CS 59000 CTT Current Topics in Theoretical CS	Sep 27, 2012

Lecture 12

Lecturer: Elena Grigorescu *Scribe: Jeff Gaither*

1 Introduction

In this lecture we show an application of expanders to data structures. Namely we will see how to build space-efficient data structure for the membership problem, as proposed by Buhrman et al in [1].

2 The Static Membership Problem

2.1 Setup and previous work

We considered the static membership problem, the setup of which as follows. We have a set

$$S \subset U = \{1, \ldots, m\} \quad \text{which satisfies} \quad |S| \leq n.$$

Our goal is to store S (a set of keys) in such a way that we can quickly (i.e. with few memory accesses) answer queries of the form "Is $i \in S$?". So we want to encode S in a table via some encoding scheme $S \to E(S)$. An algorithm for the membership problem takes a query "Is $i \in S$" and makes a few 'probes' into the table $E(S)$, where a probe is just an index (cell) in the table. Based on the values seen at the probed locations the algorithm should correctly answer 'yes' or 'no'.

This is a fundamental problem in data structures. Yao [3] showed that if the keys are stored explicitly (so we have n keys of size $\log m$) then the best one can do is to store the set S in sorted order on a given query perform binary search to find the answer. This means that $E(S)$ has size $n \log m$ bits, and our search could require reading $\log n$ keys, so $\log n \log m$ bits, which is somewhat inefficient.

In the so-called 'cell-probe model' introduced by Yao [3], a cell contains a number of bits and the time complexity of the scheme is counted in terms of the number of cells probed (as opposed to the number of bits seen). Fredman et al [2] conceived a scheme which improved on the sorted-storage approach, in that it required only a constant number c of probes. They stored their data in a table of size n cells (of size $\log m$ bits each) but the number of probes needed is only a constant c (so the algorithm sees $c \log m$ bits).

Another well-studied model is the 'bit-probe' model, where a cell contains only one bit. So in this model the goal is to store the set S in a small bit-string (hopefully close to information-theoretic optimal) but such that any query can be answered with only a few bit-probes.

What is the information-theoretic minimum number of bits needed to store sets S of size $\leq n$ from $U = [m]$? Since there are $\sum_{i=1}^{n} \binom{n}{k}$ distinct sets, it follows that these sets can be uniquely represented with $\Omega(n \log m)$ bits. So the goal of a storing scheme is to get as close to $n \log m$ bits of storage as possible, but also be able to efficiently answer any query. In today's lecture we'll construct a scheme where the data structure has $O(n \log m)$ bits and queries can be answered correctly with high probability by making only *one* bit-probe into the encoding.

2.2 Storing Schemes

An (n, m, t)- **storing scheme** E is simply a scheme characterized by the parameters n,m and t in the following manner: Let $\mathcal{S} = \{S \subset [m] : |S| \leq n\}$ be the set of all subsets of $[m]$ of size $\leq n$, and $E : \mathcal{S} \to \{0, 1\}^t$ is an encoding of the sets S by strings(tables) of t bits.
A **randomized** (n, m, t, ϵ) **storing scheme** is an (n, m, t) storing scheme such that $\exists$ randomized algorithm A which answers the query "Is $i \in S$?" with the following degree of accuracy:

- if $i \in S$ then A answers "yes" with probability $\geq 1 - \epsilon$, and
- if $i \notin S$ then A answers "no" with probability $\geq 1 - \epsilon$.

The bit probe is said to be **one-sided** if $\Pr(A$ says "yes"$) = 1$ whenever $i \in S$.

3 One-sided error schemes

We will show next the following result from [1]

Theorem 1 *For every $\epsilon > 0$ there is a (n, m, t, ϵ) one-sided error storing scheme for storing subsets of size n from a universe $[m]$ such that $t = O(\frac{n^2}{\epsilon^2} \log m)$, and which answers queries after only probing 1 bit.*

Observation: As shown in [1], the quadratic dependence on n in the above theorem is optimal.

If we allow a few more bit-probes, then the size of the data structure could be reduced to slightly superlinear, as described next.

Theorem 2 *For every $\delta > 0$ there is a (n, m, t, ϵ)-storing scheme (for storing subsets of size n from a universe $[m]$), of size $t = O(n^{1+\delta} \log n)$ such that each query is answered by probing $O(\frac{1}{\delta})$ bits.*

We'll prove Theorem 1 next.

The idea is to model the encoding problem as bipartite graph $G = L \cup R$. There are m vertices labelled $1, 2, ..., m$ on the left representing the elements of the universe $U = [m]$. There are t elements on the right representing the t bits of the encoding. We want to represent a set $S \subset [m]$ by its corresponding vertices on the left. We want to assign $0/1$ 'colors' (or labels) to the vertices on the right such that if $i \in S$ the color of the neighbors of i is mostly 1 and if $i \notin S$ the color of its neighbors is mostly 0. In this way, we could conceive a bit probe answering scheme where for a query i it is enough to probe a random neighbor of i and answer the query with the color of that neighbor.

So, a one-sided scheme would be to simply encode S by coloring all its neighbors with 1 and the remaining right vertices with 0.

We want to build graphs for which this encoding scheme is in fact an (n, m, t, ϵ) storing scheme, that is not only that if $i \in S$ the queries are answered correctly, but also if $i \notin S$ the error probability when answering according to the value of a random neighbor of i is at most ϵ.

Let $N(i)$ denote the set of neighbors of the vertex i. The above discussion suggests that we want to build graphs such that, for any set S and $\forall i \notin S$ we have $|N(i) \cap \cup_{u \in S} N(u)| \leq \epsilon |N(i)|$. To accomplish this, we refer to the following lemma.

Lemma 3 *(Erdos, Frankl, Furedi.) Let t, a, m be such that $m \leq \frac{\binom{t}{a}^2}{\binom{\ell}{a}}$. Then there is a family of m sets in $[t]$, each of size ℓ and such that any two have at most a elements in common. This is called an (m, ℓ, a) design.*

Proof [of Theorem 1]

We fisrt adapt Lemma 3 to our purposes and we set $a = [\log m]$, $\ell = \lceil \frac{na}{\epsilon} \rceil$ and $t = 2e^2 \frac{\ell^2}{a} = O(\frac{n^2}{\epsilon^2} \log m)$. And we set $S = \{S_1, \ldots, S_m\}$ be the sets given by the above lemma (these are therefore the sets of neighbors in $[t]$ of the m vertices on the left).

As already mentioned, we let the encoding E of a set $S = \{i_1, i_2, \ldots, i_k\}$ (where $k \leq n$) color all the vertices of the set $M = \bigcup_{j=1,k} N(i_j)$ by 1 and the remaining right vertices by 0.

Query scheme: given a query "Is $i \in S$?", we pick uniformly at random some $u \in N(i)$ and answer with the color of u. When $u \in S$, the test is always perfect since all neighbors of S were colored 1. When $u \notin S$ then we want to show that only an ϵ fraction of the neighbors of u were colored 1 and the rest were correctly colored with 0. For that we analyze the quantity $|M \cap N(u)|$. We have $|M \cap N(u)| \leq \sum_{i_j \in S} |N(i_j) \cap N(u)| \leq na$.

Then $\Pr[\text{answer is } 1] = \frac{|M \cap N(u)|}{|N(u)|} \leq \frac{na}{\ell} \leq \epsilon$.

This concludes that the above encoding and querying scheme is a valid one-sided error storing scheme that makes at most an ϵ fraction of error.

■

4 2-sided error schemes

If we allow a randomized scheme with 2-sided error we can dramatically reduce the size of the data structure to within a constant factor from optimal and still answer queries with only one bit-probe into the data structure.

We'll next show the following result of [1]

Theorem 4 *For any $\epsilon \leq 1/4$, there exists a randomized 2-side error scheme for storing subsets of size n from $[m]$ using $O(\frac{1}{\epsilon^2} n \log m)$ bits, s.t. a query '$i \in S$?' can be answered with error probability at most ϵ by making only one bit-probe to the data structure.*

The idea of the scheme is similar to the 1-sided error case. Here however we can color the neighbors of vertices $u \in S$ with 0 as well, as long as u does not have more than an ϵ fraction of its neighbors colored 0.

The goal is to find the right bipartite graphs such that the sets of neighbors of the left vertices have some nice intersection properties. We'll show that we can achieve the intersection properties that we want if we use certain expander graphs.

To formalize this discussion, if $S \subset [m]$ is a set that we'd like to encode using a 0/1 coloring of the vertices on the right, then we'd want this coloring to satisfy the following property

1. $\forall u \in S$ we should have $|N(u) \cap \{\text{vertices colored } 0\}| \leq \epsilon |N(u)|$, and
2. $\forall u \notin S$ we should have $|N(u) \cap \{\text{vertices colored } 1\}| \leq \epsilon |N(u)|$.

More abstractly, let $\mathcal{C}_1 = \{N(u)\}_{u \in S}$ and $\mathcal{C}_0 = \{N(u)\}_{u \notin S}$. We have $|\mathcal{C}_1| + |\mathcal{C}_0| = n$. An ϵ-*coloring* of the graph is a coloring of the right vertices such that

1. $\forall T \in \mathcal{C}_1$ we have $|T \cap \{\text{vertices colored } 0\}| \leq \epsilon |T|$
2. $\forall T \in \mathcal{C}_0$ we have $|T \cap \{\text{vertices colored } 1\}| \leq \epsilon |T|$.

Whenever an ϵ-coloring exists we'll say that we can ϵ-color the graph.

Definition 5 *We now recall that a graph $G = (L \cup R)$ which is d-left regular graph with $|L| = m$, $|R| = t$ is a $(2n, (1-\frac{\epsilon}{2})d)$- expander if $\forall S \subset L$ such that $|S| \leq 2n$ we have $|N(S)| \geq (1-\frac{\epsilon}{2})d|S|$. In what follows we will call such a graph and (m, t, n, d, ϵ)-expander.*

Such expanders have a very useful intersection property described in the following lemma.

Lemma 6 *Let G be a (m, t, n, d, ϵ)-expander. Then the collection of sets $\{N(u)\}_{u \in L}$ has the following property, denoted the (n, ϵ)- intersection property:*

For any fixed $k \leq n$ and any fixed k-tuple $u_1, u_2, \ldots, u_k \in L$, define the set

$$Bad = \{v \in L : |N(v) \cap \bigcup_{i \in [k]} N(u_i)| > \epsilon d\}.$$

Then $|Bad| < k$.

In other words, for any $k \leq n$ there can be at most $k-1$ vertices $v_1, v_2, \ldots, v_i \in L$ that have a large intersection with the neighbors of some given k vertices $u_1, \ldots, u_k$. We will make use of this property to color the vertices u_i with one color and the vertices v_j with the other color.

This lemma is useful because such expanders with the right parameters exists. This can be shown by the probabilistic method but we won't prove the next statement here.

Lemma 7 *For $\epsilon > 0$ $m \geq 8$ and $n \leq m/2$ there exists an $(m, \frac{200}{\epsilon^2} n \log m, n, \frac{\log m}{\epsilon}, \epsilon)$ expander.*

Note that this says that our data structure will have $t = O(n \log m)$ and that the left degree is $\frac{\log m}{\epsilon}$.

Proof [of Lemma 6] We'll prove the lemma by contradiction. Suppose $\exists k \leq n$ such that there exist $u_1, \ldots, u_k$ such that $|Bad| \geq k$.

This means that $\exists v_1, \ldots, v_k \in L$ such that $\forall j \in [k]$, $|N(v_j) \cap \bigcup_{i \in [k]} N(u_i)| > \epsilon d$.

We now let $S = \{v_1, \ldots, v_k, u_1, \ldots, u_k\}$. Then $|S| \leq 2n$.

We can now bound the size of $N(S)$, using that the graph is d-left regular. We have

$$|N(S)| \leq 2kd - \sum_{j=1}^{k} |N(v_j) \cap (\bigcup_{i=1}^{k} N(u_i))| \leq 2kd - \epsilon kd = 2k(1 - \frac{\epsilon}{2})d = (1 - \frac{\epsilon}{2})d|S|.$$

But since the graph is an $(2n, (1-\frac{\epsilon}{2})d)$ expander it means that we arrived at a contradiction and the proof is done.

■

Lemma 8 *If G is an (m, t, n, d, ϵ)-expander then we can ϵ-color the graph G.*

We remark that this lemma, together with lemma 7 are enough to prove Theorem 4. Indeed, the parameters are satisfied and the fact that we can ϵ-color the graph G means that the probing scheme where we pick a uniformly random neighbor of the query i and return its value can only fail with probability ϵ.

Now we prove lemma 8 and conclude the proof.

Proof [of lemma 8] Let S be a set of size $\leq n$ in $[m]$. Recall that $\mathcal{C}_1 = \{N(u)\}_{u \in S}$ and $\mathcal{C}_0 = \{N(u)\}_{u \notin S}$. We have $|\mathcal{C}_1| + |\mathcal{C}_0| = n$. We want to color the set $\mathcal{C}_1$ with mostly $1's$ and $\mathcal{C}_0$ with mostly 0's such that

1. $\forall T \in \mathcal{C}_1$, we have $|T \cap \{\text{vertices colored } 0\}| \leq \epsilon |T|$ and
2. $\forall T \in \mathcal{C}_0$, we have $|T \cap \{\text{vertices colored } 1\}| \leq \epsilon |T|$.

We will do this inductively.

Let $\tilde{\mathcal{C}}_0$ and $\tilde{\mathcal{C}}_1$ be the sets mostly colored 0 or 1 in our inductive steps. (We want to build an ϵ-coloring for $\mathcal{C}_0$ and $\mathcal{C}_1$ from an ϵ-coloring for $\tilde{\mathcal{C}}_0$ and $\tilde{\mathcal{C}}_1$, where $|\tilde{\mathcal{C}}_0| + |\tilde{\mathcal{C}}_1|$ is of smaller size.) We can think of the sets in $\tilde{\mathcal{C}}_0$ as being collections of neighborhoods of subsets of vertices outside S and of the sets in $\tilde{\mathcal{C}}_1$ as being collections of neighborhoods of subsets of vertices from S. We want to keep adding sets of vertices (in fact neighborhoods of the vertices on the left) to these sets in order to complete them to the sets from $\mathcal{C}_0$ and $\mathcal{C}_1$ respectively, but such that we keep track of the colors of the vertices on the right-hand side inductively.

We start with the sets $\tilde{\mathcal{C}}_0$ and $\tilde{\mathcal{C}}_1$ being empty. Then the restriction of the graph G to the vertices on the left that determine $\tilde{\mathcal{C}}_0 \cup \tilde{\mathcal{C}}_1$ and on the neighbors of these vertices is trivially ϵ-colorable (there is nothing to color). From now on, whenever the bipartite sub-graph restricted to the left/right vertices corresponding to $\tilde{\mathcal{C}}_0 \cup \tilde{\mathcal{C}}_1$ is ϵ-colorable we'll simply say that $(\tilde{\mathcal{C}}_0, \tilde{\mathcal{C}}_1)$ is ϵ-colorable.

Fix now some integer $r < n$ and let $\tilde{\mathcal{C}}_0 \subseteq \mathcal{C}_0$ and $\tilde{\mathcal{C}}_1 \subseteq \mathcal{C}_1$ be sets s.t. $|\tilde{\mathcal{C}}_0| + |\tilde{\mathcal{C}}_1| = r$. The induction hypothesis is that for any sets $\tilde{\mathcal{C}}'_0 \subseteq \tilde{\mathcal{C}}_0$ and $\tilde{\mathcal{C}}'_1 \subseteq \tilde{\mathcal{C}}_1$ if $|\tilde{\mathcal{C}}'_0| + |\tilde{\mathcal{C}}'_1| < r$ then $(\tilde{\mathcal{C}}'_0, \tilde{\mathcal{C}}'_1)$ is ϵ-colorable.

Assume that $|\tilde{\mathcal{C}}_1| \leq |\tilde{\mathcal{C}}_0|$ (otherwise interchange the role of 0 and 1 in what follows).

Let $\tilde{\mathcal{C}}'_0 = \{T \in \tilde{\mathcal{C}}_0 : |T \cap (\bigcup_{T' \in \tilde{\mathcal{C}}_1} T')| > \epsilon d\}$ (recall that we think of a set T as a set of neighbors of some vertex on the left). Notice that this corresponds to the set of vertices whose neighbors should be mostly colored 0 but which have more than an ϵ fraction of neighbors intersecting the sets that should be mostly colored 1. This is also exactly the same as a Bad set defined in lemma 6. By that lemma it follows that $|\tilde{\mathcal{C}}'_0| < |\tilde{\mathcal{C}}_1| \leq n$. So $|\tilde{\mathcal{C}}'_0| + |\tilde{\mathcal{C}}_1| < r$ and we can now apply the induction hypothesis. That is we know that there is an ϵ-coloring of $(\tilde{\mathcal{C}}'_0, \tilde{\mathcal{C}}_1)$. More explicitly, there exists a coloring of the vertices on the right s.t.

1. $\forall T \in \tilde{\mathcal{C}}_1$ we have $|T \cap \{\text{vertices colored } 0\}| \leq \epsilon |T|$
2. $\forall T \in \tilde{\mathcal{C}}'_0$ we have $|T \cap \{\text{vertices colored } 1\}| \leq \epsilon |T|$

Claim 9 *We may assume $\{$vertices colored 1$\} \subset \bigcup_{T \in \tilde{\mathcal{C}}_1} T$.*

Proof If v is colored 1 and $v \notin \bigcup_{T \in \tilde{\mathcal{C}}_1} T$ then recolor v with 0. Clearly the equation in Item 1 above still holds since $v \notin T$ when $T \in \tilde{\mathcal{C}}_1$. Also the equation in Item 2 could only get stronger since now fewer vertices are colored 1. ■

Claim 10 *The ϵ-coloring of $(\tilde{\mathcal{C}}_0', \tilde{\mathcal{C}}_1)$ implies an ϵ-coloring of $(\tilde{\mathcal{C}}_0, \tilde{\mathcal{C}}_1)$.*

Proof If $T \in \tilde{\mathcal{C}}_1$ then T is well-colored since it satisfies Item 1 above.

If $T \in \tilde{\mathcal{C}}_0'$ then T is well-colored since it satisfies Item 2 above.

If $T \in \tilde{\mathcal{C}}_0 - \tilde{\mathcal{C}}_0'$ then we can color with 0 any vertex in T that has not yet been colored and we have

$$|T \cap \{ \text{ vertices colored } 1\}| \leq |T \cap \bigcup_{T' \in \tilde{\mathcal{C}}_1} T'| \leq \epsilon |T|,$$

where the first inequality follows from Claim 9 and the last inequality follows from the definition of $\tilde{\mathcal{C}}_0'$.

■

The claim above completes the proof that $(\tilde{\mathcal{C}}_0, \tilde{\mathcal{C}}_1)$ is ϵ-colorable, and so that $(\mathcal{C}_0, \mathcal{C}_1)$ is ϵ-colorable. ■

References

[1] Harry Buhrman, Peter Bro Miltersen, Jaikumar Radhakrishnan, and Srinivasan Venkatesh. Are bitvectors optimal? *SIAM J. Comput.*, 31(6):1723–1744, 2002.

[2] Michael L. Fredman, János Komlós, and Endre Szemerédi. Storing a sparse table with 0(1) worst case access time. *J. ACM*, 31(3):538–544, 1984.

[3] Andrew Chi-Chih Yao. Should tables be sorted? *J. ACM*, 28(3):615–628, 1981.