Vertex identifying codes and the random graph

Ryan Martin
rymartin@iastate.edu

Assistant Professor
Mathematics Department
Iowa State University
Joint work

This is joint work with

- Alan Frieze, Carnegie Mellon University
Joint work

This is joint work with

- Alan Frieze, Carnegie Mellon University
- Julien Moncel, Université Joseph Fourier
Joint work

This is joint work with

- Alan Frieze, Carnegie Mellon University
- Julien Moncel, Université Joseph Fourier
- Miklós Ruszinkó, Computer and Automation Institute of the Hungarian Academy of Sciences
Joint work

This is joint work with

- Alan Frieze, Carnegie Mellon University
- Julien Moncel, Université Joseph Fourier
- Miklós Ruszinkó, Computer and Automation Institute of the Hungarian Academy of Sciences
- Cliff Smyth, Carnegie Mellon University and Massachusetts Institute of Technology
Origins

Let’s begin with a network.
Origins

Let’s begin with a network.

At some point, we know there will be a failed node somewhere in the network.
Origins

Let’s begin with a network.

At some point, we know there will be a failed node somewhere in the network.

We monitor a subset of the nodes, called a code.
Origins

Let’s begin with a network.

At some point, we know there will be a failed vertex somewhere in the network.

We monitor a subset of the vertices, called a code.
Origins

Let’s begin with a network.

At some point, we know there will be a failed vertex somewhere in the network.

We monitor a subset of the vertices, called a code.

Each vertex in the code will test itself and its neighbors for failure.
An example

Consider the following graph and code.
An example

Consider the following graph and code.

The code vertices are in red.
An example

Consider the following graph and code.

This code will detect when any failure occurs.
An example

Consider the following graph and code.

This code will detect when any failure occurs.
An example

Consider the following graph and code.

This code will distinguish which node fails.
An example

Consider the following graph and code.

This code will distinguish which node fails.

If the magenta vertex reads failure and the red do not, then the failed node must be the green one.
An example

Consider the following graph and code.

This code will distinguish which node fails.

If the magenta vertices read failure, then the failed node must be the green/magenta checked one.
Identifying code

Let $G$ be a graph with vertex set $V(G)$ and edge set $E(G)$. 
Identifying code

Let $G$ be a graph with vertex set $V(G)$ and edge set $E(G)$.

Let $N(v)$ be the neighborhood of a vertex.
Identifying code

Let $G$ be a graph with vertex set $V(G)$ and edge set $E(G)$.

Let $N(v)$ be the neighborhood of a vertex.

Let $N[v] = N(v) \cup \{v\}$ be the closed neighborhood of a vertex.
Identifying code

Let $N(v)$ be the neighborhood of a vertex.

Let $N[v] = N(v) \cup \{v\}$ be the closed neighborhood of a vertex.

A dominating set is a $C \subseteq V(G)$ such that $N[v] \cap C \neq \emptyset \quad \forall v \in V(G)$. 
Identifying code

Let $N(v)$ be the neighborhood of a vertex.

Let $N[v] = N(v) \cup \{v\}$ be the closed neighborhood of a vertex.

A **dominating set** is a $C \subseteq V(G)$ such that $N[v] \cap C \neq \emptyset \quad \forall v \in V(G)$.

An **identifying code** is a $C \subseteq V(G)$ such that
Identifying code

Let \( N(v) \) be the neighborhood of a vertex.

Let \( N[v] = N(v) \cup \{v\} \) be the closed neighborhood of a vertex.

A dominating set is a \( C \subseteq V(G) \) such that
\[
N[v] \cap C \neq \emptyset \quad \forall v \in V(G).
\]

An identifying code is a \( C \subseteq V(G) \) such that
- \( C \) is a dominating set and
Identifying code

Let $N(v)$ be the neighborhood of a vertex.

Let $N[v] = N(v) \cup \{v\}$ be the closed neighborhood of a vertex.

A dominating set is a $C \subseteq V(G)$ such that $N[v] \cap C \neq \emptyset \quad \forall v \in V(G)$.

An identifying code is a $C \subseteq V(G)$ such that

- $C$ is a dominating set and
- $N[v] \cap C \neq N[w] \cap C$ for all $v \neq w$. 
Petersen code

Let’s return to the example of the Petersen graph.
Petersen code

Let’s return to the example of the Petersen graph.
Petersen code

Let’s return to the example of the Petersen graph.

The red vertices are an identifying code of size 5.
Petersen code

Let’s return to the example of the Petersen graph.

This is a smaller set.
Petersen code

Let’s return to the example of the Petersen graph.

This is a smaller set. Smaller is better for this problem. But, is it a code?
Petersen code

Let’s return to the example of the Petersen graph.

This is a smaller set. Smaller is better for this problem. But, is it a code?
Petersen code

Let’s return to the example of the Petersen graph.

This is a smaller set. Smaller is better for this problem. But, is it a code?
Petersen code

Let’s return to the example of the Petersen graph.

This is a smaller set. Smaller is better for this problem. But, is it a code?
Petersen code

Let’s return to the example of the Petersen graph.

This is a smaller set. Smaller is better for this problem. But, is it a code?
Petersen code

Let’s return to the example of the Petersen graph.

This is a smaller set. Smaller is better for this problem. But, is it a code?
Petersen code

Let’s return to the example of the Petersen graph.

This is a smaller set. Smaller is better for this problem. But, is it a code?
Petersen code

Let’s return to the example of the Petersen graph.

This is a smaller set. Smaller is better for this problem. But, is it a code?
Petersen code

Let’s return to the example of the Petersen graph.

This is a smaller set. Smaller is better for this problem. But, is it a code?
Petersen code

Let’s return to the example of the Petersen graph.

This is a smaller set. Smaller is better for this problem. But, is it a code?

Yes!
Petersen code

Let’s return to the example of the Petersen graph.

This is a smaller set. Smaller is better for this problem. But, is it a code?

Yes!

Is there a smaller code?
Lower bound

If $C$ is a code of size 3 in the Petersen graph, $P_{10}$,
Lower bound

If $C$ is a code of size 3 in the Petersen graph, $P_{10}$, then the map
Lower bound

If $C$ is a code of size 3 in the Petersen graph, $P_{10}$, then the map

$$f : V(P_{10}) \rightarrow 2^C \setminus \emptyset$$
Lower bound

If $C$ is a code of size 3 in the Petersen graph, $P_{10}$, then the map

$$f : V(P_{10}) \rightarrow 2^C \setminus \emptyset$$

$$f(v) = N[v] \cap C$$
Lower bound

If $C$ is a code of size 3 in the Petersen graph, $P_{10}$, then the map

$$f : V(P_{10}) \to 2^C \setminus \emptyset$$

$$f(v) = N[v] \cap C$$

is an injection.
Lower bound

If $C$ is a code, then the map

$$f : V(G) \rightarrow 2^C \setminus \emptyset$$

$$f(v) = N[v] \cap C$$

is an injection.

Thus, $10 \leq 2^3 - 1$, a contradiction.
Theorem. [Karpovsky, Chakrabarty, Levitin]

Let \( G \) be a graph with identifying code \( C \), then \( \vert C \vert \geq \log_2 (n+1) \).

Proof.

If \( C \) is a code, then the map

\[
\begin{align*}
  f : \ V(G) &\to 2^C \setminus \emptyset \\
  f(v) &\mapsto N[v] \cap C
\end{align*}
\]

is an injection.

Thus, \( n \leq 2^{|C|} - 1 \)
Lower bound

If $C$ is a code, then the map

$$f : V(G) \rightarrow 2^C \setminus \emptyset$$

$$f(v) = N[v] \cap C$$

is an injection.

Thus, $n \leq 2^{|C|} - 1 \iff |C| \geq \log_2(n + 1)$. 
Lower bound

If $C$ is a code, then the map

$$f : V(G) \rightarrow 2^C \setminus \emptyset$$

$$f(v) = N[v] \cap C$$

is an injection.

Thus, $n \leq 2^{|C|} - 1 \Leftrightarrow |C| \geq \lceil \log_2(n + 1) \rceil$. 
Lower bound

Theorem. [Karpovsky, Chakrabarty, Levitin]
Let $G$ be a graph with identifying code $C$, then $|C| \geq \lceil \log_2(n + 1) \rceil$.

Proof.
If $C$ is a code, then the map

$$ f : V(G) \rightarrow 2^C \setminus \emptyset $$

$$ f(v) = N[v] \cap C $$

is an injection.

Thus, $n \leq 2^{|C|} - 1 \Leftrightarrow |C| \geq \lceil \log_2(n + 1) \rceil$. 
Lower bound

**Theorem.** [Karpovsky, Chakrabarty, Levitin]
Let $G$ be a graph with identifying code $C$, then $|C| \geq \lceil \log_2(n + 1) \rceil$.

**Proof.**
If $C$ is a code, then the map

$$f : V(G) \rightarrow 2^C \setminus \emptyset$$

$$f(v) = N[v] \cap C$$

is an injection.

Thus, $n \leq 2^{|C|} - 1 \iff |C| \geq \lceil \log_2(n + 1) \rceil$. \(\square\)
We don’t need no stinkin’ code

What if no code exists?
We don’t need no stinkin’ code

What if no code exists?

The complete graph has no code.
We don’t need no stinkin’ code

What if no code exists?

The complete graph, $K_n$, $n \geq 1$, has no code.
We don’t need no stinkin’ code
What if no code exists?

The complete graph, $K_n$, $n \geq 1$, has no code.

This is because

$$N[v] \cap C = N[w] \cap C.$$
We don’t need no stinkin’ code

What if no code exists?

The complete graph, $K_n$, $n \geq 1$, has no code.

This is because

$$N[v] \cap C = N[w] \cap C,$$

for all $C \subseteq V(K_n)$. 
We don’t need no stinkin’ code

What if no code exists?

The complete graph, \( K_n, n \geq 1 \), has no code.

This is because

\[
N[v] \cap C = N[w] \cap C,
\]

for all \( C \subseteq V(K_n) \),

and all \( v, w \in V(G) \).
Codes among us

If there exists a code in graph \( G \), . . .
Theorem. \[ \text{KCL} \]

A graph \( G \) admits an identifying code if and only if 
\[ N[v] \neq N[w], \]
for all distinct \( v, w \in V(G) \).

Proof. If there exists a code in graph \( G \), then 
\[ N[v] \neq N[w], \]
for all distinct \( v, w \in V(G) \).

If \( N[v] \neq N[w] \) for all distinct \( v, w \in V(G) \), then 
\( C = V(G) \) is a code. (Closed neighborhoods are always nonempty.)
If there exists a code in graph $G$, then $N[v] \neq N[w]$, for all distinct $v, w \in V(G)$.

If $N[v] \neq N[w]$ for all distinct $v, w \in V(G)$, . . .
Theorem. \[KCL\]

A graph \(G\) admits an identifying code if and only if \(N[v] \neq N[w]\), for all distinct \(v, w \in V(G)\).

**Proof.** If there exists a code in graph \(G\), then \(N[v] \neq N[w]\), for all distinct \(v, w \in V(G)\).

If \(N[v] \neq N[w]\) for all distinct \(v, w \in V(G)\), then \(C = V(G)\) is a code.
Codes among us

If there exists a code in graph $G$, then $N[v] \neq N[w]$, for all distinct $v, w \in V(G)$.

If $N[v] \neq N[w]$ for all distinct $v, w \in V(G)$, then $C = V(G)$ is a code. (Closed neighborhoods are always nonempty.)
Codes among us

**Theorem. [KCL]** A graph $G$ admits an identifying code if and only if $N[v] \neq N[w]$, for all distinct $v, w \in V(G)$.

**Proof.**
If there exists a code in graph $G$, then $N[v] \neq N[w]$, for all distinct $v, w \in V(G)$.

If $N[v] \neq N[w]$ for all distinct $v, w \in V(G)$, then $C = V(G)$ is a code. (Closed neighborhoods are always nonempty.)
Random graphs

Consider the usual model of a random graph.
Random graphs

Consider the usual model of a random graph.

Fix $n$ vertices.
Random graphs

Consider the usual model of a random graph.

Fix $n$ vertices.

For each pair $\{v, w\} \in \binom{[n]}{2}$,
Random graphs

Consider the usual model of a random graph.

Fix \( n \) vertices.

For each pair \( \{v, w\} \in \binom{[n]}{2} \), make \( v \sim w \) with probability \( p \),
Random graphs

Consider the usual model of a random graph.

Fix $n$ vertices.

For each pair $\{v,w\} \in \binom{[n]}{2}$, make $v \sim w$ with probability $p$, each pair is independent.
Theorem. [FMMRS] If $E(n, p)$ is the event that $G_{n,p}$ has an identifying code, then $\lim_{n \to \infty} \Pr (E(n, p))$ is
Theorem. [FMMRS] If $\mathcal{E}(n, p)$ is the event that $G_{n,p}$ has an identifying code, then $\lim_{n \to \infty} \Pr(\mathcal{E}(n, p))$ is

\[ \begin{align*}
1, & \quad \text{if } p = o(n^{-2}); \\
1, & \quad \text{if } p = \frac{\ln n}{n}; \\
1, & \quad \text{if } p = \frac{\ln^2 n}{n}; \\
0, & \quad \text{if } p = \frac{\ln n}{n^2}.
\end{align*} \]
Codes in random graphs

**Theorem. [FMMRS]** If $\mathcal{E}(n, p)$ is the event that $G_{n,p}$ has an identifying code, then $\lim_{n \to \infty} \Pr(\mathcal{E}(n, p))$ is

\[
\begin{align*}
1, & \quad \text{if } p = o(n^{-2}); \\
\exp\left(-c_1/2\right), & \quad \text{if } p = c_1 n^{-2};
\end{align*}
\]
Codes in random graphs

**Theorem. [FMMRS]** If $\mathcal{E}(n, p)$ is the event that $G_{n,p}$ has an identifying code, then $\lim_{n \to \infty} \Pr (\mathcal{E}(n, p))$ is

$$1, \text{ if } p = o(n^{-2});$$

$$e^{-c_1/2}, \text{ if } p = c_1 n^{-2};$$

$$0, \text{ if } \omega(n^{-2}) = p = \frac{\ln n + \ln \ln n - \omega(1)}{2n};$$
Theorem. [FMMRS] If $E(n, p)$ is the event that $G_{n,p}$ has an identifying code, then $\lim_{n \to \infty} \Pr (E(n, p))$ is

$$
\begin{align*}
1, & \quad \text{if } p = o(n^{-2}); \\
\ e^{-c_1/2}, & \quad \text{if } p = c_1 n^{-2}; \\
\ 0, & \quad \text{if } \omega(n^{-2}) = p = \frac{\ln n + \ln \ln n - \omega(1)}{2n}; \\
\ e^{-e^{-c_2/4}}, & \quad \text{if } p = \frac{\ln n + \ln \ln n + c_2}{2n};
\end{align*}
$$
Theorem. [FMMRS] If $\mathcal{E}(n, p)$ is the event that $G_{n,p}$ has an identifying code, then $\lim_{n \to \infty} \Pr(\mathcal{E}(n, p))$ is

\[
\begin{align*}
1, & \quad \text{if } p = o(n^{-2}); \\
e^{-c_1/2}, & \quad \text{if } p = c_1 n^{-2}; \\
0, & \quad \text{if } \omega(n^{-2}) = p = \frac{\ln n + \ln \ln n - \omega(1)}{2n}; \\
e^{-e^{-c_2/4}}, & \quad \text{if } p = \frac{\ln n + \ln \ln n + c_2}{2n}; \\
1, & \quad \text{if } \frac{\ln n + \ln \ln n + \omega(1)}{2n} = p
\end{align*}
\]

and $p = 1 - \frac{\ln n + \omega(1)}{n}$;
Theorem. [FMMRS] If $\mathcal{E}(n, p)$ is the event that $G_{n,p}$ has an identifying code, then $\lim_{n \to \infty} \Pr \left( \mathcal{E}(n, p) \right)$ is

\[
\begin{align*}
1, & \quad \text{if } p = o(n^{-2}); \\
e^{-c_1/2}, & \quad \text{if } p = c_1 n^{-2}; \\
o, & \quad \text{if } \omega(n^{-2}) = p = \frac{\ln n + \ln \ln n - \omega(1)}{2n} \\
e^{-e^{-c_2/4}}, & \quad \text{if } p = \frac{\ln n + \ln \ln n + c_2}{2n} \\
1, & \quad \text{if } \frac{\ln n + \ln \ln n + \omega(1)}{2n} = p \\
e^{-e^{-c_3}}(1 + e^{-c_3}), & \quad \text{if } p = 1 - \frac{\ln n + c_3}{n};
\end{align*}
\]
Theorem. [FMMRS] If $\mathcal{E}(n, p)$ is the event that $G_{n,p}$ has an identifying code, then $\lim_{n \to \infty} \Pr (\mathcal{E}(n, p))$ is

\[
\begin{align*}
1, & \quad \text{if } p = o(n^{-2}); \\
e^{-c_1/2}, & \quad \text{if } p = c_1 n^{-2}; \\
0, & \quad \text{if } \omega(n^{-2}) = p = \frac{\ln n + \ln \ln n - \omega(1)}{2n}; \\
e^{-e^{-c_2/4}}, & \quad \text{if } p = \frac{\ln n + \ln \ln n + c_2}{2n}; \\
1, & \quad \text{if } \frac{\ln n + \ln \ln n + \omega(1)}{2n} = p \\
\text{and } p = 1 - \frac{\ln n + \omega(1)}{n}; \\
e^{-e^{-c_3} (1 + e^{-c_3})}, & \quad \text{if } p = 1 - \frac{\ln n + c_3}{n}; \\
0, & \quad \text{if } p = 1 - \frac{\ln n - \omega(1)}{n}.
\end{align*}
\]
Theorem. [FMMRS] If $\mathcal{E}(n, p)$ is the event that $G_{n,p}$ has an identifying code, then

$$
\lim_{{n \to \infty}} \Pr (\mathcal{E}(n, p)) = 1,
$$

if

$$
\frac{\ln n + \ln \ln n + \omega(1)}{2n} = p = 1 - \frac{\ln n + \omega(1)}{n}.
$$
Codes in random graphs

**Theorem. [FMMRS]** If $\mathcal{E}(n, p)$ is the event that $G_{n,p}$ has an identifying code, then

$$
1 - \frac{1}{n} \log n
$$
For a graph $G$, define $c(G)$.
Code sizes in random graphs

For a graph $G$, define $c(G)$ to be the size of a smallest identifying code in $G$. 
Code sizes in random graphs

For a graph $G$, define $c(G)$ to be the size of a smallest identifying code in $G$. (If none, $c(G) = \infty$.)
Code sizes in random graphs

For a graph $G$, define $c(G)$ to be the size of a smallest identifying code in $G$. (If none, $c(G) = \infty$.)

**Theorem.** [FMMRS] Let $0 < p < 1$. 
Code sizes in random graphs

For a graph $G$, define $c(G)$ to be the size of a smallest identifying code in $G$. (If none, $c(G) = \infty$.)

**Theorem. [FMMRS]** Let $0 < p < 1$.

$$q \overset{\text{def}}{=} p^2 + (1 - p)^2.$$
Code sizes in random graphs

For a graph $G$, define $c(G)$ to be the size of a smallest identifying code in $G$. (If none, $c(G) = \infty$.)

**Theorem. [FMMRS]** Let $0 < p < 1$.

\[ q \overset{\text{def}}{=} p^2 + (1 - p)^2. \]

For almost every graph in $G(n, p)$, we have
Code sizes in random graphs

For a graph $G$, define $c(G)$ to be the size of a smallest identifying code in $G$. (If none, $c(G) = \infty$.)

**Theorem. [FMMRS]**  Let $0 < p < 1$.

\[ q \overset{\text{def}}{=} p^2 + (1 - p)^2. \]

For almost every graph in $G(n, p)$, we have

\[ c(G_{n,p}) \sim \frac{2 \log n}{\log(1/q)}. \]
Theorem. [FMMRS] Let $0 < p < 1$.

$$q \overset{\text{def}}{=} p^2 + (1 - p)^2.$$ 

For almost every graph in $G(n, p)$, we have

$$c(G_{n,p}) \sim \frac{2 \log n}{\log(1/q)}.$$ 

I.e., for all $\epsilon > 0$,

$$\lim_{n \to \infty} \Pr \left( \left| c(G_{n,p}) \cdot \left( \frac{2 \log n}{\log(1/q)} \right)^{-1} - 1 \right| \geq \epsilon \right) = 0.$$
Code sizes in random graphs

**Theorem. [FMMRS]** Let \( p, 1 - p \geq 4 \ln \ln n / \ln n \).

\[ q \overset{\text{def}}{=} p^2 + (1 - p)^2. \]

For almost every graph in \( G(n, p) \), we have

\[ c(G_{n,p}) \sim \frac{2 \log n}{\log(1/q)}. \]

I.e., for all \( \epsilon > 0 \),

\[ \lim_{n \to \infty} \Pr \left( \left| c(G_{n,p}) \cdot \left( \frac{2 \log n}{\log(1/q)} \right)^{-1} - 1 \right| \geq \epsilon \right) = 0. \]
Code sizes in random graphs

Theorem. [FMMRS] Let $p, 1 - p \geq 4 \ln \ln n / \ln n$.

$$q \overset{\text{def}}{=} p^2 + (1 - p)^2.$$  

For almost every graph in $G(n, p)$, we have

$$c(G_{n,p}) \sim \frac{2 \log n}{\log(1/q)}.$$  

The proof uses an inequality of Suen.
Theorem. [FMMRS] Let $p, 1 - p \geq 4 \ln \ln n / \ln n$.

$$q \overset{\text{def}}{=} p^2 + (1 - p)^2.$$  

For almost every graph in $G(n, p)$, we have

$$c(G_{n,p}) \sim \frac{2 \log n}{\log(1/q)}.$$  

The proof uses an inequality of Suen. Suen’s inequality resembles the Lovász Local Lemma, in that there is a dependency graph.
Suen’s inequality

In 1990, Stephen Suen found a very useful correlation inequality.

Theorem. (Suen)

Let \( f_i \) and \( g_i \) be a set of events.

Let \( I_i \) be the indicator of event \( A_i \).

Construct a dependency graph with the following:

If any disjoint vertex-subsets \( J_1 \) and \( J_2 \) have no edges from \( J_1 \) to \( J_2 \), then \( f_i g_i \) and \( f_j g_j \) are independent.

(That is, any Boolean combination of events in \( J_1 \) is independent of any Boolean combination of events in \( J_2 \).)

With all this, we can conclude that

\[
\Pr_X \left( \bigcap_{i \in I} E(i) \right) \geq \exp \left( -\max_i \Pr_X \left( \bigcup_{j \neq i} E(j) \right) \right)
\]
Suen’s inequality

In 1990, Stephen Suen found a very useful correlation inequality.

In 1998, Svante Janson expanded and generalized the idea.

Let $\mathbf{A}_i$ and $\mathbf{g}_i$ be a set of events.

Let $\mathbf{I}_i$ be the indicator of event $\mathbf{A}_i$.

Construct a dependency graph with the following:

If any disjoint vertex-subsets $J_1$ and $J_2$ have no edges from $J_1$ to $J_2$,
then $\mathbf{I}_i$ and $\mathbf{I}_j$ are independent.

(That is, any Boolean combination of events in $J_1$ is independent of any Boolean combination of events in $J_2$.)

With all this, we can conclude that

$$\Pr\left( \bigcap_{i \in I} \mathbf{I}_i \right) = \max_{i \in I} \Pr(\mathbf{I}_i).$$
Suen’s inequality

In 1990, Stephen Suen found a very useful correlation inequality.

In 1998, Svante Janson expanded and generalized the idea.

Our work uses a simplified, but powerful, version.

**Theorem. [Suen]** Let \( \{A_i\}_{i \in I} \) be a set of events.
Suen’s inequality

In 1990, Stephen Suen found a very useful correlation inequality.

In 1998, Svante Janson expanded and generalized the idea.

Our work uses a simplified, but powerful, version.

**Theorem. [Suen]** Let $\{A_i\}_{i \in \mathcal{I}}$ be a set of events. Let $I_i$ be the indicator of event $A_i$. 
Suen’s inequality

**Theorem. [Suen]** Let \( \{A_i\}_{i \in \mathcal{I}} \) be a set of events. Let \( I_i \) be the indicator of event \( A_i \).
Suen’s inequality

**Theorem. [Suen]** Let \( \{ A_i \}_{i \in I} \) be a set of events.

Let \( I_i = \begin{cases} 1, & \text{if } A_i \text{ occurs;} \\ 0, & \text{otherwise.} \end{cases} \)
Suen’s inequality

**Theorem. [Suen]** Let \( \{A_i\}_{i \in I} \) be a set of events.

Let \( I_i = \begin{cases} 1, & \text{if } A_i \text{ occurs;} \\ 0, & \text{otherwise.} \end{cases} \)

Construct a dependency graph with the following:

"..."
Suen’s inequality

**Theorem. [Suen]** Let \( \{A_i\}_{i \in I} \) be a set of events.

Let \( I_i = \begin{cases} 
1, & \text{if } A_i \text{ occurs;} \\
0, & \text{otherwise.}
\end{cases} \)

Construct a dependency graph with the following:

If any **disjoint** vertex-subsets \( J_1 \) and \( J_2 \) have **no** edges from \( J_1 \) to \( J_2 \),
Suen’s inequality

**Theorem.** [Suen] Let \( \{ A_i \}_{i \in \mathcal{I}} \) be a set of events.

Let \( I_i = \begin{cases} 1, & \text{if } A_i \text{ occurs;} \\ 0, & \text{otherwise.} \end{cases} \)

Construct a dependency graph with the following:

If any **disjoint** vertex-subsets \( J_1 \) and \( J_2 \) have **no** edges from \( J_1 \) to \( J_2 \),
then \( \{ A_i \}_{i \in J_1} \) and \( \{ A_j \}_{j \in J_2} \) are independent.
Suen’s inequality

**Theorem. [Suen]** Let \( \{A_i\}_{i \in I} \) be a set of events. Let \( I_i = \begin{cases} 1, & \text{if } A_i \text{ occurs;} \\ 0, & \text{otherwise.} \end{cases} \)

Construct a dependency graph with the following:

If any disjoint vertex-subsets \( J_1 \) and \( J_2 \) have no edges from \( J_1 \) to \( J_2 \), then \( \{A_i\}_{i \in J_1} \) and \( \{A_j\}_{j \in J_2} \) are independent.

(That is, any Boolean combination of events in \( J_1 \) is independent of any Boolean combination of events in \( J_2 \).)
**Theorem.** [Suen]

Construct a dependency graph with the following:

If any **disjoint** vertex-subsets $J_1$ and $J_2$ have no edges from $J_1$ to $J_2$, then \( \{A_i\}_{i \in J_1} \) and \( \{A_j\}_{j \in J_2} \) are independent.

(That is, any Boolean combination of events in $J_1$ is independent of any Boolean combination of events in $J_2$.)

\[
\mu := \sum_{i \in I} \mathbb{E}(I_i)
\]
Suen’s inequality

**Theorem.** [Suen]

Construct a dependency graph with the following:

If any **disjoint** vertex-subsets $J_1$ and $J_2$ have **no** edges from $J_1$ to $J_2$, then $\{A_i\}_{i \in J_1}$ and $\{A_j\}_{j \in J_2}$ are independent.

(That is, any Boolean combination of events in $J_1$ is independent of any Boolean combination of events in $J_2$.)

- $\mu := \sum_{i \in I} \mathbb{E}(I_i)$
- $\Delta := \sum_{\{\{i,j\}: i \sim j\}} \mathbb{E}(I_i I_j)$
Suen’s inequality

**Theorem. [Suen]**

Construct a dependency graph with the following:

If any **disjoint** vertex-subsets $J_1$ and $J_2$ have no edges from $J_1$ to $J_2$, then $\{A_i\}_{i \in J_1}$ and $\{A_j\}_{j \in J_2}$ are independent.

(That is, any Boolean combination of events in $J_1$ is independent of any Boolean combination of events in $J_2$.)

- $\mu := \sum_{i \in I} \mathbb{E}(I_i)$
- $\Delta := \sum_{\{i,j\}: i \sim j} \mathbb{E}(I_i I_j)$
- $\delta := \max_{i \in I} \sum_{j \sim i} \mathbb{E}(I_j)$
Suen’s inequality

**Theorem. [Suen]**
Construct a dependency graph with the following:

If any **disjoint** vertex-subsets $J_1$ and $J_2$ have **no** edges from $J_1$ to $J_2$, then $\{A_i\}_{i \in J_1}$ and $\{A_j\}_{j \in J_2}$ are independent.

- $\mu := \sum_{i \in I} \mathbb{E}(I_i)$
- $\Delta := \sum_{\{i,j\} : i \sim j} \mathbb{E}(I_i I_j)$
- $\delta := \max_{i \in I} \sum_{j \sim i} \mathbb{E}(I_j)$
Suen’s inequality

**Theorem.** [Suen]
Construct a dependency graph with the following:

If any **disjoint** vertex-subsets $J_1$ and $J_2$ have **no** edges from $J_1$ to $J_2$, then $\{A_i\}_{i \in J_1}$ and $\{A_j\}_{j \in J_2}$ are independent.

*●* $\mu := \sum_{i \in \mathcal{I}} \mathbb{E}(I_i)$
*●* $\Delta := \sum_{\{i,j\}:i \sim j} \mathbb{E}(I_i I_j)$
*●* $\delta := \max_{i \in \mathcal{I}} \sum_{j \sim i} \mathbb{E}(I_j)$

With all this, we can conclude that
Suen’s inequality

**Theorem. [Suen]**

Construct a dependency graph with the following:

If any disjoint vertex-subsets $J_1$ and $J_2$ have no edges from $J_1$ to $J_2$, then \( \{A_i\}_{i \in J_1} \) and \( \{A_j\}_{j \in J_2} \) are independent.

- \( \mu := \sum_{i \in \mathcal{I}} \mathbb{E}(I_i) \)
- \( \Delta := \sum_{\{i,j\} : i \sim j} \mathbb{E}(I_i I_j) \)
- \( \delta := \max_{i \in \mathcal{I}} \sum_{j \sim i} \mathbb{E}(I_j) \)

With all this, we can conclude that

\[
\Pr\left(\sum_{i \in \mathcal{I}} I_i = 0\right) \leq \exp \left\{ -\mu + \Delta e^{2\delta} \right\}.
\]
Using Suen

We use this for the lower bound.
Using Suen

We use this for the lower bound.

That is, if $C$ is a vertex-subset that is too small, $C$ cannot be a code in $G_{n,p}$. 
Using Suen

We use this for the lower bound.

That is, if $C$ is a vertex-subset that is too small, $C$ cannot be a code in $G_{n,p}$.

Fix $C$
Using Suen

We use this for the lower bound.

That is, if $C$ is a vertex-subset that is too small, $C$ cannot be a code in $G_{n,p}$.

Fix $C$, $c := |C|$, 

Using Suen

We use this for the lower bound.

That is, if $C$ is a vertex-subset that is too small, $C$ cannot be a code in $G_{n,p}$.

Fix $C$, $c := |C|$, and let $\mathcal{I}$ consist of the pairs of vertices

$$\mathcal{I} = \{i, j \mid i \neq j, i, j \in C\}$$

Let $a = p^2 + (1 - p)^2$.

Finally, we can conclude that if $c \geq (2 - a) \log n \log(1/p)$, then $Pr(C$ is a code $) = o(n^c)$. 
Using Suen

We use this for the lower bound.

That is, if $C$ is a vertex-subset that is too small, $C$ cannot be a code in $G_{n,p}$.

Fix $C$, $c := |C|$, and let $\mathcal{I}$ consist of the \textbf{pairs of vertices} of $V \setminus C$.
Using Suen

We use this for the lower bound.

That is, if \( C \) is a vertex-subset that is too small, \( C \) cannot be a code in \( G_{n,p} \).

Fix \( C, c := |C| \), and let \( \mathcal{I} \) consist of the pairs of vertices of \( V \setminus C \).

So, each \( A_i \) is the event that two vertices have the same neighborhood in \( C \).
Using Suen

We use this for the lower bound.

That is, if \( C \) is a vertex-subset that is too small, \( C \) cannot be a code in \( G_{n,p} \).

Fix \( C, \ c := \vert C \vert \), and let \( \mathcal{I} \) consist of the **pairs of vertices** of \( V \setminus C \).

So, each \( A_i \) is the event that two vertices have the same neighborhood in \( C \).

Let \( i \sim j \) if and only if the vertex sets that they represent overlap.
Using Suen

We use this for the lower bound.

That is, if $C$ is a vertex-subset that is too small, $C$ cannot be a code in $G_{n,p}$.

Fix $C$, $c := |C|$, and let $\mathcal{I}$ consist of the pairs of vertices of $V \setminus C$.

So, each $A_i$ is the event that two vertices have the same neighborhood in $C$.

Let $i \sim j$ if and only if the vertex sets that they represent overlap. This is a dependency graph.
Using Suen

Fix $C$, $c := |C|$, and let $\mathcal{I}$ consist of the pairs of vertices of $V \setminus C$.

Let $i \sim j$ if and only if the vertex sets that they represent overlap. This is a dependency graph.
Using Suen

Fix $C$, $c := |C|$, and let $\mathcal{I}$ consist of the pairs of vertices of $V \setminus C$.

Let $i \sim j$ if and only if the vertex sets that they represent overlap. This is a dependency graph.

Pairs that produce an edge.
Using Suen

Fix $C, c := |C|$, and let $\mathcal{I}$ consist of the \textbf{pairs of vertices} of $V \setminus C$.

Let $i \sim j$ if and only if the vertex sets that they represent overlap. This is a dependency graph.

Pairs that produce a nonedge.

Let $q = p^2 + (1 - p)^2$.

Finally, we can conclude that if $c (2^{1/2}) \log n \log (1 - q)$, then $\Pr (C \text{ is a code}) = o(n^{c - 1})$. 
Using Suen

Fix $C$, $c := |C|$, and let $\mathcal{I}$ consist of the pairs of vertices of $V \setminus C$.

Let $i \sim j$ if and only if the vertex sets that they represent overlap. This is a dependency graph.

- $\mu := \sum_{i \in \mathcal{I}} \mathbb{E}(I_i)$
- $\Delta := \sum_{\{i,j\}:i \sim j} \mathbb{E}(I_i I_j)$
- $\delta := \max_{i \in \mathcal{I}} \sum_{j \sim i} \mathbb{E}(I_i)$
Using Suen

Fix $C$, $c := |C|$, and let $\mathcal{I}$ consist of the **pairs of vertices** of $V \setminus C$.

Let $i \sim j$ if and only if the vertex sets that they represent overlap. This is a dependency graph.

- $\mu := \sum_{i \in \mathcal{I}} \left( p^2 + (1 - p)^2 \right)^c$
- $\Delta := \sum_{\{i, j\}: i \sim j} \mathbb{E}(I_i I_j)$
- $\delta := \max_{i \in \mathcal{I}} \sum_{j \sim i} \mathbb{E}(I_i)$
Using Suen

Fix $C$, $c := |C|$, and let $\mathcal{I}$ consist of the pairs of vertices of $V \setminus C$.

Let $i \sim j$ if and only if the vertex sets that they represent overlap. This is a dependency graph.

- $\mu := \sum_{i \in \mathcal{I}} (p^2 + (1 - p)^2)^c$
- $\Delta := \sum_{\{\{i,j\}: i \sim j\}} (p^3 + (1 - p)^3)^c$
- $\delta := \max_{i \in \mathcal{I}} \sum_{j \sim i} \mathbb{E}(I_i)$
Using Suen

Fix $C$, $c := |C|$, and let $\mathcal{I}$ consist of the pairs of vertices of $V \setminus C$.

Let $i \sim j$ if and only if the vertex sets that they represent overlap. This is a dependency graph.

- $\mu := \sum_{i \in \mathcal{I}} (p^2 + (1 - p)^2)^c$
- $\Delta := \sum_{\{i, j\} : i \sim j} (p^3 + (1 - p)^3)^c$
- $\delta := \max_{i \in \mathcal{I}} \sum_{j \sim i} (p^2 + (1 - p)^2)^c$
Using Suen

Fix $C$, $c := |C|$, and let $\mathcal{I}$ consist of the **pairs of vertices** of $V \setminus C$.

Let $i \sim j$ if and only if the vertex sets that they represent overlap. This is a dependency graph.

- $\mu := \binom{n-c}{2} \left( p^2 + (1 - p)^2 \right)^c$
- $\Delta := \sum_{\{i,j\}: i \sim j} \left( p^3 + (1 - p)^3 \right)^c$
- $\delta := \max_{i \in \mathcal{I}} \sum_{j \sim i} \left( p^2 + (1 - p)^2 \right)^c$

Finally, we can conclude that if

$$c(2 \log_2 n \log(1 - q)) \leq \frac{\log n}{q}$$

then

$$\Pr(\text{$C$ is a code}) = o(\frac{1}{n^c}).$$
Using Suen

Fix $C$, $c := |C|$, and let $\mathcal{I}$ consist of the **pairs of vertices** of $V \setminus C$.

Let $i \sim j$ if and only if the vertex sets that they represent overlap. This is a dependency graph.

- $\mu := \binom{n-c}{2} \left( p^2 + (1 - p)^2 \right)^c$
- $\Delta := 3 \binom{n-c}{3} \left( p^3 + (1 - p)^3 \right)^c$
- $\delta := \max_{i \in \mathcal{I}} \sum_{j \sim i} \left( p^2 + (1 - p)^2 \right)^c$
Fix $C$, $c := |C|$, and let $I$ consist of the pairs of vertices of $V \setminus C$.

Let $i \sim j$ if and only if the vertex sets that they represent overlap. This is a dependency graph.

- $\mu := \binom{n-c}{2} \left( p^2 + (1 - p)^2 \right)^c$
- $\Delta := 3 \binom{n-c}{3} \left( p^3 + (1 - p)^3 \right)^c$
- $\delta := 2(n - c - 2) \left( p^2 + (1 - p)^2 \right)^c$
Using Suen

Fix $C$, $c := |C|$, and let $\mathcal{I}$ consist of the **pairs of vertices** of $V \setminus C$.

Let $i \sim j$ if and only if the vertex sets that they represent overlap. This is a dependency graph.

- $\mu := \binom{n-c}{2} \left(p^2 + (1 - p)^2\right)^c$
- $\Delta := 3\binom{n-c}{3} \left(p^3 + (1 - p)^3\right)^c$
- $\delta := 2(n - c - 2) \left(p^2 + (1 - p)^2\right)^c$

Let $q = p^2 + (1 - p)^2$. 
Using Suen

Fix $C$, $c := |C|$, and let $\mathcal{I}$ consist of the pairs of vertices of $V \setminus C$.

Let $i \sim j$ if and only if the vertex sets that they represent overlap. This is a dependency graph.

- $\mu := \binom{n-c}{2} q^c$
- $\Delta := 3 \binom{n-c}{3} \left( p^3 + (1 - p)^3 \right)^c$
- $\delta := 2(n - c - 2) q^c$

Let $q = p^2 + (1 - p)^2$.

Finally, we can conclude that
Using Suen

Fix $C, c := |C|$, and let $\mathcal{I}$ consist of the pairs of vertices of $V \setminus C$.

Let $i \sim j$ if and only if the vertex sets that they represent overlap. This is a dependency graph.

- $\mu := \binom{n-c}{2} q^c$
- $\Delta := 3 \binom{n-c}{3} \left(p^3 + (1 - p)^3\right)^c$
- $\delta := 2(n - c - 2) q^c$

Let $q = p^2 + (1 - p)^2$.

Finally, we can conclude that if $c \leq \frac{(2-\epsilon) \log n}{\log(1/q)}$
Fix $C$, $c := |C|$, and let $\mathcal{I}$ consist of the pairs of vertices of $V \setminus C$.

Let $i \sim j$ if and only if the vertex sets that they represent overlap. This is a dependency graph.

- $\mu := \binom{n-c}{2} q^c$
- $\Delta := 3 \binom{n-c}{3} \left(p^3 + (1 - p)^3\right)^c$
- $\delta := 2(n - c - 2) q^c$

Let $q = p^2 + (1 - p)^2$.

Finally, we can conclude that if $c \leq \frac{(2-\epsilon) \log n}{\log(1/q)}$, then

$$\Pr (C \text{ is a code}) = o \left( \binom{n}{c}^{-1} \right).$$
Using Suen

Fix $C$, $c := |C|$, and let $\mathcal{I}$ consist of the **pairs of vertices** of $V \setminus C$.

Let $i \sim j$ if and only if the vertex sets that they represent overlap. This is a dependency graph.

- $\mu := \binom{n-c}{2} q^c$
- $\Delta := 3 \binom{n-c}{3} (p^3 + (1 - p)^3)^c$
- $\delta := 2(n - c - 2)q^c$

Finally, we can conclude that if $c \leq \frac{(2-\epsilon)\log n}{\log(1/q)}$, then

$$\Pr (C \text{ is a code}) = o \left(\binom{n}{c}^{-1}\right).$$
Using Suen

Fix $C$, $c := |C|$, and let $\mathcal{I}$ consist of the pairs of vertices of $V \setminus C$.

Let $i \sim j$ if and only if the vertex sets that they represent overlap. This is a dependency graph.

- $\mu := \binom{n-c}{2} q^c$
- $\Delta := 3 \binom{n-c}{3} \left(p^3 + (1 - p)^3\right)^c$
- $\delta := 2(n - c - 2) q^c$

Finally, we can conclude that if $c \leq \frac{(2-\epsilon) \log n}{\log(1/q)}$, then

$$\Pr (\exists \text{ a code of size } c) = o(1).$$
Existentialism

We can generalize the idea of a code, even when one does not exist.
Existentialism

We can generalize the idea of a code, even when one does not exist.

Construct a relation $R$ on the vertex set of a graph $G$. 
Existentialism

We can generalize the idea of a code, even when one does not exist.

Construct a relation $R$ on the vertex set of a graph $G$.

We say $x R y$ if $N[x] = N[y]$. 

So, if vertices have different closed neighborhoods, they can be distinguished. If the balls are the same, then we cannot distinguish them anyway.
Existentialism

We can generalize the idea of a code, even when one does not exist.

Construct a relation $R$ on the vertex set of a graph $G$.

We say $xRy$ if $N[x] = N[y]$.

It is easy to see that this is an equivalence relation on $V(G)$:
Existentialism

We say $x R y$ if $N[x] = N[y]$.

It is easy to see that this is an equivalence relation on $V(G)$:
Existentialism

We say $xRy$ if $N[x] = N[y]$.

It is easy to see that this is an equivalence relation on $V(G)$:

- **Reflexivity:**

So, if vertices have different closed neighborhoods, they can be distinguished. If the balls are the same, then we cannot distinguish them anyway.
Existentialism

We say $x R y$ if $N(x) = N(y)$.

It is easy to see that this is an equivalence relation on $V(G)$:

- **Reflexivity:**
  
  $N(u) = N(u)$. ✔️
Existentialism

We say $xRy$ if $N[x] = N[y]$.

It is easy to see that this is an equivalence relation on $V(G)$:

- **Reflexivity:**
  $N[u] = N[u].$

- **Symmetry:**
**Existentialism**

We say $x R y$ if $N[x] = N[y]$.

It is easy to see that this is an equivalence relation on $V(G)$:

- **Reflexivity:**

- **Symmetry:**
Existentialism

We say $xRy$ if $N[x] = N[y]$.

It is easy to see that this is an equivalence relation on $V(G)$:

- **Reflexivity:**

- **Symmetry:**

- **Transitivity:**
Existentialism

We say $x R y$ if $N[x] = N[y]$.

It is easy to see that this is an equivalence relation on $V(G)$:

- **Reflexivity:**

- **Symmetry:**

- **Transitivity:**
Existentialism

We say \( xRy \) if \( N[x] = N[y] \).

It is easy to see that this is an equivalence relation on \( V(G) \):

- **Reflexivity:** ✓
- **Symmetry:** ✓
- **Transitivity:** ✓
Existentialism

We say $x \mathbin{R} y$ if $N[x] = N[y]$.

It is easy to see that this is an equivalence relation on $V(G)$:

- **Reflexivity:** ✓
- **Symmetry:** ✓
- **Transitivity:** ✓

So, if vertices have different closed neighborhoods, they can be distinguished.
Existentialism

We say $x R y$ if $N[x] = N[y]$.

It is easy to see that this is an equivalence relation on $V(G)$:

- Reflexivity: ✓
- Symmetry: ✓
- Transitivity: ✓

So, if vertices have different balls, they can be distinguished.
Existentialism

We say $x R y$ if $N[x] = N[y]$.

It is easy to see that this is an equivalence relation on $V(G)$:

- **Reflexivity:** ✔
- **Symmetry:** ✔
- **Transitivity:** ✔

So, if vertices have different balls, they can be distinguished.

If the balls are the same, then we cannot distinguish them anyway.
Existentialism

We say $x R y$ if $N[x] = N[y]$.

It is easy to see that this is an equivalence relation on $V(G)$:

- **Reflexivity:** ✓
- **Symmetry:** ✓
- **Transitivity:** ✓

So, if vertices have different balls, they can be distinguished.

If the balls are the same, then we cannot distinguish them anyway, so why bother?
You get a stinkin’ code anyway

So, instead of a *vertex*-identifying code, we can get a *ball*-identifying code.
You get a stinkin’ code anyway

So, instead of a vertex-identifying code, we can get a ball-identifying code.

Such a code is always well-defined.
You get a stinkin’ code anyway

So, instead of a vertex-identifying code, we can get a ball-identifying code.

Such a code is always well-defined.

Although it enables us to get around the question of the existence of a code,
You get a stinkin’ code anyway

So, instead of a vertex-identifying code, we can get a ball-identifying code.

Such a code is always well-defined.

Although it enables us to get around the question of the existence of a code, it doesn’t seem to have any great practical value.
You get a stinkin’ code anyway

So, instead of a vertex-identifying code, we can get a ball-identifying code.

Such a code is always well-defined.

Although it enables us to get around the question of the existence of a code, it doesn’t seem to have any great practical value.

Since a code exists in the random graph $G_{n,p}$ for most reasonable values of $p$, 
You get a stinkin’ code anyway

So, instead of a vertex-identifying code, we can get a ball-identifying code.

Such a code is always well-defined.

Although it enables us to get around the question of the existence of a code, it doesn’t seem to have any great practical value.

Since a code exists in the random graph $G_{n,p}$ for most reasonable values of $p$, this is not a useful generalization because a vertex-identifying code already exists, with high probability.
A topology

If you like, this can be thought of as a tennis topology.
A topology

If you like, this can be thought of as a tennis topology.
A topology

If you like, this can be thought of as a point topology.
A topology

If you like, this can be thought of as a point-set topology.
A topology

If you like, this can be thought of as a point-set-match? topology.
A topology

If you like, this can be thought of as a point-set topology.

It is the discrete topology on the closed neighborhoods.
A topology

If you like, this can be thought of as a point-set topology.

It is the discrete topology on the closed neighborhoods.

It happens to be a so-called $T_1$ topology, in that for any pair of distinct neighborhoods,
A topology

If you like, this can be thought of as a point-set topology.

It is the discrete topology on the closed neighborhoods.

It happens to be a so-called $T_1$ topology, in that for any pair of distinct neighborhoods, there is an open set that contains one but not the other.
More failure

Suppose we can allow more than one node to fail.
More failure

Suppose we can allow more than one node to fail.

We will fix $\ell$ and try to distinguish between subsets of size at most $\ell$. 
More failure

Suppose we can allow more than one node to fail.

We will fix \( \ell \) and try to distinguish between subsets of size at most \( \ell \).

Recall \( N[v] = N(v) \cup \{v\} \).
More failure
Suppose we can allow more than one node to fail.

We will fix \( \ell \) and try to distinguish between subsets of size at most \( \ell \).

Recall \( N[v] = N(v) \cup \{v\} \). For \( S \subseteq V(G) \), let
More failure

Suppose we can allow more than one node to fail.

We will fix $\ell$ and try to distinguish between subsets of size at most $\ell$.

Recall $N[v] = N(v) \cup \{v\}$. For $S \subseteq V(G)$, let

$$N[S] = \bigcup_{v \in S} N[v].$$
More failure

We will fix $\ell$ and try to distinguish between subsets of size at most $\ell$.

Recall $N[v] = N(v) \cup \{v\}$. For $S \subseteq V(G)$, let

$$N[S] = \bigcup_{v \in S} N[v].$$

We say $C \subseteq V(G)$ is an $\ell$-identifying code if
More failure

We will fix $\ell$ and try to distinguish between subsets of size at most $\ell$.

Recall $N[v] = N(v) \cup \{v\}$. For $S \subseteq V(G)$, let

$$N[S] = \bigcup_{v \in S} N[v].$$

We say $C \subseteq V(G)$ is an $\ell$-identifying code if

$$N[S] \cap C \neq N[T] \cap C$$
Recall $N[v] = N(v) \cup \{v\}$. For $S \subseteq V(G)$, let

$$N[S] = \bigcup_{v \in S} N[v].$$

We say $C \subseteq V(G)$ is an \(\ell\)-identifying code if

$$N[S] \cap C \neq N[T] \cap C$$

for all distinct nonempty $S, T \subseteq V(G)$ with $|S|, |T| \leq \ell$. 
More code sizes

It is clear that every \((\ell + 1)\)-identifying code is an \(\ell\)-identifying code.
More code sizes

It is clear that every \((\ell + 1)\)-identifying code is an \(\ell\)-identifying code.

So, how large is the smallest \(\ell\)-identifying code in the random graph?
More code sizes

It is clear that every \((\ell + 1)\)-identifying code is an \(\ell\)-identifying code.

So, how large is the smallest \(\ell\)-identifying code in the random graph?

It is clearly at least as large as the smallest 1-identifying code.
More code sizes

It is clear that every \((\ell + 1)\)-identifying code is an \(\ell\)-identifying code.

So, how large is the smallest \(\ell\)-identifying code in the random graph?

It is clearly at least as large as the smallest 1-identifying code.

In the random graph, \(G_{n,p}\), for \(p\) constant, is it still \(\Theta(\log n)\)?
More random code sizes

For a graph $G$, define $c_\ell(G)$
More random code sizes

For a graph \( G \), define \( c_\ell(G) \) to be the size of a smallest \( \ell \)-identifying code in \( G \).

Theorem. [FMMRS]

Let \( 0 < p < 1 \).

\[ q^* \triangleq \min \left( p, 2p \left(1 - \frac{1}{p}\right) \right) \]

For any \( q > 0 \) and almost every graph \( G(n; p) \), we have

\[ c_\ell(G(n;p)) \leq 2(q + 1) \log n \log \left(1 + q^*\right) \]

Note that we may have \( c_\ell(G(n;p)) = O(\ln n) \); because \( \left(\log (1 + q^*)\right) = O(2q^*) \).
More random code sizes

For a graph $G$, define $c_\ell(G)$ to be the size of a smallest $\ell$-identifying code in $G$. (If none, $c_\ell(G) = \infty$.)
More random code sizes

For a graph $G$, define $c_\ell(G)$ to be the size of a smallest $\ell$-identifying code in $G$. (If none, $c_\ell(G) = \infty$.)

**Theorem.** [FMMRS] Let $0 < p < 1$. 
More random code sizes

For a graph $G$, define $c_\ell(G)$ to be the size of a smallest $\ell$-identifying code in $G$. (If none, $c_\ell(G) = \infty$.)

**Theorem.** [FMMRS] Let $0 < p < 1$.

$$q_\ell \overset{\text{def}}{=} 1 - \min\{p, 2p(1 - p)\}(1 - p)^{\ell-1}.$$
More random code sizes

For a graph $G$, define $c_\ell(G)$ to be the size of a smallest $\ell$-identifying code in $G$. (If none, $c_\ell(G) = \infty$.)

**Theorem.** [FMMRS] Let $0 < p < 1$.

\[
q_\ell \overset{\text{def}}{=} 1 - \min\{p, 2p(1 - p)\}(1 - p)^{\ell-1}.
\]

For any $\epsilon > 0$ and almost every graph in $G(n, p)$, we have
More random code sizes

For a graph $G$, define $c_{\ell}(G)$ to be the size of a smallest $\ell$-identifying code in $G$. (If none, $c_{\ell}(G) = \infty$.)

**Theorem.** [FMMRS] Let $0 < p < 1$.

$$q_{\ell} \overset{\text{def}}{=} 1 - \min\{p, 2p(1 - p)\}(1 - p)^{\ell-1}.$$ 

For any $\epsilon > 0$ and almost every graph in $G(n, p)$, we have

$$c_{\ell}(G_{n,p}) \leq \frac{2(\ell + \epsilon) \log n}{\log(1/q_{\ell})}.$$
More random code sizes

**Theorem. [FMMRS]** Let $0 < p < 1$.

$$q_\ell \overset{\text{def}}{=} 1 - \min\{p, 2p(1 - p)\}(1 - p)^{\ell - 1}.$$  

For any $\epsilon > 0$ and almost every graph in $G(n, p)$, we have

$$c_\ell (G_{n,p}) \leq \frac{2(\ell + \epsilon) \log n}{\log(1/q_\ell)}.$$  

Note that we may have

$$c_\ell (G_{n,p}) \neq O(\ell \ln n),$$
More random code sizes

**Theorem. [FMMRS]** Let $0 < p < 1$.

$$q_{\ell} \overset{\text{def}}{=} 1 - \min\{p, 2p(1 - p)\}(1 - p)^{\ell-1}.$$ 

For any $\epsilon > 0$ and almost every graph in $G(n, p)$, we have

$$c_{\ell}(G_{n, p}) \leq \frac{2(\ell + \epsilon) \log n}{\log(1 / q_{\ell})}.$$ 

Note that we may have

$$c_{\ell}(G_{n, p}) \neq O(\ell \ln n),$$

because $(\log(1 / q_{\ell}))^{-1} = O(2^\ell)$. 
More random code sizes

**Theorem. [FMMRS]** Let $0 < p < 1$.

$$q_\ell \overset{\text{def}}{=} 1 - \min\{p, 2p(1 - p)\}(1 - p)^{\ell - 1}.$$

For any $\epsilon > 0$ and almost every graph in $G(n, p)$, we have

$$c_\ell(G_{n,p}) \leq \frac{2(\ell + \epsilon) \log n}{\log(1/q_\ell)}.$$

It is true that

$$c_\ell(G_{n,p}) = O\left(\ell 2^\ell \ln n\right),$$

because $(\log(1/q_\ell))^{-1} = O\left(2^\ell\right)$.
Connections to matrices

**Given:** A graph $G$, $C \subseteq V(G)$. 
Connections to matrices

**Given:** A graph $G$, $C \subseteq V(G)$.

**Construct:** Matrix $M$, $|C| \times |V(G)|$. 

If $C$ is an `identifying code for $G$, then there is no zero column and the bitwise OR of each set of `columns is unique. This is an `superimposed code (UD`-code).
Connections to matrices

**Given:** A graph $G$, $C \subseteq V(G)$.

**Construct:** Matrix $M$, $|C| \times |V(G)|$.

$m_{ij} = \begin{cases} 1, & \text{if } c_i \sim v_j \text{ or } c_i = v_j; \\ 0, & \text{else.} \end{cases}$

If $C$ is an $\ell$-identifying code for $G$, then
Connections to matrices

**Given:** A graph \( G, C \subseteq V(G) \).

**Construct:** Matrix \( M, |C| \times |V(G)| \).

\[
m_{ij} = \begin{cases} 
1, & \text{if } c_i \sim v_j \text{ or } c_i = v_j; \\
0, & \text{else.}
\end{cases}
\]

If \( C \) is an \( \ell \)-identifying code for \( G \), then there is no zero column and
Connections to matrices

**Given:** A graph $G$, $C \subseteq V(G)$.

**Construct:** Matrix $M$, $|C| \times |V(G)|$.

$$m_{ij} = \begin{cases} 
1, & \text{if } c_i \sim v_j \text{ or } c_i = v_j; \\
0, & \text{else.}
\end{cases}$$

If $C$ is an $\ell$-identifying code for $G$, then there is no zero column and

The bitwise $\lor$ of each set of $\leq \ell$ columns is unique.
Connections to matrices

**Given:** A graph $G$, $C \subseteq V(G)$.

**Construct:** Matrix $M$, $|C| \times |V(G)|$.

$$m_{ij} = \begin{cases} 
1, & \text{if } c_i \sim v_j \text{ or } c_i = v_j; \\
0, & \text{else.}
\end{cases}$$

If $C$ is an $\ell$-identifying code for $G$, then there is no zero column and

The bitwise OR of each set of $\leq \ell$ columns is unique.

This is an $\ell$-superimposed code ($\text{UD}_\ell$-code).
Connections to matrices

**Given:** A graph $G$, $C \subseteq V(G)$.

**Construct:** Matrix $M$, $|C| \times |V(G)|$.

If $C$ is an $\ell$-identifying code for $G$, then there is no zero column and

The bitwise OR of each set of $\leq \ell$ columns is unique.

This is an $\ell$-superimposed code ($UD_\ell$-code).

- $|C|$ is the dimension, and
Connections to matrices

**Given:** A graph $G$, $C \subseteq V(G)$.

**Construct:** Matrix $M$, $|C| \times |V(G)|$.

If $C$ is an $\ell$-identifying code for $G$, then there is no zero column and

The bitwise OR of each set of $\leq \ell$ columns is unique.

This is an $\ell$-superimposed code ($\text{UD}_\ell$-code).

- $|C|$ is the dimension, and
- $|V(G)|$ is the cardinality.
Superimposed codes

Theorem. [D’yachkov-Rykov, Füredi-Ruszinkó, Csűrös-Rusz., Rusz.]

There exists a constant $a$ such that, in a space of dimension $N$, a code with cardinality $n$ satisfies

$$n \geq \frac{\exp(aN \log |G|)}{2^n}.$$
Superimposed codes

Theorem. [D’yachkov-Rykov, Füredi-Ruszinkó, Csűrös-Rusz., Rusz.]

There exists a constant $a$ such that, in a space of dimension $N$, a code with cardinality $n$ satisfies

$$n \leq \exp \left\{ a^{-1} N \frac{\log \ell}{\ell^2} \right\}.$$
Superimposed codes

Theorem. [D’yachkov-Rykov, Füredi-Ruszinkó, Csűrös-Rusz., Rusz., FMMRS]
There exists a constant $a$ such that, for an $\ell$-identifying code, $C$, of an $n$-vertex graph $G$,

$$n \leq \exp \left\{ a^{-1} |C| \frac{\log \ell}{\ell^2} \right\}.$$
Superimposed codes

Theorem. [D’yachkov-Rykov, Füredi-Ruszinkó, Csűrös-Rusz., Rusz., FMMRS]
There exists a constant \( a \) such that, for an \( \ell \)-identifying code, \( C \), of an \( n \)-vertex graph \( G \),

\[
a \frac{\ell^2}{\log \ell} \log n \leq |C|.
\]
Superimposed codes

Theorem. [D’yachkov-Rykov, Füredi-Ruszinkó, Csűrös-Rusz., Rusz., FMMRS]
There exists a constant $a$ such that, for an $\ell$-identifying code, $C$, of an $n$-vertex graph $G$,

$$\frac{\ell^2}{\log \ell} \log n \leq |C|.$$ 

Note that this lower bound holds for all graphs.
Superimposed codes

Theorem. [D’yachkov-Rykov, Füredi-Ruszinkó, Csűrös-Rusz., Rusz., FMMRS]
There exists a constant $a$ such that, for an $\ell$-identifying code, $C$, of an $n$-vertex graph $G$,

\[
\frac{\ell^2}{\log \ell} \log n \leq |C|.
\]

Note that this lower bound holds for all graphs, not just random graphs.
Superimposed codes

Theorem. [D’yachkov-Rykov, Füredi-Ruszinkó, Csűrös-Rusz., Rusz., FMMRS]

There exists a constant $a$ such that, for an $\ell$-identifying code, $C$, of an $n$-vertex graph $G$,

$$a \frac{\ell^2}{\log \ell} \log n \leq |C|.$$ 

Note that this lower bound holds for all graphs, not just random graphs.

Actually, $a = 1/8$ does the job, for $n$ sufficiently large.
Smallest possible code

Let $m_\ell(n)$ be the size of the smallest $\ell$-identifying code in an $n$-vertex graph.

**Theorem. [KCL]** Let $\ell \geq 2$. There exists an absolute constant $c_0$ such that

$$c_0 \ell \log n \leq m_\ell(n).$$
Smallest possible code

Let $m_\ell(n)$ be the size of the smallest $\ell$-identifying code in an $n$-vertex graph.

**Theorem. [FMMRS]** Let $\ell \geq 2$. There exist absolute constants $c_0, c_1, c_2$ such that

$$c_1 \frac{\ell^2}{\log \ell} \log n \leq m_\ell(n) \leq c_2 \ell^2 \log n.$$
Absurd generalization

In the literature, the notation is a bit different.
Absurd generalization

In the literature, the notation is a bit different.

- Vertex identifying codes are called (1, 1)-identifying codes.
Absurd generalization

In the literature, the notation is a bit different.

- Vertex identifying codes are called $(1, 1)$-identifying codes.

- For $\ell \geq 2$, $\ell$-identifying codes are called $(1, \leq \ell)$-identifying codes.

The first coordinate indicates the size of the balls that surround each vertex.
Absurd generalization

In the literature, the notation is a bit different.

- Vertex identifying codes are called \((1, 1)\)-identifying codes.
- For \(\ell \geq 2\), \(\ell\)-identifying codes are called \((1, \leq \ell)\)-identifying codes.

The first coordinate indicates the size of the balls that surround each vertex. I.e., second neighborhood, third neighborhood, etc.
Absurd generalization

In the literature, the notation is a bit different.

- For $\ell \geq 2$, $\ell$-identifying codes are called $(1, \leq \ell)$-identifying codes.

The first coordinate indicates the size of the balls that surround each vertex. I.e., second neighborhood, third neighborhood, etc.

Let $N_i(v)$ denote the $i^{th}$ neighborhood of $v$. 
Absurd generalization

In the literature, the notation is a bit different.

The first coordinate indicates the size of the balls that surround each vertex. I.e., second neighborhood, third neighborhood, etc.

Let $N_i(v)$ denote the $i^{th}$ neighborhood of $v$. 
Absurd generalization

In the literature, the notation is a bit different.

The first coordinate indicates the size of the balls that surround each vertex. I.e., second neighborhood, third neighborhood, etc.

Let $N_i(v)$ denote the $i^{th}$ neighborhood of $v$. For $S \subseteq V(G)$, define
Absurd generalization

In the literature, the notation is a bit different.

The first coordinate indicates the size of the balls that surround each vertex. I.e., second neighborhood, third neighborhood, etc.

Let $N_i(v)$ denote the $i^{th}$ neighborhood of $v$. For $S \subseteq V(G)$, define

$$B_k[S] = \bigcup_{i=0}^{k} N_i(v).$$
Absurd generalization

Let $N_i(v)$ denote the $i^{th}$ neighborhood of $v$. For $S \subseteq V(G)$, define

$$B_k[S] = \bigcup_{v \in S} \bigcup_{i=0}^{k} N_i(v).$$
Absurd generalization

Let $N_i(v)$ denote the $i^{th}$ neighborhood of $v$. For $S \subseteq V(G)$, define

$$B_k[S] = \bigcup_{v \in S} \bigcup_{i=0}^{k} N_i(v).$$

For positive integers $k$ and $\ell$, $C \subseteq V(G)$ is a $(\leq k, \leq \ell)$-identifying code if
Absurd generalization

Let $N_i(v)$ denote the $i^{th}$ neighborhood of $v$. For $S \subseteq V(G)$, define

$$B_k[S] = \bigcup_{v \in S} \bigcup_{i=0}^{k} N_i(v).$$

For positive integers $k$ and $\ell$, $C \subseteq V(G)$ is a $(\leq k, \leq \ell)$-identifying code if

$$B_{k'}[S] \cap C$$
Absurd generalization

Let $N_i(v)$ denote the $i^{th}$ neighborhood of $v$. For $S \subseteq V(G)$, define

$$B_k[S] = \bigcup_{v \in S} \bigcup_{i=0}^{k} N_i(v).$$

For positive integers $k$ and $\ell$, $C \subseteq V(G)$ is a $(\leq k, \leq \ell)$-identifying code if

$$B_{k'}[S] \cap C$$

is unique and nonempty for all pairs $(k', S)$. 

Vertex identifying codes and the random graph – p. 24/34
Absurd generalization

Let $N_i(v)$ denote the $i^{th}$ neighborhood of $v$. For $S \subseteq V(G)$, define

$$B_k[S] = \bigcup_{v \in S} \bigcup_{i=0}^{k} N_i(v).$$

For positive integers $k$ and $\ell$, $C \subseteq V(G)$ is a $(\leq k, \leq \ell)$-identifying code if

$$B_{k'}[S] \cap C$$

is unique and nonempty for all pairs $(k', S)$, where $1 \leq k' \leq k$. 
Absurd generalization

Let $N_i(v)$ denote the $i^{th}$ neighborhood of $v$. For $S \subseteq V(G)$, define

$$B_k[S] = \bigcup_{v \in S} \bigcup_{i=0}^{k} N_i(v).$$

For positive integers $k$ and $\ell$, $C \subseteq V(G)$ is a $(\leq k, \leq \ell)$-identifying code if

$$B_{k'}[S] \cap C$$

is unique and nonempty for all pairs $(k', S)$, where $1 \leq k' \leq k$; and all $S$. 
Absurd generalization

Let $N_i(v)$ denote the $i$th neighborhood of $v$. For $S \subseteq V(G)$, define

$$B_k[S] = \bigcup_{v \in S} \bigcup_{i=0}^{k} N_i(v).$$

For positive integers $k$ and $\ell$, $C \subseteq V(G)$ is a $(\leq k, \leq \ell)$-identifying code if

$$B_{k'}[S] \cap C$$

is unique and nonempty for all pairs $(k', S)$, where $1 \leq k' \leq k$; and all $S$, $\emptyset \neq S \subseteq V(G)$. 
Absurd generalization

Let $N_i(v)$ denote the $i^{th}$ neighborhood of $v$. For $S \subseteq V(G)$, define

$$B_k[S] = \bigcup_{v \in S} \bigcup_{i=0}^{k} N_i(v).$$

For positive integers $k$ and $\ell$, $C \subseteq V(G)$ is a $(\leq k, \leq \ell)$-identifying code if

$$B_{k'}[S] \cap C$$

is unique and nonempty for all pairs $(k', S)$, where $1 \leq k' \leq k$; and all $S$, $\emptyset \neq S \subseteq V(G)$, $|S| \leq \ell$. 
Smallest codes

We established that a lower bound for the basic code on an $n$-vertex graph was

$$\left\lfloor \log_2(n + 1) \right\rfloor.$$
Smallest codes

We established that a lower bound for the basic code on an $n$-vertex graph was

$$\lfloor \log_2(n + 1) \rfloor .$$

This lower bound can actually be achieved:
Smallest codes

- Take a subset $C$ of $\lceil \log_2(n + 1) \rceil$ vertices, make it independent.

$n = 8$ vertices
Smallest codes

- Take a subset $C$ of $\lceil \log_2(n + 1) \rceil$ vertices, make it independent.
- For $n - \lceil \log_2(n + 1) \rceil$ subsets $C' \subseteq C$ with $|C'| \geq 2$, connect a unique vertex only to the vertices in $C'$.

$n = 8$ vertices
Smallest codes

- Take a subset $C$ of $\lceil \log_2(n + 1) \rceil$ vertices, make it independent.
- For $n - \lceil \log_2(n + 1) \rceil$ subsets $C' \subseteq C$ with $|C'| \geq 2$, connect a unique vertex only to the vertices in $C'$.

$n = 9$ vertices
Smallest codes

- Take a subset $C$ of $\lceil \log_2(n + 1) \rceil$ vertices, make it independent.
- For $n - \lceil \log_2(n + 1) \rceil$ subsets $C' \subseteq C$ with $|C'| \geq 2$, connect a unique vertex only to the vertices in $C'$.

$n = 10$ vertices
Smallest codes

- Take a subset $C$ of $\lceil \log_2(n + 1) \rceil$ vertices, make it independent.
- For $n - \lceil \log_2(n + 1) \rceil$ subsets $C' \subseteq C$ with $|C'| \geq 2$, connect a unique vertex only to the vertices in $C'$.

$n = 11$ vertices
Smallest codes

- Take a subset $C$ of $\lceil \log_2(n + 1) \rceil$ vertices, make it independent.
- For $n - \lceil \log_2(n + 1) \rceil$ subsets $C' \subseteq C$ with $|C'| \geq 2$, connect a unique vertex only to the vertices in $C'$.

$n = 12$ vertices
Smallest codes

- Take a subset $C$ of $\lceil \log_2(n + 1) \rceil$ vertices, make it independent.
- For $n = \lceil \log_2(n + 1) \rceil$ subsets $C' \subseteq C$ with $|C''| \geq 2$, connect a unique vertex only to the vertices in $C'$.
Smallest codes

- Take a subset $C$ of $\lceil \log_2(n + 1) \rceil$ vertices, make it independent.
- For $n - \lceil \log_2(n + 1) \rceil$ subsets $C' \subseteq C$ with $|C''| \geq 2$, connect a unique vertex only to the vertices in $C'$.

$n = 14$ vertices
Smallest codes

- Take a subset $C$ of $\lceil \log_2(n + 1) \rceil$ vertices, make it independent.
- For $n - \lceil \log_2(n + 1) \rceil$ subsets $C' \subseteq C$ with $|C'| \geq 2$, connect a unique vertex only to the vertices in $C'$.

$n = 15$ vertices
Other codes

This leads us to ask what is the smallest $\ell$-identifying code in an $n$-vertex graph?
Other codes

This leads us to ask what is the smallest $\ell$-identifying code in an $n$-vertex graph?

This is a much harder problem and such a construction is not known, even for $\ell = 2$. 
Largest codes

The empty graph has a \((1, 1)\)-code and it is both unique and trivial."
Largest codes

The empty graph has a $(1, 1)$-code and it is both unique and trivial – of size $n$. 
Largest codes

The empty graph has a $(1, 1)$-code and it is both unique and trivial – of size $n$.

If $n$ is even, we can create a graph with codes only of size $n$ and $n - 1$. 
Largest codes

The empty graph has a \((1, 1)\)-code and it is both unique and trivial – of size \(n\).

If \(n\) is even, we can create a graph with codes only of size \(n\) and \(n - 1\).

\[
c(G) \geq n - 1.
\]
Largest codes

The empty graph has a $(1, 1)$-code and it is both unique and trivial – of size $n$.

If $n$ is even, we can create a graph with codes only of size $n$ and $n - 1$.

$v$ and $w$ cannot be distinguished.
Largest codes

The empty graph has a \((1, 1)\)-code and it is both unique and trivial – of size \(n\).

If \(n\) is even, we can create a graph with codes only of size \(n\) and \(n - 1\).

\(u\) and \(v\) cannot be distinguished.
Largest codes

The empty graph has a $(1, 1)$-code and it is both unique and trivial – of size $n$.

If $n$ is even, we can create a graph with codes only of size $n$ and $n - 1$.

Any set of size $n - 1$ is a code.
Largest codes

The empty graph has a \((1, 1)\)-code and it is both unique and trivial – of size \(n\).

If \(n\) is even, we can create a graph with codes only of size \(n\) and \(n - 1\).

\[
c(G) = n - 1.
\]
Grids

For infinite graphs, we cannot talk about the **size** of a code.
Grids

For infinite graphs, we cannot talk about the size of a code.

If the graph is locally finite, then we can talk about the density.
Let $G$ be a grid and $D(G)$ denote the minimum density of a code in $G$.

$D(T) = \frac{1}{4}$ [KCL]

The lower bound comes from the fact that $d$-regular graphs on $N$ vertices have, for every code, $C$, $|C| \leq N d + 2$.

Take the limit.

The upper bound comes from the following construction.
Let $G$ be a grid and $D(G)$ denote the infimum of the density of a code in $G$. 
Let $G$ be a grid and $D(G)$ denote the infimum of the density of a code in $G$.

$$D(T) = \frac{1}{4} \quad \text{[KCL]}$$
Let $G$ be a grid and $D(G)$ denote the infimum of the density of a code in $G$.

$$D(T) = \frac{1}{4} \quad \text{[KCL]}$$

- The lower bound comes from the fact that $d$-regular graphs on $N$ vertices have, for every code, $C$, ...
The lower bound comes from the fact that $d$-regular graphs on $N$ vertices have, for every code, $C$,
Triangular grid

\[ D(\mathbb{T}) = \frac{1}{4} \]  

The lower bound comes from the fact that \( d \)-regular graphs on \( N \) vertices have, for every code, \( C \),

\[ |C| \geq \frac{2N}{d + 2}. \]
The lower bound comes from the fact that $d$-regular graphs on $N$ vertices have, for every code, $C$,

$$D \geq \frac{2}{d + 2}.$$ 

Take the limit.

\[ D(\mathbb{T}) = \frac{1}{4} \quad \text{[KCL]} \]

- Triangular grid
Triangular grid

\[ D(\mathbb{T}) = \frac{1}{4} \]  

[1] The lower bound comes from the fact that \( d \)-regular graphs on \( N \) vertices have, for every code, \( C \),

\[ D \geq \frac{2}{6 + 2}. \]  

Take the limit.

Vertex identifying codes and the random graph – p. 29/34
The upper bound comes from the following construction.
The upper bound comes from the following construction.

\[ D(T) = \frac{1}{4} \]  

[KCL]
Square grid
Square grid

\[ \frac{15}{43} \leq D(\mathbb{Z}^2) \leq \frac{7}{20} \]  

[CHLZ2,CHLZ3]
Square grid

\[
\frac{15}{43} \leq D(\mathbb{Z}^2) \leq \frac{7}{20} \quad [\text{CHLZ2, CHLZ3}]
\]

- The lower bound comes from a complex argument.
The upper bound comes from a construction.

\[ \frac{15}{43} \leq D(\mathbb{Z}^2) \leq \frac{7}{20} \quad [\text{CHLZ2,CHLZ3}] \]
\[ \frac{15}{43} \leq D(\mathbb{Z}^2) \leq \frac{7}{20} \] [CHLZ2,CHLZ3]

- The upper bound comes from constructions.
Hexagonal grids
Hexagonal grids

The upper bound comes from a construction.

$$\frac{16}{39} \leq D(H) \leq \frac{3}{7}$$

[CHLZ4]
Hexagonal grids

\[
\frac{16}{39} \leq D(\mathbb{H}) \leq \frac{3}{7}
\]  

- The lower bound comes from a complex argument.

[CHLZ4]
The upper bound comes from a construction.

\[ \frac{16}{39} \leq D(\mathbb{H}) \leq \frac{3}{7} \]

[CHLZ4]
Hexagonal grids

\[ \frac{16}{39} \leq D(\mathbb{H}) \leq \frac{3}{7} \]

- The upper bound comes from constructions.

[CHLZ4]
Open grid questions

- The size of the square grid code is still open:

\[
\frac{15}{43} \leq D(\mathbb{Z}^2) \leq \frac{7}{20}
\]
Open grid questions

- The size of the square grid code is still open:

\[ 0.3488 \leq D(\mathbb{Z}^2) \leq 0.3500 \]
Open grid questions

- (To my knowledge) the size of the square grid code is still open:

\[ 0.3488 \leq D(\mathbb{Z}^2) \leq 0.3500 \]
Open grid questions

- (To my knowledge) the size of the square grid code is still open:

\[ 0.3488 \leq D(\mathbb{Z}^2) \leq 0.3500 \]

- The size of the hexagonal grid code is still open:

\[ \frac{16}{39} \leq D(\mathbb{H}) \leq \frac{3}{7} \]
Open grid questions

- (To my knowledge) the size of the square grid code is still open:

  \[ 0.3488 \leq D(\mathbb{Z}^2) \leq 0.3500 \]

- The size of the hexagonal grid code is still open:

  \[ 0.4103 \leq D(\mathbb{H}) \leq 0.4286 \]
Open grid questions

- (To my knowledge) the size of the square grid code is still open:

$$0.3488 \leq D(\mathbb{Z}^2) \leq 0.3500$$

- The size of the hexagonal grid code is (almost surely!) still open:

$$0.4103 \leq D(\mathbb{H}) \leq 0.4286$$
Questions and open problems

- One question to resolve is the value of this $m_\ell(n)$, the size of the smallest $\ell$-identifying code in an $n$-vertex graph.
Questions and open problems

- One question to resolve is the value of this $m_\ell(n)$, the size of the smallest $\ell$-identifying code in an $n$-vertex graph.

$$m_1(n) = \lceil \log_2(n + 1) \rceil$$
Questions and open problems

- One question to resolve is the value of this $m_\ell(n)$, the size of the smallest $\ell$-identifying code in an $n$-vertex graph.

$$m_1(n) = \lceil \log_2(n + 1) \rceil$$

$$c_1 \frac{\ell^2}{\log \ell} \log n \leq m_\ell(n) \leq c_2 \ell^2 \log n.$$
Questions and open problems

- One question to resolve is the value of this $m_\ell(n)$, the size of the smallest $\ell$-identifying code in an $n$-vertex graph.

$$m_1(n) = \lceil \log_2(n + 1) \rceil$$

$$c_1 \frac{\ell^2}{\log \ell} \log n \leq m_\ell(n) \leq c_2 \ell^2 \log n.$$

- What graphs have large codes?
Questions and open problems

- One question to resolve is the value of this \( m_\ell(n) \), the size of the smallest \( \ell \)-identifying code in an \( n \)-vertex graph.

\[
m_1(n) = \lceil \log_2(n + 1) \rceil
\]

\[
c_1 \frac{\ell^2}{\log \ell} \log n \leq m_\ell(n) \leq c_2 \ell^2 \log n.
\]

- What graphs have large (linear sized) codes?
Questions and open problems

- One question to resolve is the value of this $m_\ell(n)$, the size of the smallest $\ell$-identifying code in an $n$-vertex graph.

\[
m_1(n) = \lfloor \log_2(n + 1) \rfloor
\]

\[
c_1 \frac{\ell^2}{\log \ell} \log n \leq m_\ell(n) \leq c_2 \ell^2 \log n.
\]

- What graphs have large (linear sized) codes?

Can the only code be trivial, if $G$ is not empty?
Questions and open problems

- Another question is to find the distribution of the size of the smallest $\ell$-identifying code in the random graph.

$$c_1 \frac{\ell^2}{\log \ell} \log n \leq c_\ell (G_{n,p}) \leq \frac{2(\ell + \epsilon)}{\log(1/q_\ell)} \log n$$
Questions and open problems

Another question is to find the distribution of the size of the smallest $\ell$-identifying code in the random graph.

$$c_1 \frac{\ell^2}{\log \ell} \log n \leq c_\ell \left( G_{n, \frac{1}{2}} \right) \leq \frac{2(\ell + \epsilon)}{\log(1/(1 - 2^{-\ell}))} \log n$$
Questions and open problems

- Another question is to find the distribution of the size of the smallest $\ell$-identifying code in the random graph.

\[
c_1 \frac{\ell^2}{\log \ell} \log n \leq c_\ell \left( G_{n, \frac{1}{2}} \right) \leq \frac{2(\ell + \epsilon)}{\log(1/(1 - 2^{-\ell}))} \log n \leq (\ell + \epsilon)2^{\ell+1} \ln n
\]
Questions and open problems

- Another question is to find the distribution of the size of the smallest $\ell$-identifying code in the random graph.

\[
c_1 \frac{\ell^2}{\log \ell} \log n \leq c_\ell \left(G_{n, \frac{1}{2}}\right) \leq \frac{2(\ell + \epsilon) \log n}{\log(1/(1 - 2^{-\ell}))} \\
\leq (\ell + \epsilon)2^{\ell+1} \ln n
\]

- For what values of $p$ does an $\ell$-identifying code exist with high probability in $G_{n,p}$?
Thanks

Thank you for your indulgence.
Thanks

Thank you for your indulgence.

I’ll stop now.
Thanks

Thank you for your indulgence.

I’ll stop now.

(You’re welcome.)
Thanks

Thank you for your indulgence.

I’ll stop now.

(You’re welcome.)

The file for this talk is available online at my website:

http://orion.math.iastate.edu/rymartin

These slides were created by the Prosper document preparation system.