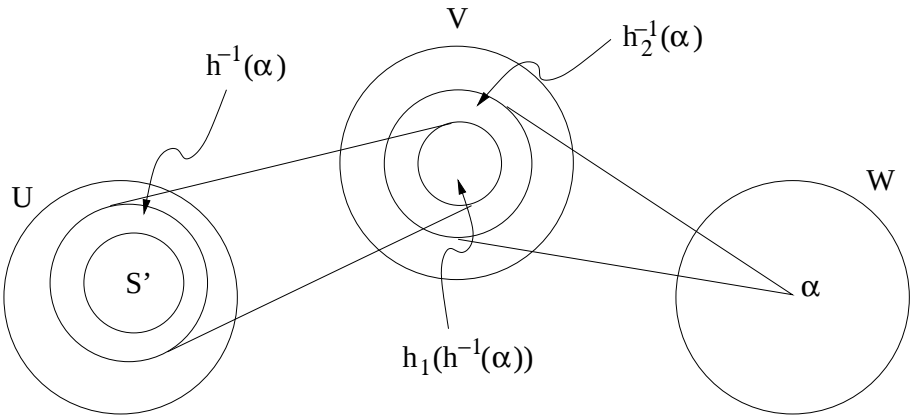

FIGURE 5: Does $h_1(S')$ cover $h_2^{-1}(\alpha)$?

Fix h_2 and consider the restricted distribution on h_1 such that $h = h_1 \circ h_2$. Since $h^{-1}(\alpha)$ and $h_2^{-1}(\alpha)$ are affine spaces, restricting h_1 to $h^{-1}(\alpha)$ then results in a uniform distribution on affine maps from $h^{-1}(\alpha)$ to $h_2^{-1}(\alpha)$. Now apply Theorem 2. In particular, how likely is it that $h_1(S')$ covers $h_2^{-1}(\alpha)$, and hence, that $\mathcal{E}_2$ holds (figure 5)? Since S' is large, this probability must be large. Formally, since h_2 is onto, it follows that $|h_2^{-1}(\alpha)| = 2^\ell/n$; but, $|S'| > t = \lceil c_{1/2}(2^\ell/n)\log(2^\ell/n)\rceil$. So Theorem 2 applies and the lemma follows. □

REMARK 3 We tacitly assumed that the restricted distribution on h_1 in the proof of Lemma 6 is uniform. This is an easy exercise. Secondly we assumed in the proof of Lemma 6 that Theorem 2 can be applied to affine maps in addition to linear maps. This also is easy to verify.

At this point we should admit that we haven't actually proved Theorem 4, but only something weaker. In particular, while we have shown that equation (2) holds, and thus have bound the probability that the largest bucket is "large", equation (2) is not strong enough to imply that the *expected* size of the biggest bucket is $O(\log n \log\log n)$. This is due to a disadvantageous dependence of d_ϵ on ϵ (this dependence was not explicitly given above—see the paper for details). Consequently, to obtain the full statement of the theorem, one needs to obtain a better dependence of d_ϵ on ϵ. This is achieved by proving a tighter

version of Lemma 5. Specifically, in the proof of Lemma 5 choose ℓ so that the set $V \backslash h_1(S)$ mapped by h_2 to W is a lot larger than required by Theorem 2. Under these conditions, the proof of Theorem 2 can be adapted (in fact simplified) to show a corresponding improvement on the probability that W is covered. Details are in the paper.

3 Open Problems

Concerning bucket size, we have proved or quoted results on the size of the largest buckets when the hash families are between vector spaces over fields of very small size ($\mathbf{Z}_2$) or very large size. What about in between? The results in the paper generalize to show that a random linear map hashing a subset of F^m of size $n = |F|^k$ to F^k will have a largest bucket with expected size of $O((\log n \log\log n)^{\log |F|})$ and $\Omega(|F|^{1/3})$. How do we close this gap?

A second open problem is to prove analogous results to those in the paper for other hash families. Candidate families for analysis include arithmetic over $\mathbf{Z}_p$, integer multiplication, and Boolean convolution.

Finally, the paper can be thought of as analysing the difference between picking a hash function from the collection of all possible functions versus picking a linear hash function. This can be viewed as analysing the effect of limiting the amount of randomness used to pick a hash function. For instance, picking a completely random hash function from $\mathbf{Z}_2^m$ to $\mathbf{Z}_2^n$ requires $n2^m$ random bits, whereas picking a random *linear* hash function from $\mathbf{Z}_2^m$ to $\mathbf{Z}_2^n$ requires only mn random bits, i.e., the bits required to define a random $m \times n$ 0-1 matrix. What happens to the performance of the hash family if we further restrict the number of random bits by only considering families of $m \times n$ 0-1 matrices described by less than mn bits?