9/17/2003

# Notes for Lecture 7

In this lecture, we will study using expanders to make or improve codes. Sipser and Spielman [SS96] used expanders to create codes. Alon et al [ABNNR92] developed a technique, perfected by Guruswami and Indyk [GI01], of using expanders to improve existing codes. Spielman [S96] gave a recursive code construction using parts of both ideas.

## 7.1 Expander-Based Code

### 7.1.1 Combinatorial construction

We want to constuct a binary code $[n, k, d]_2$.

Consider a bipartite graph $G = (L, R, E)$ with $|L| = n$ vertices on the left, and some smaller number $|R| = m$ vertices on the right. Vertices on the left have constant degree $\Delta$. (Right vertices need not be regular.) Place the bits of an $n$-bit string $x$ along the left nodes. For each vertex on the right, compute the exor (sum mod 2) of the labels of its neighboring vertices on the left. Define the set $O$ of codewords to be

$$O = \{x \in \{0,1\}^n : \text{all vertices on the right are zero}\} \ .$$

Equivalently, $O$ is the set of $x$ satisfying $m$ linear equations, one for each right node. Therefore $O$ is a linear subspace. The dimension $k$ of $O$ is at least $n - m$, but could be more since some equations may be dependent. $O$ can be achieved by a linear code $C : \{0,1\}^k \to \{0,1\}^n$.

The minimum distance of $C$ is the minimum weight of a nonzero codeword. $C$ has minimum distance $d$ provided that for every subset $S \subset L$ with $|S| < d$, some vertex in $R$ has an odd number of neighbors in $S$. We will define an even stronger property:

**Definition 1** *$G = (L, R, E)$ is a $d$-unique neighbor expander if for every $S \subset L$, $|S| < d$, there is a $v \in R$ with exactly one neighbor in $S$.*

The definition is trivially satisfied by a matching in the case $|L| = |R|$. Note that this is not a typical expander definition. In particular, adding edges to $G$ can destroy this property! The standard expander definition is:

**Definition 2** *$G = (L, R, E)$ is a $(\gamma, \delta)$-expander if for all $S \subset L$ with $|S| \le \delta|L|$, the set $\Gamma(S) \subset R$ of neighbors of $S$ has $|\Gamma(S)| \ge \gamma|S|$ elements.*

**Claim 1** *If $G$ is a $(\gamma, \delta)$-expander, with $\gamma > \frac{\Delta}{2}$, where $\Delta$ is the left-degree, then $G$ is also a $\delta|L|$ unique-neighbor expander.*

PROOF: Take $S \in L$ of size $< \delta|L|$. Then $2|\Gamma(S)| \geq 2\gamma|S| > \Delta|S|$. The total number of edges leaving $S$ is $\Delta|S|$, but at least $2|\Gamma(S)|$ edges would be needed if each $v \in \Gamma(S)$ had two neighbors in $S$. So some vertex in $\Gamma(S)$ must have only one neighbor in $S$. $\square$

Hence if $G = ([n], [m], E)$ with left-degree $\Delta$ is a $(\gamma, \delta)$-expander with $\gamma > \frac{\Delta}{2}$, then we get an $[n, n-m, \delta n]_2$ code.

Such expanders can be generated at random with high probability, but there is no known algorithm for checking the unique-neighbor property. However, Alon and Capalbo [AC02] recently showed that $\forall \epsilon, \alpha$, $\exists \Delta, \delta$ for which you can explicitly construct $G = ([n], [\alpha n], E)$ with left degree $\Delta$, a $(\Delta(1-\epsilon), \delta)$-expander.

### 7.1.2 Efficient decoding

To obtain an efficient decoding algorithm (instead of just combinatorial), we will require an even stronger expander property: $\gamma > \frac{3}{4}\Delta$.

**Claim 2** *Suppose $G = ([n], [m], E)$ is a $(\gamma, \delta)$-expander with left-degree $\Delta$, $\gamma > \frac{3}{4}\Delta$. Then there is a linear-time algorithm to correct $< \frac{\delta}{2\Delta}n$ errors. (For $\Delta$ constant, the correctable distance is only a constant fraction less than the combinatorial distance.)*

PROOF: The decoding algorithm places the (possibly corrupted) codeword $y$ along $L$, and fills in the the right nodes with the exor of their neighbors in $L$. Then while there is a left vertex $v$ with strictly more unsatisfied than satisfied neighbors, it flips $v$.

Say $y = x + e$, where $x$ is a codeword and $|e| < \frac{\delta n}{2\Delta}$. Adding $x$ to the left hand side, leaving $e$, does not change the right-hand labels. So without loss of generality $x = 0^n$ and $y = e$ with $|y| < \frac{\delta n}{2\Delta}$.

Since the left degree is $\Delta$, at most $\frac{\delta n}{2}$ right vertices are unsatisfied. Since each step of the algorithm strictly decreases the number of unsatisfied vertices, the algorithm terminates after at most $\frac{\delta n}{2}$ bit flips (linear time). Since the distance is $\delta n$, the algorithm either outputs the *correct* codeword, or it stops in a local minimum with some unsatisfied vertices remaining.

We claim that the second possibility contradicts the expansion property of the graph. Indeed, let $S$ be the set of errors remaining after the algorithm finishes. $S$ has $> \frac{3}{4}\Delta|S|$ distinct neighbors. However there are at most $\frac{\Delta}{2}|S|$ unsatisfied nodes (neighbors of an odd number of vertices in $S$) – otherwise some $v$ would have more unsatisfied than satisfied neighbors and should be flipped.

Let $u$ be the number of be the number of unsatisfied right vertices with unique neighbors in $S$. Then, counting edges,

$$\Delta|S| \geq u + 2(\tfrac{3}{4}\Delta|S| - u) \ ,$$

implying $u \geq \frac{\Delta|S|}{2}$, a contradiction. (Actually, the factor of 2 in the above equation can be replaced by 3, since each unsatisfied right vertex has an odd number of neighbors in $S$. Hence $u \geq \frac{5}{8}\Delta|s|$.) We conclude that the algorithm always outputs the correct codeword. $\square$

### 7.2 Expanders to Improve Codes

Alon et al [ABNNR92] use an expander to improve an existing $[n, k, d]_2$ code $C$.

#### 7.2.1 Construction

Let $G$ be a bipartite graph with $|L| = |R| = n$, and constant right-degree $\Delta$. For a message $m \in \{0,1\}^k$, we determine the encoding $C'(m)$ as follows: label the left vertices of $G$ by $C(m) \in \{0,1\}^n$, then assign to each right vertex the labels of its $\Delta$ neighbors. Thus overall,

$$C' : \{0,1\}^k \to (\{0,1\}^\Delta)^n \ .$$

Starting from two different messages, they differ in $\geq d$ positions $S$ on the left. Their codewords differ in the positions $\Gamma(S)$. If $G$ is such that for all $S \subset L$ with $|S| \geq d$, $|\Gamma(S)| \geq d'$, then $C'$ is an $[n, \frac{k}{\Delta}, d']_{2^\Delta}$ code. Such expanders can be constructed with $d = \delta n$, $d' = (1-\epsilon)n$ and $\Delta = \text{poly}(\frac{1}{\epsilon}, \frac{1}{\delta})$.

#### 7.2.2 Decoding

Assume that $C$ has an efficient decoding algorithm correcting $\frac{d}{2}$ errors. Our decoding algorithm assigns to each left vertex the majority of the predictions (in linear time), then runs the decoding algorithm for $C$.

If a left vertex is wrong, then the majority of its neighbors have to be wrong. To correct $(\frac{1}{2} - \epsilon)n$ errors, we therefore require the following property on $G$: For every $T \subset R$ with $|T| \leq (\frac{1}{2} - \epsilon)n$, for every $S \subset L$ large enough, $|S| \geq \frac{\delta}{2}n$, not all vertices in $S$ have a majority of neighbors in $T$.

Graphs can be constructed with this property with $\Delta = \text{poly}(\frac{1}{\epsilon}, \frac{1}{\delta})$.

## References


[ABNNR92] N. Alon, J. Bruck, J. Naor, M. Naor and R. Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. IEEE Trans. on Information Theory, 38 (1992), pp. 509-516.

[AC02] Noga Alon, Michael R. Capalbo. Explicit Unique-Neighbor Expanders. FOCS 2002.

[GI01] Venkatesan Guruswami, Piotr Indyk. Expander-Based Constructions of Efficiently Decodable Codes. FOCS 2001.

[SS96] Michael Sipser, Daniel A. Spielman. Expander Codes. FOCS 1996.

[S96] Daniel A. Spielman. Linear-Time Encodable and Decodable Error-Correcting Codes. IEEE Trans. on Information Theory, 42 (1996), pp. 1723-1731.