Efficient Data Structures from Union-Free Families of Sets

Ronald de Wolf*

March 12, 2012

1 r-Union-free families of sets

We generalize the definition of an r-union-free family $\mathcal{F}$ given in the book [Juk11, Section 8.6] to the case where no set in $\mathcal{F}$ intersects much with the union of r other sets from $\mathcal{F}$:

Definition 1. *Let $\mathcal{F}$ be a family of sets over the universe $[n]$, $r \geq 1$ an integer, and $\varepsilon \in (0, 1]$. The family is called (r, ε)-*union-free *if for all distinct $A_0, A_1, \ldots, A_r \in \mathcal{F}$ we have*

$$|A_0 \cap (\cup_{i=1}^{r} A_i)| < \varepsilon |A_0|. \tag{1}$$

*The family is called r-*union-free *if it is $(r, 1)$-union-free (such families are also often called r-*cover-free*).*

Note that a 1-union-free family is just an antichain, due to the strict inequality in Eq. (1).

How big can $\mathcal{F}$ be, as a function of r, n, and ε? For the case of r-union-free families (so where $\varepsilon = 1$), [Juk11, Theorem 8.13] proves an upper bound of $|\mathcal{F}| \leq 2^{O(n \log(r)/r^2)}$. Surprisingly, this upper bound is almost achievable, even if we set ε to some constant less than 1: in Section 3 we give an existence proof of an (r, ε)-union-free family of size $|\mathcal{F}| \geq 2^{\Omega(n\varepsilon^2/r^2)}$.

2 Efficiently storing sparse sets

Consider the following data structure problem. We are given a set S which is a subset of some universe $[U]$, and we would like to store S in a way that is both space-efficient, and that allows us to efficiently answer "membership queries", i.e., decide if a given $j \in [U]$ is an element of S or not. One solution is just to store the characteristic vector of S using U bits. So our encoding of S would be some string $E(S) \in \{0, 1\}^U$. In this case, we can answer a membership query perfectly just by looking at the jth bit of $E(S)$ (looking at a bit of the data structure is called a "bitprobe"). In general, if we don't know anything more about S, then this is the best we can do.

However, suppose we know that S is "sparse", i.e., its size $|S|$ is at most some r that is much smaller than the universe size U. In this case, using U bits to store it would be wasteful: we could just write down its elements in $r \log U \ll U$ bits, which is essentially optimal.[1] Unfortunately with such an encoding it's not clear that we can still decide membership in S efficiently, with only one bitprobe. Using an (r, ε)-union-free family one can construct an encoding that takes somewhat more space ($r^2 \log U$ instead of $r \log U$

*CWI and university of Amsterdam, rdewolf@cwi.nl

[1] Since we need at least $\binom{U}{r}$ different codewords, the length of the codewords has to be at least $\log \binom{U}{r} \geq r \log(U/r)$ bits.

bits), and that allows us to answer membership queries with success probability $1-\varepsilon$ using only one bitprobe [BMRV02].

So fix some allowed error probability ε and positive integer r, and take an (r,ε)-union-free family $|\mathcal{F}|=\{A_1,\ldots,A_U\}$ over a universe $[n]$. By the result of Section 3, we can take $n=O(r^2\log U)$.[2] Here's the data structure that we use: each $S\subseteq[U]$ is encoded as an n-bit string $E(S)$ as follows

Encoding: Let $E(S)\in\{0,1\}^n$ be the characteristic vector of the set $\cup_{i\in S}A_i$

Here's how we can answer a membership query about a given element $j\in[U]$ with 1 bitprobe:

Query-answering: Pick a uniformly random $k\in A_j$, and read and output the kth bit of $E(S)$.

Let's see how well this performs. First, if $j\in S$ then $A_j\subseteq\cup_{i\in S}A_i$ so all A_j-bits in $E(S)$ are set to 1. Hence no matter which position $k\in A_j$ the algorithm probes, it will always output the correct answer in this case. Second, if $j\notin S$ then $E(S)$ is the characteristic vector of a set $\cup_{i\in S}A_i$ that has little intersection with A_j: by the (r,ε)-union-free property, only an ε-fraction of the $k\in A_j$ will lie in $\cup_{i\in S}A_i$. Hence the probability (over the choice of k) that $E(S)_k=1$ is at most ε. Accordingly, the algorithm will give the correct answer 0 with probability at least $1-\varepsilon$.

We have constructed a data structure of length $n=O(r^2\log U)$ bits that allows us to store r-subsets of the universe $[U]$ in such a way that we can answer membership queries using only one bitprobe. Note that the general upper bound $|\mathcal{F}|\leq 2^{O(n\log(r)/r^2)}$ mentioned above is equivalent $n=\Omega(\frac{r^2\log U}{\log r})$. Hence this construction cannot be improved much just by plugging in a better $\mathcal{F}$.

The length of our data structure $n=O(r^2\log U)$ is still a factor r larger than the information-theoretically minimal length $O(r\log U)$. It is in fact possible to give a 1-bitprobe data structure with this minimal length [BMRV02], but now there will be an ε error probability in both cases (also if $j\in S$). That construction is based on expander graphs, and we won't explain it here.

3 Good (r,ε)-union-free families exist

Error parameter $\varepsilon>0$, integer r, and family-size U are given. We use the probabilistic method to prove the existence of an (r,ε)-union-free family $\mathcal{F}$ of U distinct sets over a universe of size $n=O(\frac{r^2\log U}{\varepsilon^2})$.

Consider an integer a, whose value will be chosen later. Set $n=2ar/\varepsilon$, rounded up to an integer. Let A be a random variable obtained by uniformly choosing a elements from $[n]$ (with repetition, so $|A|$ is at most a). Choose $|\mathcal{F}|=\{A_1,\ldots,A_U\}$ by choosing U independent copies of A. Fix distinct indices $i_0,i_1,\ldots,i_r\in[U]$. The "bad event" for this sequence of indices is

$$(*)\ \ |A_{i_0}\cap(\cup_{j=1}^r A_{i_j})|\geq\varepsilon|A_{i_0}|$$

The set $B=\cup_{j=1}^r A_{i_j}$ has at most ar elements, hence the probability that a random element of $[n]$ lands in B is at most $ar/n=\varepsilon/2$. The set A_{i_0} consists of a such random elements, so we expect the overlap between A_{i_0} and B to be at most $a\varepsilon/2$. The bad event $(*)$ is that this overlap is at least twice as large as its expectation, hence by a Chernoff bound the probability of $(*)$ is $<2^{-c\varepsilon a}$ for some constant $c>0$.[3] Choosing a the first integer greater than $\frac{\log\binom{U}{r+1}}{c\varepsilon}$ makes the probability of $(*)$ smaller than $1/\binom{U}{r+1}$.

[2]The dependence on the fixed ε disappears in the $O(\cdot)$ notation.

[3]You can get $c=1/6\ln(2)$ by using the last bound on [Juk11, page 276] with $\mu=a\varepsilon/2$ and $\delta=1$.

Since there are $\binom{U}{r+1}$ different such sequences of indices, the union bound now implies that with positive probability *none* of the $\binom{U}{r+1}$ bad events happens, and hence there exists a choice of $\mathcal{F}$ which is (r, ε)-union-free. Note that avoiding all bad events also implies that all A_i are distinct, so $\mathcal{F}$ will have U distinct elements. The size of the required universe is $n = 2ar/\varepsilon = O(\frac{r^2 \log U}{\varepsilon^2})$. Equivalently, as a lower bound on $|\mathcal{F}| = U$ this can be written as $U \geq 2^{\Omega(n\varepsilon^2/r^2)}$.

References

[BMRV02] H. Buhrman, P. B. Miltersen, J. Radhakrishnan, and S. Venkatesh. Are bitvectors optimal? *SIAM Journal on Computing*, 31(6):1723–1744, 2002. Earlier version in STOC'00.

[Juk11] S. Jukna. *Extremal Combinatorics, with Applications in Computer Science*. EATCS Series. Springer, second edition, 2011.