03684155: On the P vs. BPP problem. 6/11/2016 – Lecture 2

List-decoding

Amnon Ta-Shma and Dean Doron

1 List Decoding

Definition 1. *We say that an $[n,k]_q$ code C is (τ, L)-list-decodable if for every word $w \in \mathbb{F}_q^n$, at most L codewords from C agree with w in at least τn coordinates.*

If we fix a codeword c and take a random word w, we expect $1/q$ agreement between c and w, and furthermore, with high probability there is such a $1/q$ agreement. Therefore, if we pick a random w, we expect there are many codewords with $1/q$ agreement with it. Specifically, there exists a word w with that many codewords having $1/q$ agreement with it. Thus, L must be large when $\tau = \frac{1}{q}$. We do want, however, codes that have a small L when $\tau \gg \frac{1}{q}$.

We state (without a proof) the Johnson's bound that shows that good distance implies decent list-decoding capabilities.

Theorem 2. *If $\tau \geq \sqrt{1-\delta}$ then any $[n,k,d=\delta n]_q$ code is (τ, qnd) list-decodable.*

The Johnoson's bound says that *every* code with a good distance has good list decoding up to some point. It is possible (and also true) that some codes outperform the Johnson's bound. For example, random codes have, w.h.p., much better list decoding. There are also explicit codes that do much better than the Johnson's bound.

We say an algorithm solves the list-decoding problem if given a word w and τ it outputs all the codewords in C that have τ-agreement with w.

In fact, one can say something tighter, we state it (without proof, but we will get back to it in the exercises) only for the binary case, and we refer to the book for the general case and the proofs.

Lemma 3. *(The Johnson's bound for binary codes) Let C be a $(n,k,d)_2$ code. I.e., C is a binary code of length n with 2^k codewords and distance d. notice that C is not necessarily linear. Then, x*

- *C is $(\tau, 2n)$ list decodable for $\tau = \frac{1}{2}(1+\sqrt{1-2\delta})$, where as usual $\delta = \frac{d}{n}$.*
- *For every L, C is (τ', L) list decodable for $\tau' = \frac{1}{2}(1+\sqrt{1-2\delta+\frac{2\delta}{L}})$.*

We recommend the reader to check that the first item implies Theorem 2 for the binary case, and is fact slightly tighter. The second item shows that with a bit more agreement the number of possible solutions drops rapidly and becomes independent of n. The case $L=1$ is not interesting, but already $L=2$ is interesting. We can deduce from the second item (check it!):

Lemma 4. *Suppose C is a $(n,k,d=(\frac{1}{2}-\varepsilon)n)_2$ code. Then C is $(\tau' = \frac{1}{2}+2\sqrt{\varepsilon}, L=\frac{1}{\varepsilon})$ list decodable.*

For example, for the Hadamard code which is a $[n=2^k, k, \frac{n}{2}]_2$ code we have,

Corollary 5. *For every k, $\mathsf{Had} : \{0,1\}^k \to \{0,1\}^{2^k}$ is $\left(\frac{1}{2}+\varepsilon, \frac{4}{\varepsilon^2}\right)$-list-decodable.*

1.1 List decoding Reed-Solomon codes up to the Johnson's bound

We first study the list-decoding of *univariate* polynomials. The lecture today closely follows Chapter 13 of the book by Guruswami, Rudra and Sudan [1]. We cite the theorem we will prove.

Theorem 6 ([2])**.** *There exists an algorithm that given as input:*

- *Code parameters:* q, $n \le q$, deg,
- *A sequence of* n *distinct pairs* $\{(\alpha_i, y_i)\}_{i=1}^{n}$, $\alpha_i, y_i \in \mathbb{F}_q$ *(alternatively, we may think of this as a potentially "corrupted" codeword), and,*
- *An agreement parameter* $\tau > \sqrt{\frac{2deg}{n}}$,

outputs a list of all polynomials $p_1, \ldots, p_\ell$ *of degree at most deg satisfying* $|\{i \in [n] : p_j(\alpha_i) = y_i\}| \ge \tau n$*. Furthermore, the list size* ℓ *is at most* $\frac{2}{\tau}$*. The algorithm runs in time* $\mathrm{poly}(n, \log q)$*.*

For the (beautiful) proof look at [1, Chapter 13].

Notice that for RS codes the Johnson's bound shows that if the agreement is at least $\sqrt{n(n-d)} = \sqrt{n \cdot deg}$ the list decoding size L is at most $qnd = \mathrm{poly}(n, q)$. The agreement we require in Theorem 6 is $\tau n = \sqrt{2n \cdot deg}$, i.e., larger by a factor of $\sqrt{2}$. Indeed, with some effort and several beautiful ideas, Theorem 6 can be improved to match the Johnson's bound. We will not cover this in class, and the interested readers are referred to [1]. We remark that it is an open problem whether the Reed-Solomon code has good list decoding beyond the Johnson's bound. We also remark that the list size we got was smaller than what we stated in Theorem 2.

References

[1] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. *Essential Coding Theory.* 2015. Available at `http://www.cse.buffalo.edu/faculty/atri/courses/coding-theory/book`.

[2] Madhu Sudan. Decoding of reed solomon codes beyond the error-correction bound. *Journal of complexity*, 13(1):180–193, 1997.