Channel Coding: Zero-error case

Information & Communication

Sander Bet & Ismani Nieuweboer

February 2015

Preface

We would like to thank Christian Schaffner for guiding us in the right direction with our presentation and this report, and of course for his excellently given introductory bachelor's course on information theory, Information & Communication. We sincerely hope that this report can be of possible use for future years to come, may it be for the course, or somewhere else.

Contents

1 Introduction

Zero-error channel coding is the subject of sending and receiving distinctive messages over a channel, without any chance of error occurring. This differs from the usual notion of asymptotic channel coding, in which the error involved in sending messages can be made arbitrarily small. [2] The amount of such messages we can send through a channel, per channel use, is the zero-error capacity of a channel. In the ideal case the channel would be noiseless, however, it is unrealistic to depend on the existence of such channels. It is unknown how to generally determine the zero-error capacity of a channel. Using graph theory we will explain why, and show an example of a channel with a known zero-error capacity.

2 Channels

In order to lay the foundations, we are first going to look at channels themselves. In Shannon's Communication System representation (figure 1) the channel is the middle node. It carries the signal from the transmitter to the receiver. Only the errors can occur in the channel, if that is the case the channel is noisy. That can be a problem when trying to send a distinctive message. To analyse this problem further we will define a channel.

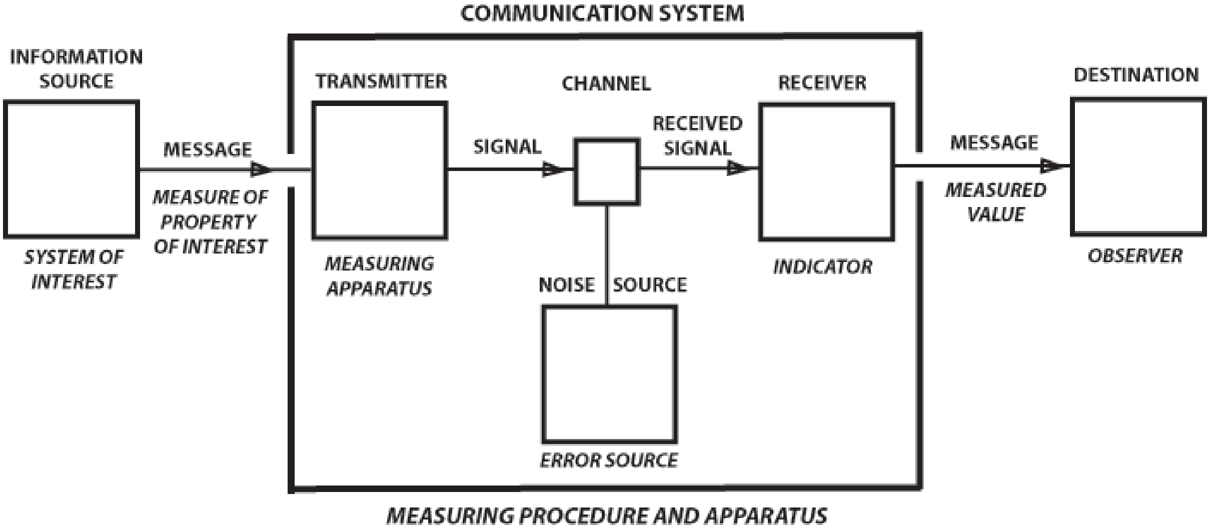


Figure 1: Communication System [5]

Definition 2.1. A *discrete channel* is denoted by $(\mathcal{X}, P_{Y|X}(y|x), \mathcal{Y})$ where $\mathcal{X}$ and $\mathcal{Y}$ are finite, non-empty sets and $P_{Y|X}(y|x)$ a conditional probability distribution with $x \in \mathcal{X}, y \in \mathcal{Y}$ that satisfies:

- $P_{Y|X}(y|x) \geq 0 \qquad : \forall x \in \mathcal{X}, \forall y \in \mathcal{Y}$
- $\sum_{y \in \mathcal{Y}} P_{Y|X}(y|x) = 1 \qquad : \forall x \in \mathcal{X}$

$\mathcal{X}$ is also referred to as the *input set*, and $\mathcal{Y}$ as the *output set*.

Definition 2.2. A *memory-less channel* is a channel where the probability distribution $P_{Y|X}(y|x)$ is *independent* of previous channel inputs and outputs.

Example 2.3. The discrete channel $(\mathcal{X}, P_{Y|X}(y|x), \mathcal{Y})$ with $\mathcal{X} = \{0, 1\}$, $\mathcal{Y} = \{2, 3\}$ and

$$P_{Y|X}(y|x) = \begin{cases} p & \text{if } (y|x) = (2|0) \text{ or } (y|x) = (3|1) \\ 1-p & \text{if } (y|x) = (2|1) \text{ or } (y|x) = (3|0) \end{cases}$$

is represented in figure 2, for $p \in [0, 1]$.

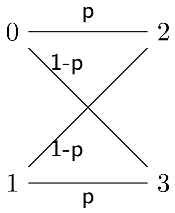


Figure 2: Visual representation of a discrete channel

2.1 Multiple Channel Usage

We will now refine the concept of a channel by looking at multiple usage. We will provide the definitions and prove some theorems.

Definition 2.4. *Multiple use of a discrete memory-less channel* $(\mathcal{X}, P_{Y|X}(y|x), \mathcal{Y})$ are denoted with $(\mathcal{X}^n, P_{\mathcal{Y}^n|\mathcal{X}^n}(y|x), \mathcal{Y}^n)$ where $\mathcal{X}^n = \{(x_1, x_2, \ldots, x_n) : x_i \in \mathcal{X}, 1 \leq i \leq n\}$, $\mathcal{Y}^n = \{(y_1, y_2, \ldots, y_n) : y_i \in \mathcal{Y}, 1 \leq i \leq n\}$ and $P_{\mathcal{Y}^n|\mathcal{X}^n}(y|x) = \prod_{i=1}^n P_{\mathcal{Y}|\mathcal{X}}(y_i|x_i)$ with $x \in \mathcal{X}^n, y \in \mathcal{Y}^n$.

Lemma 2.5. $\mathcal{X}^n$ and $\mathcal{Y}^n$ are non-empty finite sets.

Proof. $\mathcal{X}$ is non-empty and finite, so $\exists m \in \mathbb{N}$ such that $|\mathcal{X}| = m$. Because of the way, we defined $\mathcal{X}^n$ it is the same as n times Cartesian product of $\mathcal{X}$. Hence $\mathcal{X}^n = \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n$. $|\mathcal{X}^n| = |\mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n| = |\mathcal{X}_1| \times |\mathcal{X}_2| \times \ldots \times |\mathcal{X}_n| = m^n$. Now we see $\exists k \in \mathbb{N}, k = m^n, |\mathcal{X}^n| = k$. Hence $\mathcal{X}^n$ is a non-empty finite set. The proof for $\mathcal{Y}^n$ is analogous. □

Lemma 2.6. The probability function of multiple uses for a discrete memory-less channel satisfies:

$$P_{\mathcal{Y}^n|\mathcal{X}^n}(y|x) \geq 0 : \forall x \in \mathcal{X}^n, \forall y \in \mathcal{Y}^n, \forall n \in \mathbb{N}$$

Proof. The probability function of multiple uses for a discrete memory-less channel is defined as:

$$P_{\mathcal{Y}^n|\mathcal{X}^n}(y|x) = \prod_{i=1}^n P_{\mathcal{Y}|\mathcal{X}}(y_i|x_i) : x \in \mathcal{X}^n, y \in \mathcal{Y}^n$$

By definition of a discrete channel the probability function of a discrete channel satisfies:

$$P_{Y|X}(y|x) \geq 0 : \forall x \in \mathcal{X}, \forall y \in \mathcal{Y}$$

When we combine them we see that the probability function of multiple uses of a discrete memory-less channel satisfies:

$$\prod_{i=1}^n P_{\mathcal{Y}|\mathcal{X}}(y_i|x_i) \geq \prod_{i=1}^n 0 = 0$$

Hence

$$P_{\mathcal{Y}^n|\mathcal{X}^n}(y|x) \geq 0 : \forall x \in \mathcal{X}^n, \forall y \in \mathcal{Y}^n$$

□

Lemma 2.7. The probability function of multiple uses of a discrete memory-less channel satisfies the following condition:

$$\sum_{y \in \mathcal{Y}^n} \prod_{i=1}^n P_{\mathcal{Y}|\mathcal{X}}(y_i|x_i) = \prod_{i=1}^n \sum_{y_i \in \mathcal{Y}} P_{\mathcal{Y}|\mathcal{X}}(y_i|x_i) : \forall n \in \mathbb{N}$$

Proof. To prove the lemma we use induction. For n=1 we see that the condition holds, because

$$\sum_{y\in\mathcal{Y}^1}\prod_{i=1}^{1}P_{\mathcal{Y}|\mathcal{X}}(y_1|x_1)=\sum_{y\in\mathcal{Y}^1}P_{\mathcal{Y}|\mathcal{X}}(y_1|x_1)=\sum_{y_1\in\mathcal{Y}}P_{\mathcal{Y}|\mathcal{X}}(y_1|x_1)=\prod_{i=1}^{1}\sum_{y_1\in\mathcal{Y}}P_{\mathcal{Y}|\mathcal{X}}(y_1|x_1).$$

Now we prove the statement for $n+1$;

$$\begin{aligned}\sum_{y\in\mathcal{Y}^{n+1}}\prod_{i=1}^{n+1}P_{\mathcal{Y}|\mathcal{X}}(y_i|x_i)&=(\sum_{y\in\mathcal{Y}^n}\prod_{i=1}^{n}P_{\mathcal{Y}|\mathcal{X}}(y_i|x_i))(\sum_{y_{n+1}\in\mathcal{Y}}P_{\mathcal{Y}|\mathcal{X}}(y_{n+1}|x_{n+1}))\\&\overset{\text{IH}}{=}(\prod_{i=1}^{n}\sum_{y_i\in\mathcal{Y}}P_{\mathcal{Y}|\mathcal{X}}(y_i|x_i))(\sum_{y_{n+1}\in\mathcal{Y}}P_{\mathcal{Y}|\mathcal{X}}(y_{n+1}|x_{n+1}))\\&=\prod_{i=1}^{n+1}\sum_{y_i\in\mathcal{Y}}P_{\mathcal{Y}|\mathcal{X}}(y_i|x_i)\end{aligned}$$

Hence

$$\sum_{y\in\mathcal{Y}^n}\prod_{i=1}^{n}P_{\mathcal{Y}|\mathcal{X}}(y_i|x_i)=\prod_{i=1}^{n}\sum_{y_i\in\mathcal{Y}}P_{\mathcal{Y}|\mathcal{X}}(y_i|x_i):\forall n\in\mathbb{N}$$

□

Lemma 2.8. The probability function of multiple uses of a discrete memory-less channel satisfies the following condition:

$$\sum_{y\in\mathcal{Y}^n}\prod_{i=1}^{n}P_{\mathcal{Y}|\mathcal{X}}(y_i|x_i)=1:\forall x\in\mathcal{X}^n,\forall n\in\mathbb{N}$$

Proof. By the definition of the probability function of multiple uses of a discrete memory-less channel the following equation holds:

$$\sum_{y\in\mathcal{Y}^n}(P_{\mathcal{Y}^n|\mathcal{X}^n}(y|x))=\sum_{y\in\mathcal{Y}^n}(\prod_{i=1}^{n}P_{\mathcal{Y}|\mathcal{X}}(y_i|x_i))$$

By definition of a discrete channel the probability function of a discrete channel satisfies:

$$\sum_{y\in\mathcal{Y}}P_{Y|X}(y|x)=1:\forall x\in\mathcal{X}$$

Combined with lemma 2.7 it follows that:

$$\sum_{y\in\mathcal{Y}^n}\prod_{i=1}^{n}P_{\mathcal{Y}|\mathcal{X}}(y_i|x_i)=\prod_{i=1}^{n}\sum_{y_i\in\mathcal{Y}}P_{\mathcal{Y}|\mathcal{X}}(y_i|x_i)=\prod_{i=1}^{n}1=1$$

Hence

$$\sum_{y\in\mathcal{Y}^n}\prod_{i=1}^{n}P_{\mathcal{Y}|\mathcal{X}}(y_i|x_i)=1:\forall x\in\mathcal{X}^n,\forall n\in\mathbb{N}$$

□

Theorem 2.9. Multiple uses of a discrete memory-less channel $(\mathcal{X}^n,P_{\mathcal{Y}^n|\mathcal{X}^n}(y|x),\mathcal{Y}^n)$ is also a discrete channel.

Proof. The multiple uses of a discrete memory-less channel satisfies all the condition of the discrete channel as proven in Lemma 2.5, 2.6 and 2.8. □

2.2 (M, n)-codes

Definition 2.10. A (M, n)-*code* for a channel $(\mathcal{X}^n, P_{\mathcal{Y}^n|\mathcal{X}^n}(y|x), \mathcal{Y}^n)$ is:

- A message index set $\{1, 2, \ldots, M\}$
- An encoding function e: $\{1, 2, \ldots, M\} \to \mathcal{X}^n$
- A decoding function d: $\mathcal{Y}^n \to \{1, 2, \ldots, M\}$

Remark 2.10.1. Until now we have only seen examples of channels with $\mathcal{X}^1$ and $\mathcal{Y}^1$, obviously the corresponding codes to these channels are $(M, 1)$-codes.

The message index set will be an index to the messages that can be sent over a channel. In a zero-error (M, n)-code however, it will be an index to all the distinctive messages we are able to send across the channel. The decoding function has to assign the right message to a received signal, while the encoding function will return the input that corresponds to a given index. Note that the encoding function in the general case does not have to be surjective, but has to be in our setting of zero-error codes; otherwise, our definition of the bitrate of the code for the channel will be off.

As a last global definition needed for zero-error codes, we can now give an expression for the amount of bits sendable through a multiple-use channel:

Definition 2.11. Suppose we have a multiple-use discrete memoryless channel $(\mathcal{X}^n, P_{\mathcal{Y}^n|\mathcal{X}^n}(y|x), \mathcal{Y}^n)$ and a corresponding (M, n) code. Then the *transmission rate* of this channel is defined as $R = \frac{\log M}{n}$.

Example 2.12. Consider the discrete channel in figure 3.

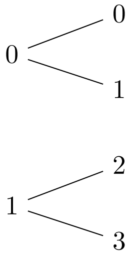


Figure 3: A discrete channel without noise

An M-code for this channel would be the code with:

- message index set $\{1, 2\}$
- $d(x) = \begin{cases} 1 & \text{if } x \in \{0, 1\} \\ 2 & \text{if } x \in \{2, 3\} \end{cases}$
- $e(x) = \begin{cases} 0 & \text{if } x = 1 \\ 1 & \text{if } x = 2 \end{cases}$

The transmission rate of this code is $\frac{\log(M)}{1} = \log(2) = 1$.

In zero-error channel theory, we are interested in the maximum amount of information we can send through the channel. In order to do that we found that we need distinctive messages.

The problem is, how do we know that our M-code is the best possible code? The best possible code would be the code that has the most distinctive messages. So, if we can maximize the number of distinctive messages, we know that we have the best possible code. How can we maximise the distinctive messages? We look at all the possible ones and pick the ones that cannot be confused.

3 Graph Theory

To be able to pick messages that cannot be confused with each other, graph theory will be used. This section will go over the basic notions needed when one wants to analyze a channel for the maximum amount of distinctive messages available.

Definition 3.1. A *graph* G is defined by a set of *vertices* $V(G)$, and a set of *edges* $E(G)$. An edge is denoted by $xy := \{x, y\}$ with x and y two distinct vertices from $V(G)$; more abstractly, $E(G) \subseteq \{xy : x, y \in V(G) \wedge x \neq y\}$.

The previous may vary across literature, but in our case we do not allow edges to go from a vertex to itself. Some definitions also mention *directed* graphs in which arrows are used instead of edges. As a last note, other terms include *nodes* and *points* for vertices, and *lines* for edges.

Example 3.2. Consider the complete graph with 5 vertices in figure 4.

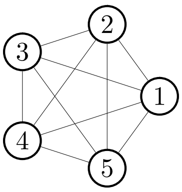


Figure 4: A complete graph with 5 vertices

Here, $V = \{1, 2, 3, 4, 5\}$ and $E = \{12, 13, 14, \ldots, 45\}$.

Definition 3.3. For a graph G, An *independent set* is a set of vertices $I \subseteq V(G)$ such that no edge $xy \in E(G)$ contains two vertices from I

Example 3.4. Consider the following graph in figure 5.

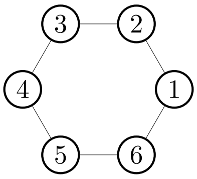


Figure 5: A cycle graph with 6 vertices

I can be $\{1\}, \{2\}, \{1, 3\}, \{1, 3, 5\}$, et cetera; a simple counterexample would be $\{1, 2\}$.

Definition 3.5. For a graph G, the *independence number* $\alpha(G)$ is the cardinality of the maximum independent set.

Since finding the maximum independent set is an NP-hard problem [6], it is unlikely to find an efficient algorithm for finding the independence number. This will have implications for finding zero-error codes given a channel, as we will see later on.

Example 3.6. Consider the graph from the previous example, with alternating nodes colored for clarity, in figure 6.

Looking at the black and red colored nodes, we find that $\alpha(G) = |\{1, 3, 5\}| = |\{2, 4, 6\}| = 3$. A pidgeonhole argument shows why this is true; since each vertex is connected by two edges, choosing four vertices results in $2 \cdot 4 = 8$ connections. This disregards that these vertices might already share an edge, but in that case the chosen vertices already do not form an independent set. Since there are only 6 edges in this graph, by the pidgeonhole principle there have to be at least two edges that are shared by the vertices. This shows that any vertex set with 4 or more edges cannot be an independent set.

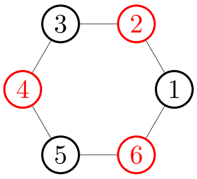


Figure 6: Cycle graph with independent sets emphasized

Note that in general it might not be possible at all to construct such an argument. In that case, one has to go across all possible subsets of the vertex set to verify that there is no bigger independent set. This gives an indication towards the difficulty of finding independent sets and the independence number.

4 Confusability Graphs

To find out which messages are not distinct, or, confusable with each other, a formal definition is needed.

Definition 4.1. Two input messages x_1 and x_2 from $\mathcal{X}$ are *confusable* if and only if there exists an output message $y \in \mathcal{Y}$ such that $P_{Y|X}(y|x_1) \neq 0$ and $P_{Y|X}(y|x_2) \neq 0$.

An input message is always confusable with itself, but since we do not allow any loops (edges from a vertex to itself) this will not be a problem.

Of course, this definition can also be extended to multiple usage of a channel:

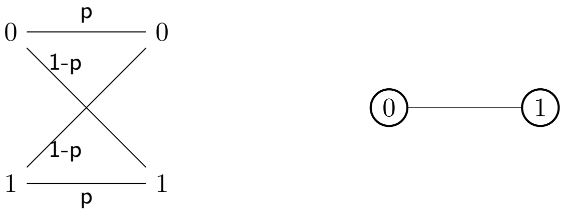
Definition 4.2. Two input messages $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_n)$ from $\mathcal{X}^n$ are *confusable* if and only if it holds for all $i \in \{1, 2, \dots, n\}$ that $x_i = y_i$ or x_i is confusable with y_i

Now, to be able to extend the notion of confusability to the entire channel, we define the confusability graph of a channel; this will be the tool to decide whether a code is zero-error or not.

Definition 4.3. Given a discrete memoryless channel $(\mathcal{X}, P_{Y|X}(y|x), \mathcal{Y})$, the confusability graph G is defined by $V(G) = \mathcal{X}$ and $E(G) = \{vw : v \text{ is confusable with } w\}$

As can be seen from the definition, two vertices representing two input messages are connected if and only if they can be confused with each other.

Example 4.4. Consider the following channel on the left with its corresponding confusability graph on the right, with $p \in (0, 1)$:



Note that for a zero-error application, this channel is unusable. Since both input messages can come out as either a zero or one, there is no way to receive a bit without having some uncertainty about what was originally sent. Another way to see this is to look at the independence number of the confusability graph. Seeing that $\alpha(G) = 1$ and $\log(1) = 0$, one can send no bits of zero-error information across this channel.

5 Zero-error codes

We will make use of confusability graphs to decide whether a code is zero-error or not. Note that we deliberately only defined the confusability graph for a single use of a channel, since doing otherwise would be overcomplicated.

Definition 5.1. For a discrete memoryless channel $(\mathcal{X}, P_{Y|X}(y|x), \mathcal{Y})$, an $(M,1)$-code for this channel is *zero-error* if and only if in the confusability graph G of the channel, the image of the encoding function $e[\{1,2,\ldots,M\}]$ is an independent set.

It is easy to see that this definition is equivalent to requiring that no two input messages are confusable with each other. Since we have not defined confusability graphs for multiple usage, one can as well take this as the definition:

Definition 5.2. For a discrete memoryless channel $(\mathcal{X}^n, P_{\mathcal{Y}^n|\mathcal{X}^n}(y|x), \mathcal{Y}^n)$, an (M,n)-code for this channel is *zero-error* if and only if no two messages $e(x)$ and $e(y)$ with x, y from the message set $\{1,2,\ldots,M\}$ are confusable.

Finally having the basic notion of a zero-error code, the natural question to ask would be what the maximum zero-error code for a channel is. For this, we use the last definition of section 2.2 and require the code to be without error:

Definition 5.3. Suppose we have a multiple-use discrete memoryless channel $(\mathcal{X}^n, P_{\mathcal{Y}^n|\mathcal{X}^n}(y|x), \mathcal{Y}^n)$ and a corresponding *zero-error* (M,n) code. Then the *zero-error rate* of this channel is defined as $R = \frac{\log M}{n}$.

To be able to actually use this rate for multiple usage, one needs to also define the confusability graph for a channel with multiple usage, which is relevant but not very instructive for our needs. Because of this, we will go out of our way and only derive the maximum zero-error rate of a channel for a single use. But, keeping the definition of the rate for general n available, we will still be able to show an interesting example.

Definition 5.4. Given a discrete memoryless channel $(\mathcal{X}, P_{Y|X}(y|x), \mathcal{Y})$. Let G be the confusability graph of this channel, and $\alpha(G)$ the independence number of this graph. Then the *zero-error capacity* for a single use is given by $C_0 = \log \alpha(G)$.

The intuition behind this concept follows from the definition of a zero-error code. A code can only be zero-error if the input messages used from the channel form an independent set within the confusability graph, and the rate is defined as $R = \frac{\log M}{n} = \log M$ (for $n = 1$); the only way to maximize the rate is to find the maximum independent set, which implies $M = \alpha(G)$.

The formula for the more general case looks similar, plugging in the confusability graph for the corresponding multiple-use channel, and dividing the whole by n as in the definition of the rate.

Since finding the independence number of a graph is difficult, it is also difficult to determine the zero-error capacity of a graph. Up to a certain size one can try to examine channels and their corresponding confusability graphs for the independence number. But since the amount of possible graphs and the difficulty of finding independent sets only increases with the size, this obviously cannot be done for all graphs.

5.1 Noisy typewriter

We will now discuss an example that will make use of all the previous concepts, and also show why it is difficult to find the zero-error capacity of a multiple-use channel.

The channel in question is also nicknamed the *noisy typewriter* (as portrayed in figure 7), due to the corresponding behavior; pressing a key on a typewriter could induce the outcome of different letter than intended.

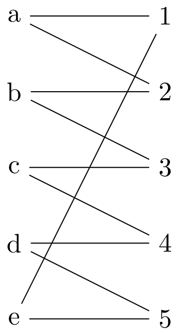


Figure 7: "Noisy typewriter" channel

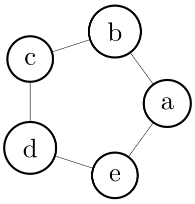


Figure 8: Confusability graph for the noisy typewriter

Now we consider the confusability graph of this channel in figure 8.

Using a pidgeonhole or exhaustion argument from section 3.6, one can deduce that the independence number of this graph is $\alpha(G) = 2$.

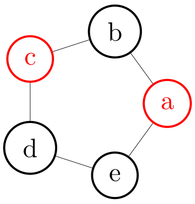


Figure 9: A possible choice for the independent set

We can therefore choose our index set for the $(M, 1)$ code to be $\{1, 2\}$. If we choose a and c as the messages for our code, our encoding and decoding function might look like this:

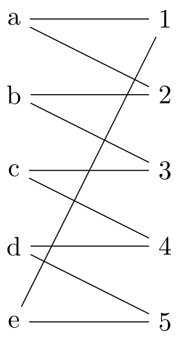
- $e(x) = \begin{cases} 1 & \text{if } x = \text{a} \\ 2 & \text{if } x = \text{b} \end{cases}$
- $d(x) = \begin{cases} \text{a} & \text{if } x \in \{1, 2\} \\ \text{b} & \text{if } x \in \{3, 4\} \end{cases}$

As a result, $R = \frac{\log(2)}{1} = 1$ bit.

Now, one would not expect any difference for $n = 2$; indeed it is possible to choose $\{aa, ac, ca, cc\}$, and due to the channel being independent of previous usage, this is also a zero-error code with $R = \frac{\log M}{n} = \frac{\log 4}{2} = 1$.

But, looking further into the possibilities when $n = 2$, it turns out that the set $\{aa, bc, ce, db, ed\}$ can also be used for a valid zero-error code. We will prove this by writing out all possible outcomes for these codes, in figure 10.

As all of these sets are disjoint, none of these messages can be confused with each other; suppose two messages would be confusable, then an output message would need to occur in two different sets. One



$$aa \to \{11, 12, 21, 22\}$$
$$bc \to \{23, 24, 33, 34\}$$
$$ce \to \{35, 31, 45, 41\}$$
$$db \to \{42, 43, 52, 53\}$$
$$ed \to \{54, 55, 14, 15\}$$

Figure 10: Noisy typewriter with possible outcome messages for the input messages

can also assign a message set to these codes and check to see that the definition of a zero-error code holds.

Since we now have $M = 5$, it follows that $R = \frac{\log 5}{2}$ bits. This is clearly an improvement on the previous rate, since $\frac{\log 5}{2} > \frac{\log 4}{2} = 1$. Now, one could ask whether the rate keeps improving over multiple uses; it turns out that this rate is the upper bound for this particular channel for all amounts of multiple use, as proved by Lovasz[7]. Since this code was found by Shannon[1], giving a lower bound for the zero-error capacity, it follows that the zero-error capacity, for all $n \in \mathbb{N}$, is equal to $\frac{log(5)}{2}$.

6 Conclusion

We have tried to convey the concepts related to, and including that of zero-error codes for discrete channels. One thing that stands out is that even for such a seemingly simple problem, there are still a lot of open questions. On first hand it is even more bizarre that there is not even a general way to determine the maximum zero-error rate for any given channel. But we hope that through the connections made to graph theory, we have at least given off an idea of where the difficulties of finding zero-error codes lie.

References

[1] Shannon, C.E., *The zero-error capacity of a noisy channel*, Research Paper, Massachusetts Institute of Technology, Cambridge.

[2] MacKay, D.J.C., *Information Theory, Inference, and Learning Algorithms* - `http://www.inference.phy.cam.ac.uk/mackay/itila/book.html`

[3] Körner, J. & Orlitsky, A., *Zero-error information theory*, Survey, June 24 1998.

[4] Schaffner, C., Blackboard photos, Master Course Information Theory 2014.

[5] Figure 1: *Communication System*, Picture,
http://www.informationphilosopher.com/solutions/scientists/rothstein/

[6] Alekseev, V.E., *The Maximum Independent Set Problem in Planar Graphs* - `http://link.springer.com/chapter/10.1007%2F978-3-540-85238-4_7`

[7] Lovasz, L., *On the Shannon capacity of a graph* - `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=1055985`