DD2442 AUTUMN 2014 LECTURE 14

CODING THEORY – TIME TO WRAP UP

Last time: Algorithmic focus

Efficient encoding: Easy for linear code given generator matrix
Ignored this

Efficient decoding: Find unique nearest codeword for received word at distance $< \Delta(C)/2$ from code

(Briefly discussed other problems such as list decoding: find all codewords within given radius)

Welch–Berlekamp algorithm for unambiguous decoding of Reed-Solomon codes

Time cplx $O(n^3)$ [can do $O(n \operatorname{polylog} n)$]

Main idea: Wishful thinking

Error-locating polynomial $E(x)$

$E(x_i) = 0$ if received $y_i \neq p(x_i)$

for message $p(x)$

$N(x) := E(x)\,p(x)$

Solve for $N(x)$ and $E(x)$ – linear system of equations

Output $N(x) / E(x)$.

Reed-Solomon codes have (very) large alphabets.

Want binary codes that are <u>asymptotically good</u>

Positive rate $\lim \left(\frac{\text{message length}}{\text{block length}} \right) > 0$

Positive relative distance $\lim \left(\frac{\text{distance}}{\text{block length}} \right) > 0$

and that can be encoded and decoded in <u>linear time</u>

Look at parity check matrix $H \in \mathbb{F}_2^{n \times (n-k)}$

$$C = \{ x \mid xH = 0 \}$$

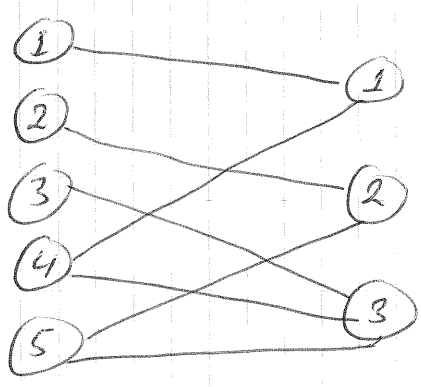
View H as bipartite graph

Rows = positions in code words
= left-hand side vertices

Columns = parity constraints
= right-hand side vertices

H

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$



Can also go in other direction

Given $G = (U \cup V, E)$

$|U| = n$, $|V| = m$

Let n = block length

Let H be parity check matrix with

$$h_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

(H is just the bipartite adjacency matrix)

Code with rate $n - m$ (if H full-rank)

We focus on <u>sparse graphs</u> G

$O(n)$ edges

Codes with corresponding parity check matrices are

LOW DENSITY PARITY CHECK CODES

(LDPC codes)

Concluded last lecture with claim/hope that if G really good bipartite expander then resulting code $C(G)$

- is asymptotically good
- can be encoded and decoded in linear time

This is what we will see today. But first a brief historic overview:

Idea of constructing codes from graphs in this way by [Gallager '63]
Used sparse bipartite random graphs

Picked up in Russia by [Pinsker and Bassalygo '73]
Realized expansion was at the heart of the arguments
Formally defined expander graphs
Observed such graphs useful also in other application, e.g., robust communication networks

[Pinsker '73]: Most sparse graphs are expanders
[Margulis '73]: First explicit expander construction

Then coding theory switched focus to algebraic techniques for a while

Revival in works of
[Tanner '81]
[Alon, Bruck, Naor, Naor, Roth '92]
[Sipser, Spielman '96]
[Spielman '96]
culminating in asymptotically good codes with linear time encoding and decoding

And further work constructing nearly linear-time-decodable codes with rates approaching channel capacity for variety of models of noisy communication channels (not just the simple model we have studied).

Will now see such a family of codes and prove that these codes have fantastic properties

From now on
$G = (V_L \dot\cup V_R, E)$ bipartite graph
k-regular on the left (reserve d for distance)
$|V_L| = n$, $|V_R| = m$

DEF 1 The LEFT VERTEX EXPANSION RATIO $\alpha(G, d)$ is $\min_{0 < |S| \le d, S \subseteq V_L} \frac{|N(S)|}{|S|}$

Note that $\alpha(G, d) \le k$
But it follows from [Capalbo, Reingold, Vadhan, Wigderson '02] as mentioned last time that there are explicit bipartite graphs for any $m = \Omega(n)$ with expansion ratio $(1-\varepsilon)k$ for $d = \Omega(n)$.
(but $m < n$)
Such graphs G yield linear codes $C(G)$ with constant rate

We now show:
- Relative distance is constant
- Decoding can be done in linear time

Encoding in time $O(n^2)$ follows by linearity but can be done in time $O(n)$ with a bit of care — won't discuss this

Following two theorems by Sipser and Spielman (proven in 1996) before explicit constructions of lossless expanders were known

DISTANCE

THEM 2 [Code $C(G)$ has large distance]
If $\alpha(G, d) > k/2$, then $\Delta(C(G)) > d$

Proof: G is k-regular with expansion ratio $> k/2$
$\Rightarrow$ every $S \subseteq V_L$, $|S| \leq d$ has a unique neighbour,
i.e., vertex $v \in V_R$ s.t. $|N(v) \cap S| = 1$

This is so since

$$|E(S, N(S))| = k|S| \quad \text{[by regularity]}$$
$$|N(S)| > k|S|/2 \quad \text{[by expansion]}$$

Hence average right degree in $N(S)$ < 2
i.e., $\exists\, v \in N(S)$ with exactly one neighbour in S (wrt edges $E(S, N(S))$)

Recall: min distance of linear code = min weight of codeword

Let $x \in C(G)$ and suppose $wt(x) \leq d$
Let $S \subseteq V_L$ support of x, i.e., $S = \{ i \in [n] \mid x_i = 1 \}$
[Note $x_j = 0$ for $j \in [n] \setminus S$]

Let $v \in V_R$ be unique neighbour of S.

Then v corresponds to parity check

$$x_i + x_{j_1} + \dots + x_{j_\ell} = 0$$

for $i \in S$, $j_s \notin S$ $\forall s$
i.e. $x_i = 1$, $x_{j_1} = \dots = x_{j_\ell} = 0$

But then constraint v is not satisfied,
so x is not a codeword in $C(G)$ $\square$

EFFICIENT DECODING

Given received word $y = (y_1, \ldots, y_n)$,
how to decode?
And what do we mean by decoding?
We will, strictly speaking, show that our code $C(G)$ is an ERROR-REDUCTION CODE,
i.e., given y of distance $< \Delta(C(G))/2$,
can find unique closest $x \in C(G)$.
Given such a code, can modify
it to get true decoding of message
also in linear time.
(important)
From now on, ignore this subtlety
and focus on recovering correct codeword.

Ok, so how to "decode" $y = (y_1, \ldots, y_n)$ at
distance $< \Delta(C(G))/2$ to $x = (x_1, \ldots, x_n) \in C(G)$?

If all constraints on the right satisfied
– done.
Otherwise, each coordinate y_i can look
at its constraints on the right and
see if they are satisfied.

Naive idea:
If a majority of constraints involving y_i are
unhappy, then maybe y_i is wrong? Flip it!

That is, given $y \notin C(G)$, if
$wt((y+e_j)H) < wt(yH)$ for H parity check matrix
then set $y_j := y_j + e_j$.

Naive decoding algorithm

Set $T := 0$ $y' := y$ (for K sufficiently large constant)
while $T \leq Kn$ and $y'H \neq 0$
{
  Find j s.t. $wt((y'+e_j)H) < wt(y'H)$
} $y' := y' + e_j$; $T := T + 1$;
if $y'H = 0$ and $\Delta(y', y) \leq \Delta(C(G))/2$
  output y'
else
  output FAIL

Two challenges: (i.e. constant time per iteration)

(i) Implement in linear time — can be done
we will ignore this (or maybe leave
as exercise)

(ii) Analyze why algorithm terminates with
correct answer for received word y
at distance at most $\Delta(C(G))/2$
— this we will do

Sidenote: This kind of algorithm often called BELIEF PROPAGATION. Powerful theme in error correction, artificial intelligence, computational learning theory...

Most often more of a heuristic.
Here this strategy actually provably works!

THM 3
Let G be k-left-regular bipartite graph with left vertex expansion ratio $\alpha(G,d) > \frac{3}{4}k$.
Let $y \in \{0,1\}^n$ be string at distance $\leq d/2$ from $C(G)$. Then "Naive decoding algorithm" returns $x \in C(G)$ s.t $\Delta(x,y) \leq d/2$ (and in particular does so in linear time)

Proof Start by setting up notation

$y^{(i)}$ = vector after i iterations
$y^{(0)} = y$ = received word

$A^{(i)}$ set of errors after iteration i

$$A^{(i)} = \{ j \mid y_j^{(i)} \neq x_j \}$$

Need to prove $A^{(K+1)} = \emptyset$ for suitably chosen K constant.

Look at what happens iteration $i+1$
Let $A = A^{(i)}$ (to avoid clutter)
Assume $A \neq \emptyset$ (else we're done)
Assume also $|A| \leq d$ (return to this later)
Look at constraints involving positions in A
Partition into ← "fooled" constraints

$S^{(i)} = S$: satisfied constraints in $N(A)$
$U^{(i)} = U$: unsatisfied constraints in $N(A)$

(i.e., $U^{(i)}$ is support of $y^{(i)} H$)

X

By expansion

$$|S| + |U| = |N(A)| > \frac{3}{4} k |A| \qquad (1)$$

Look at edges. At least one edge between every $u \in U$ and A. At least two edges for every $s \in S$ (since constraint mistakenly satisfied). At least $|U| + 2|S|$ from A ending $N(A) = S \cup U$, so

$$|U| + 2|S| \leq k|A| \qquad (2)$$

Take $2 \cdot (1) - (2)$ to get

$$|U| > \frac{1}{2} k |A| \qquad (3)$$

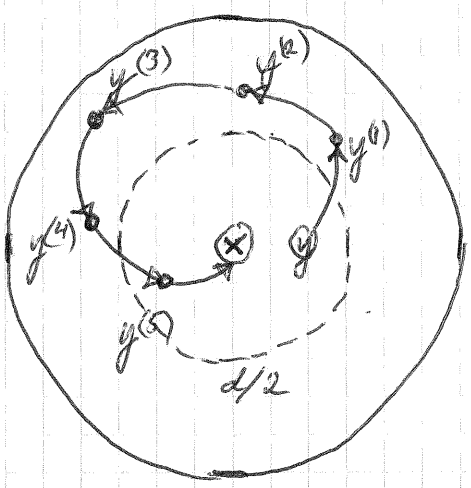
By (3) there is some position $j \in A$ with $> k/2$ constraints unhappy, so we find a coordinate to flip.

Hence:

- As long as $|A^{(i)}| \leq d$, will find coordinate to flip.
- Also, since we flip position with strict majority of violated constraints, $|U^{(i)}|$ decreases in every iteration
- So if $|A^{(i)}| \leq d$ always holds, then algorithm terminates in linear # steps with output x

Remains to prove that $|A^{(i)}| \le d$,
given assumption $|A^{(0)}| \le d/2$

Schematically



WANT TO SHOW:

Starting inside $B(x, d/2)$,
might go out into $B(x, d)$
but will stay in this larger ball
and eventually more back into
$B(x, d/2)$ and hit x.

$|A^{(i)}|$ changes by ± 1 (flip 1 position in each step)

If we exit $B(x, d)$, there is (i^* a first time step)
s.t. $|A^{(i)}| = d$. By (3) we get

$$|U^{(i^*)}| > \frac{1}{2} k |A^{(i^*)}| = kd/2 \qquad (4)$$

On the other hand, in the beginning
we have $|A^{(0)}| \le d/2$ by assumption

and therefore { So K_n iterations for $K = \frac{kd}{2n}$ sufficient }

$$|U^{(0)}| \leq |N(A^{(0)})| \leq kd/2 \qquad (5)$$

But for $i = 0, 1, \ldots, i^*+1$ we have $|A^{(i)}| \leq d$, and as long as that holds we proved that $|U^{(i)}|$ decreased monotonically. Applying this to (4) and (5), there is no way we can have

$$|U^{(i^*)}| > |U^{(0)}|$$

and hence $|A^{(i)}| \leq d$ holds throughout whole algorithm. ▨

There is also a version with "parallel belief propagation" — on every iteration, flip <u>all</u> positions with a majority of violated constraints

Can be shown that this converges in $O(\log n)$ iterations and total work (= total # flips) $O(n)$

Analysis uses lossless expansion to show that $|A^{(i)}|$ shrinks by constant factor for each iteration.

Summing up

Have discussed error-correcting codes $(n, k, d)_q$

Block length n — minimize
Message length k — maximize
Distance d — maximize
Alphabet size q — minimize

Given code C with distance d
- Can DETECT $d-1$ errors
- Can CORRECT $\lfloor \frac{d-1}{2} \rfloor$ errors

We saw a few different code constructions using algebraic techniques (and resulting in LINEAR CODES)

Also saw some upper bounds on what is possible (Hamming bound, Singleton bound)

REED-SOLOMON CODES
Codeword = evaluations of low-degree univariate polynomial
Optimal relation between message length and distance
But alphabet large

REED-MULLER CODES

Codeword = evaluations of low-degree multivariate polynomial

Smaller alphabet size

Still good distance

Schwartz-Zippel: Degree-d $f \in \mathbb{F}_q[x_1, \ldots, x_m]$ non-zero on at most d/q fraction of $\mathbb{F}_q^m$

HADAMARD CODE

Can view encoding of $x \in \{0,1\}^k$ as evaluations of all linear functions on x, i.e., $(\ell(x) \mid \ell: \{0,1\}^k \to \{0,1\}$ linear$)$

Family of codes $\mathcal{C} = \{C_i\}_{i \in \mathbb{N}}$

$C_i = (n_i, k_i, d_i)_{q_i}$

$\lim_{i \to \infty} n_i = \infty$

Message rate $R(\mathcal{C}) = \liminf_{i \to \infty} \frac{k_i}{n_i}$

Relative distance $\delta(\mathcal{C}) = \liminf_{i \to \infty} \frac{d_i}{n_i}$

$\mathcal{C}$ asymptotically good if $R(\mathcal{C}), \delta(\mathcal{C}) > 0$.

Random binary codes are asymptotically good

$$R \geq 1 - H(\delta)$$

Gilbert-Varshamov bound

$$\left(H(\delta) = \delta \log_2\left(\frac{1}{\delta}\right) + (1-\delta)\log_2\left(\frac{1}{1-\delta}\right) \right)$$

How to build asymptotically good codes?

Start with smaller good codes and use concatenation.

Or, as today, construct sparse bipartite graphs G with excellent expansion and generate LDPC codes $C(G)$ from these graphs

During most of our coding theory lectures only viewed codes as combinatorial objects.

However, we also saw algorithmic results:

(1) Efficient decoding of Reed-Solomon codes

(2) Super-efficient decoding of asymptotically good LDPC codes