Error Correcting Codes: Combinatorics, Algorithms and Applications **(Spring 2009)**

Lecture 25: l-wise independent sources

March 23, 2009

Lecturer: Atri Rudra *Scribe: Krishna Ramkumar*

1 Introduction

In the last lecture, we introduced and discussed about BCH codes. In today's lecture, we digress a little bit and talk about the notion of *l*-wise independent sources (these are generally called *k*-wise independent sources, but for us *k* is already taken).

Definition 1.1 (*l*-wise independent random variables)**.** $X_1, X_2, ..., X_n \in \{0,1\}$ *are l-wise independent (for* $l \geq 1$*) if* $\forall \{i_1, ..., i_l\} \subseteq [n]$ *and* $(a_1, ..., a_l) \in \{0,1\}^l$*, the probability* *Pr* $[\wedge_{j=1}^{l} X_{i_j} = a_j]$ *equals* 2^{-l} *where* $(X_{i_1}, ..., X_{i_l}) \in \{0,1\}^l$

2 *l*-wise independent sources

Definition 2.1 (*l*-wise independent sources)**.** $S \subseteq \{0,1\}^n$ *is an l-wise independent source if for a uniformly chosen random* $(X_1, ..., X_n) \in S$ *,* $X_1, ..., X_n$ *are l-wise independent random variables. In other words, each* $v \in \{0,1\}^l$ *occurs* $\frac{|S|}{2^l}$ *times. Example of an n-wise independent source is* $\{0,1\}^n$ *.*

Proposition 2.2. *An n-wise independent source is also an l-wise independent source. In other words,* $(l+1)$ *wise independence implies l-wise independence.*

3 Application of *l*-wise independence

In this section, we illustrate an application of *l*-wise independence. We discuss the MAX3ESAT (or in general the MAX*l*ESAT) problem.

Definition 3.1 (MAX3(*l*)ESAT)**.** *We are given clauses* $C_1, ..., C_m$ *such that each* C_i *where* $1 \leq i \leq m$ *has exactly* $3(l)$ *distinct literals. Example,* $C_i = X_{i_1} \vee \overline{X_{i_2}} \vee X_{i_3}$*. The goal is to find an assignment that satisfies as many clauses as possible. This problem is known to be NP-Hard.*

Since the above problem is known to be NP-Hard we resort to approximate (in other words, the best possible) solutions to the same. This motivates the following definition

Definition 3.2 (α-approx algorithm)**.** *For any α where $0 \leq \alpha \leq 1$, an algorithm that always satisfies greater than or equal to α-fraction of the maximum number of satisfiable clauses is an α-approx algorithm for MAX3ESAT (It is to be noted that 1-approx is NP-Hard for any l greater than or equal to 2).*

Question: So what is the largest α-approx that can be achieved?
A $\frac{7}{8}$-approx algorithm for MAX3ESAT. In general, $(1 - 2^{-l})$-approx algorithm for MAX*l*ESAT.

Observation: For each i, pick $X_i = 0$ with probability $\frac{1}{2}$ independently. For a fixed i (where $1 \leq i \leq m$), the probability that a clause is satisfied is given by

$Pr[C_i\ is\ satisfied] = \frac{7}{8}$ (for 3-wise independent random variables)

By linearity of expectation (where the expectation is over the choice of random variables), the expected number of satisfied clauses equals $\frac{7m}{8}$. This implies that there exists an assignment that satisfies greater than or equal to $\frac{7}{8}$ fraction of the clauses. Note that for a clause $C_i = X_{i_1} \vee \overline{X_{i_2}} \vee X_{i_3}$, the choices for $X_{i_1}, X_{i_2}, X_{i_3}$ need to be independent. For this, the solution is to pick a random assignment from an *l*-wise independent source.

In the next lecture, we will see that the dual of the $BCH_{2,logn,l+1}$ is an *l*-wise independent source. By the bounds on the dimension of these codes, this means that there exists an *l*-wise independent source of size $O(n^{\lfloor \frac{l}{2} \rfloor})$ (For example, $O(n)$ size for 3-wise independence). Further, as these codes are linear codes, each codeword can be generated in time $O(n^2)$. This implies that we have an $1 - 2^{-l}$ approximation algorithm for MAX*l*ESAT that runs in time $O(n^{2+\lfloor \frac{l}{2} \rfloor})$.