Error Correction with Codes on Graphs

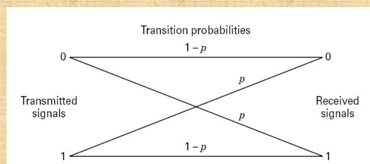
Arya Mazumdar. Advisor: Alexander Barg.

The Central Problem in Communication Systems

*Cartoon by **John Richardson**, for PHYSICS WORLD, March 1998*

The **Channel**, over which we have little control can introduce **errors, random** or **adversarial,** in the message.

An example of Noisy Channel where bits are flipped independently with error probability p, this is called a Binary Symmetric Channel



Error Correcting Codes can reduce the probability of an incorrect decision: They can correct a certain number of errors.

An example of a **code** : '0' ⟶ '000', '1' ⟶ '111'
The encoding comes with a **decoding** map : Go with the majority.
This code can correct any **one** error. The **rate** (proportion of information sent in one channel use) is only 1/3.

There are many families of codes who can correct one or more, but a fixed number of errors. For example: **Hamming Code, BCH Code**.

Oh Bob! I wish we have a code family that can correct a number of errors proportional to the length. Also the code family should have a good rate for all lengths.

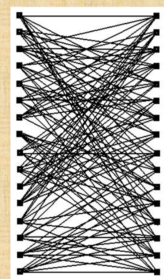
Can we help them to achieve all these using the short codes we already have!!!

There should be an efficient decoding algorithm too! We cannot wait till eternity for the message!

Concatenate short codes in a clever way: Codes on Graphs

A linear Code of rate R and length N is a set of vectors that satisfy (1-R)N linear constraints. If any two codevectors in the code are at least Hamming distance δN away then approximately δN/2 errors are correctable with the most powerful decoding.

We want code families such that both δ,R > 0 and a fast decoding algorithm that corrects τN errors for τ > 0.



Consider a regular bipartite graph $G(V_1,V_2, E)$ such that $|V_1|=|V_2|$ =m, degree = n and |E| = mn =N. For a variable {0,1}-vector **x** of length N, identify each edge of G with the coordinates of **x.**

Local code A is a short code of length n, rate R_0 and comes with a decoding that corrects e errors. The minimum distance of A is d.

With every vertex of G we associate an instance of the local code A. **x** will be a codeword of the Bipartite Graph code C(G,A) if every vertex sees a codeword of A on the edges incident to it.

The code C(G,A) has rate R to be at least $2R_0$ -1.

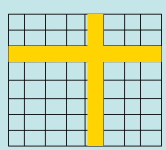
A Natural Decoding
Decode the local codes on the left side first and replace the values on the edges. Decode the right side codes now with the replaced values as the input.
Perform the above steps for a few iteration.

We give expressions for the error correcting capability for this type of codes. For example, for most of the Bipartite Graph codes:

Local Code (A)	Rate (R)	Errors correctable (τN)
Golay Code (n=23, e=3)	0.0435	0.00063N
BCH Code (n=31, e= 2)	0.3548	0.0000023N

A Special Case – Product Codes:
When n=m, the bipartite graph is 'complete'. This code is called a tensor product of A with itself.

c =



Any codeword **c** of the product code C = AOA can be represented as a matrix **c**. Any row or column of **c** is a codeword of A. This code has length n^2, rate R^2 and distance d^2.

The most powerful decoding algorithm (Maximum Likelihood) corrects many more errors than $(d^2-1)/2$, but is unacceptably complex.

Product Codes – Continues:
A simple but clever low complexity soft decision algorithm, also known as Min-Sum decoding, can correct any $(d^2-1)/2$ errors.

The product codes with local Hamming codes (or Simplex codes) have been implemented. With the min-sum decoding, they perform extremely well and in fact come close to the Maximum Likelihood performance

With a bipartite graph code C of rate R>0, The natural decoding algorithm corrects any **O(N) errors in O(N) time**. Can we improve the implicit constants of this statement?

Improving the trade-off: Codes on Hypergraph

Generalize the codes on bi-partite graphs to **t-partite hypergraphs**. Each hyper-edge (variable) is connected to (part of) exactly l vertices (local codes), one from each part. As before the hypergraph is regular with degree n and an instance of local code A is associated with each vertex.

The hypergraph code has rate R to be at least tR_0 –(t-1)

The decoding Unfortunately the natural decoding for bipartite graph code do not extend to the l-partite hyper-graph case. We still give a low complexity decoding algorithm which can correct many errors! **How many**?

We can still give expressions for number of errors correctable! Here is a set of examples that holds for most hypergraph codes:

Local Hamming Codes	Rate (R)	Errors Correctable (τN)
n=511, t= 17	0.7006	0.000235N
t=28	0.5069	0.000521N
t=40	0.2955	0.000747N
t=51	0.1018	0.000898N

Remember, we found the **guaranteed error correcting** capabilities in the worst case scenario. In practice many more errors can be corrected. For random errors, it can be shown that the graph codes can achieve **the capacity** for the **binary symmetric channel**.

I Just Want My Phone call.

Give him a phone, will ya! Make sure it uses a good code. I think the dude wants to talk to Alice desperately.

Thank You!!!!
(We solve only communication problems)