

15-853: Algorithms in the Real World

- Error Correcting Codes III
- Expander graphs
 - Tornado codes

Thanks to Shuchi Chawla for the slides

15-853

Page1

Why Tornado Codes?

Designed by Luby, Mitzenmacher, Shokrollahi et al

Linear codes like RS & random linear codes

The other two give nearly optimal rates

But they are slow :

Code	Encoding	Decoding
Random Linear	$O(n^2)$	$O(n^3)$
RS	$O(n \log n)$	$O(n^2)$
Tornado	$O(n \log 1/\epsilon)$	$O(n \log 1/\epsilon)$

Assuming an $(n, (1-p)n, (1-\epsilon)^{pn+1})_2$ tornado code

15-853

Page2

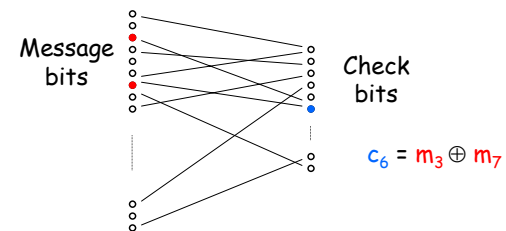
The idea behind Tornado codes

Easy coding/decoding:
linear codes with explicit construction

Fast coding/decoding:
each check bit depends on only a few message bits

15-853

Page3



Think of this as a "regular" Bipartite Graph

Each message bit is used in only a few check bits

=> Low degree bipartite graph

15-853

Page4

Properties of a good code

There should be "few" check bits

Linear time encoding

- Average degree on the left should be a small constant

Easy error detection/decoding

- Each set of message bits should influence many check bits
- Existence of unshared neighbors

15-853

Page5

Outline

Expander Graphs

- Applications
- Properties
- Constructions

Tornado Codes

- Encoding/Decoding Algorithms
- Brief Analysis

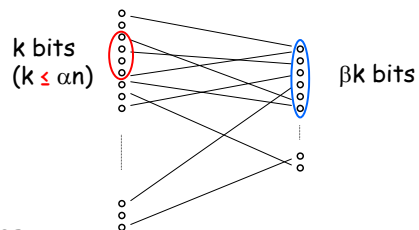
Expander Codes

- Construction
- Brief Analysis

15-853

Page6

Expander Graphs (bipartite)



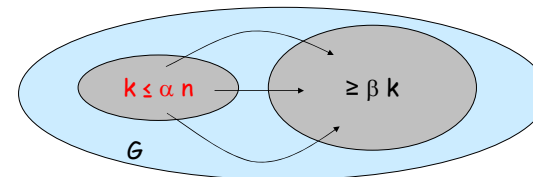
Properties

- **Expansion:** every small subset ($k \leq \alpha n$) on left has many ($\geq \beta k$) neighbors on right
- **Low degree** - not technically part of the definition, but typically assumed

15-853

Page7

Expander Graphs (non-bipartite)



Properties

- **Expansion:** every small subset ($k \leq \alpha n$) has many ($\geq \beta k$) neighbors
- **Low degree**

15-853

Page8

Expander Graphs: Applications

Pseudo-randomness: implement randomized algorithms with few random bits

Cryptography: strong one-way functions from weak ones.

Hashing: efficient n-wise independent hash functions

Random walks: quickly spreading probability as you walk through a graph

Error Correcting Codes: several constructions

Communication networks: fault tolerance, gossip-based protocols, peer-to-peer networks

15-853

Page9

d-regular graphs

An undirected graph is **d-regular** if every vertex has d neighbors.

A bipartite graph is **d-regular** if every vertex on the left has d neighbors on the left.

The constructions we will be looking at are all d -regular.

15-853

Page10

Expander Graphs: Properties

If we start at a node and wander around randomly, in a "short" while, we can reach any part of the graph with "reasonable" probability. (rapid mixing)

Expander graphs do not have small separators.

The eigenvalues of the adjacency matrix of a graph carry information about the expansion of the graph.

15-853

Page11

Expander Graphs: Eigenvalues

Consider the normalized adjacency matrix A_{ij} for an undirected graph G (all rows sum to 1)

The (x_i, λ_i) satisfying

$$A x_i = \lambda_i x_i$$

are the **eigenvectors** and **eigenvalues** of A .

Consider the eigenvalues $\lambda_0 \geq \lambda_1 \geq \lambda_2 \geq \dots$

For a d -regular graph, $\lambda_0 = 1$. Why?

The separation of the eigenvalues tell you a lot about the graph (we will revisit this several times).

If λ_1 is much smaller than λ_0 then the graph is an expander.

Expansion $\beta \geq (1/\lambda_1)^2$

15-853

Page12

Expander Graphs: Constructions

Important parameters: size (n), degree (d), expansion (β)

Randomized constructions

- A random d -regular graph is an expander with a high probability
- Construct by choosing d random perfect matchings
- Time consuming and cannot be stored compactly

Explicit constructions

- Cayley graphs, Ramanujan graphs etc
- Typical technique - start with a small expander, apply operations to increase its size

15-853

Page13

Expander Graphs: Constructions

Start with a small expander, and apply operations to make it bigger while preserving expansion

Squaring

- G^2 contains edge (u,w) if G contains edges (u,v) and (v,w) for some node v
- $A' = A^2 - 1/d I$
- $\lambda' = \lambda^2 - 1/d$
- $d' = d^2 - d$

Size	≡
Degree	↑
Expansion	↑

15-853

Page14

Expander Graphs: Constructions

Start with a small expander, and apply operations to make it bigger while preserving expansion

Tensor Product

- $G = A \times B$ nodes are $(a,b) \quad \forall a \in A \text{ and } b \in B$
- edge between (a,b) and (a',b') if A contains (a,a') and B contains (b,b')
- $n' = n_1 n_2$
- $\lambda' = \max(\lambda_1, \lambda_2)$
- $d' = d_1 d_2$

Size	↑
Degree	↑
Expansion	↓

15-853

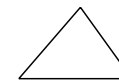
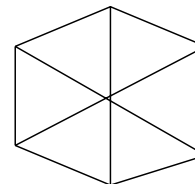
Page15

Expander Graphs: Constructions

Start with a small expander, and apply operations to make it bigger while preserving expansion

Zig-Zag product

- "Multiply" a big graph with a small graph



$$n_2 = d_1$$

$$d_2 = \sqrt{d_1}$$

15-853

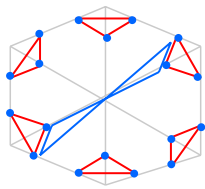
Page16

Expander Graphs: Constructions

Start with a small expander, and apply operations to make it bigger while preserving expansion

Zig-Zag product

- "Multiply" a big graph with a small graph



15-853

Page17

Outline

Expander Graphs

- Applications
- Properties
- Constructions

Tornado Codes

- Encoding/Decoding Algorithms
- Brief Analysis

Expander Codes

- Construction
- Brief Analysis

15-853

Page18

The loss model

Random Erasure Model:

- Each bit is lost independently with some probability μ
- We know the positions of the lost bits

For a **rate** of $(1-p)$ can correct $(1-\epsilon)p$ fraction of the errors.

Seems to imply a

$$(n, (1-p)n, (1-\epsilon)pn+1)_2$$

code, but not quite because of random errors assumption.

We will assume $p = .5$.

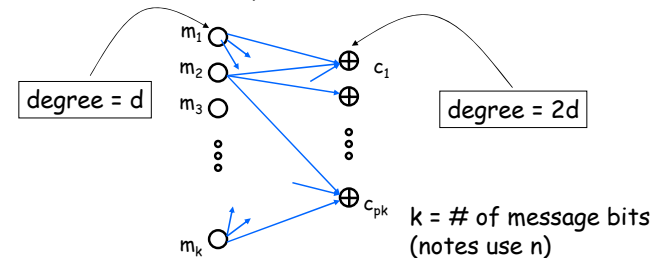
Error Correction can be done with some more effort

15-853

Page19

Tornado codes

Will use d -regular bipartite graphs with n nodes on the left and pn on the right (notes assume $p = .5$)
Will need $\beta > d/2$ expansion.

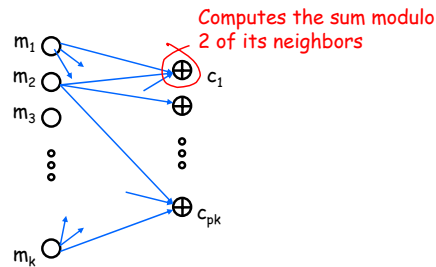


15-853

Page20

Tornado codes: Encoding

Why is it linear time?

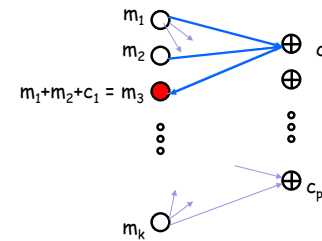


15-853

Page21

Tornado codes: Decoding

Assume that all the check bits are intact
 Find a check bit such that only one of its neighbors
 is erased (an *unshared neighbor*)
 Fix the erased code, and repeat.



15-853

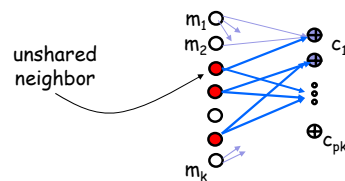
Page22

Tornado codes: Decoding

Need to ensure that we can always find such a check bit
 "Unshared neighbors" property

Consider the set of corrupted message bit and their
 neighbors. Suppose this set is small.

=> at least one message bit has an unshared neighbor.



15-853

Page23

Tornado codes: Decoding

Can we always find unshared neighbors?

Expander graphs give us this property if $\beta > d/2$
 (see notes)

Also, [Luby et al] show that if we construct the
 graph from a specific kind of degree sequence,
 then we can always find unshared neighbors.

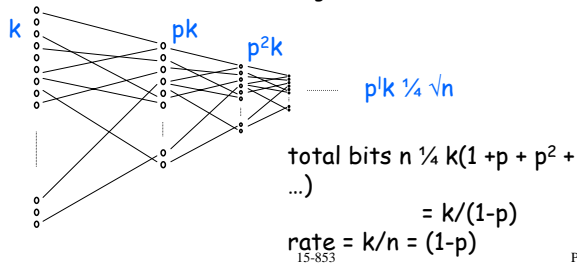
15-853

Page24

What if check bits are lost?

Cascading

- Use another bipartite graph to construct another level of check bits for the check bits
- Final level is encoded using RS or some other code



15-853

Page25

Cascading

Encoding time

- for the first k stages : $|E| = d \times |V| = O(k)$
- for the last stage: $\sqrt{k} \times \sqrt{k} = O(k)$

Decoding time

- start from the last stage and move left
- again proportional to $|E|$
- also proportional to d , which must be at least $1/\epsilon$ to make the decoding work

Can fix $kp(1-\epsilon)$ random erasures

15-853

Page26

Outline

Expander Graphs

- Applications
- Properties
- Constructions

Tornado Codes

- Encoding/Decoding Algorithms
- Brief Analysis

Expander Codes

- Construction
- Brief Analysis

15-853

Page27

Expander Codes

Input:

Regular expander G on n nodes, degree d
 Code C of block length d , rate r , rel. distance δ

Output:

Code $C(G,C)$ of block length $dn/2$, rate $2r-1$,
 rel. distance $\frac{1}{4} \delta^2$

Linear time encoding/decoding

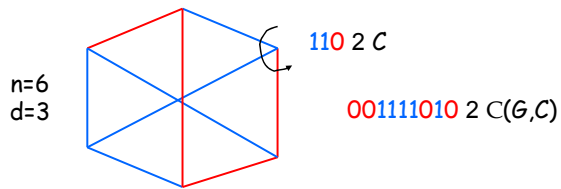
15-853

Page28

Expander Codes: Construction

We associate each edge in G with a bit of the code
 For every vertex, the edges around it form a code word in C

Block length = number of edges = $nd/2$



15-853

Page29

Expander Codes: Construction

Linear code C has rate r

\Rightarrow there are $(1-r)d$ linear constraints on its bits
 (these constraints define a linear subspace of dimension rd)

Total number of constraints in the entire graph G
 $= (1-r)nd$

Total length of code = $nd/2$

\Rightarrow Total number of message bits = $nd(r-1/2)$

Therefore, rate is $2(r-1/2) = 2r-1$

15-853

Page30

Expander Codes: Construction

For linear codes, the minimum distance between two code words = minimum weight of a code word

Intuition:

If the weight of a code word is small, then the weight of edges around some vertex is small

\Rightarrow distance of C is small \Rightarrow contradiction

15-853

Page31

Expander Graphs: Construction

Expander graphs:

Any set of αn nodes must have at most

$$m = (\alpha^2 + (\alpha - \alpha^2) \lambda/d) dn/2 \text{ edges}$$

So, a group of m edges must touch at least αn vertices

One of these vertices touches at most $m/2\alpha n$ edges

But these should be at least δd for the code to be valid

$$\text{So, } (\alpha + (1-\alpha) \lambda/d) d > \delta d$$

$$\Rightarrow \alpha > (\delta - \lambda/d)/(1-\lambda/d)$$

Minimum distance is at least $\alpha (\alpha + (1-\alpha) \lambda/d) \frac{1}{4} \delta^2$

15-853

Page32

Some extra slides

15-853

Page33

Expander Graphs: Properties

Prob. Dist. - π ; Uniform dist. - u

Small $|\pi - u|$ indicates a large amount of "randomness"

Show that $|A\pi - u| \cdot \lambda_2 |\pi - u|$

Therefore small $\lambda_2 \Rightarrow$ fast convergence to uniform

Expansion $\beta \frac{1}{4} (1/\lambda_2)^2$

15-853

Page34

Expander Graphs: Properties

To show that $|A\pi - u| \cdot \lambda_2 |\pi - u|$

Let $\pi = u + \pi'$

u is the principle eigenvector

$$Au = u$$

π' is perpendicular to u

$$A\pi' \cdot \lambda_2 \pi'$$

So, $A\pi = u + \lambda_2 \pi'$

Thus, $|A\pi - u| \cdot \lambda_2 |\pi'|$

15-853

Page35