Next Up Previous

Next: Block codes -- the **Up:** Error correcting codes for **Previous:** Error correcting codes for

Repetition codes

A straightforward idea is to repeat every bit of the message some prearranged number of times, for example, three times, as shown in figure 1.4.

Source sequence $\mathbf{s}$	Transmitted sequence $\mathbf{t}$
0	000
1	111

Figure: The repetition code `R3'.

Let us imagine what happens if we transmit the source message $\mathbf{s} = (0\ 0\ 1\ 0\ 1\ 1\ 0)$ over a binary symmetric channel with noise level f=0.1 using this repetition code. We can describe the channel as adding a sparse noise vector $\mathbf{n}$ to the transmitted vector (in modulo 2 arithmetic, *i.e.*, the binary algebra in which `1`+`1`=`0`).

$$\begin{array}{c|ccccccc}
\mathbf{s} & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
\mathbf{t} & \overbrace{000} & \overbrace{000} & \overbrace{111} & \overbrace{000} & \overbrace{111} & \overbrace{111} & \overbrace{000} \\
\mathbf{n} & 000 & 001 & 000 & 000 & 101 & 000 & 000 \\
\hline
\mathbf{r} & 000 & 001 & 111 & 000 & 010 & 111 & 000
\end{array}$$

How do we decode this received vector? Clearly, the optimal algorithm looks at the received bits three at a time and takes a *majority vote*. The majority vote decoder is shown in figure 1.5. If all bits are 0, we decode the triplet as a 0. If, as in the second triplet, there are two `0`s and one `1`, we decode the triplet as a 0 -- which in this case corrects the error.

Received sequence $\mathbf{r}$	Decoded sequence $\hat{\mathbf{s}}$
000	0
001	0
010	0
100	0
101	1
110	1
011	1
111	1

Figure: Decoding algorithm for R3.

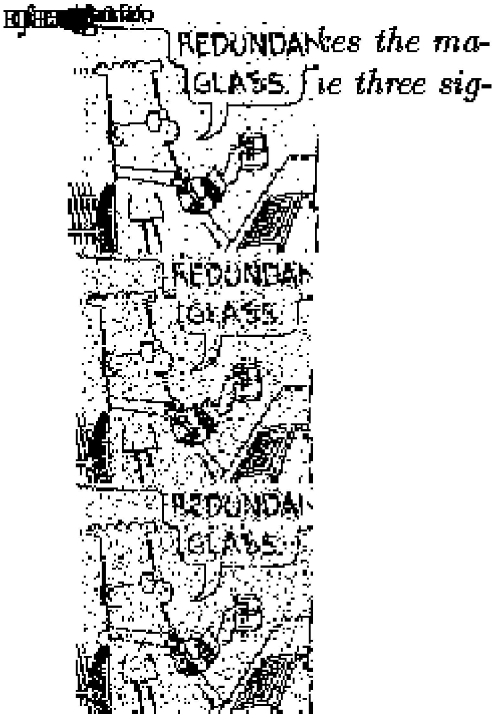
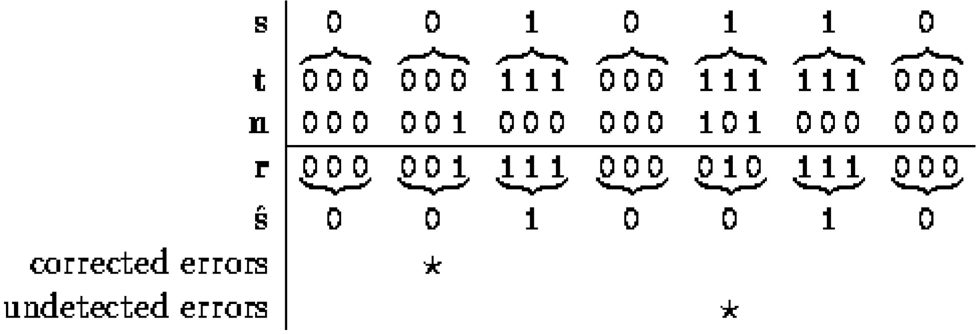


Figure: Transmitting 10000 source bits over a binary symmetric channel with *f*=0.1 using a repetition code. The probability of decoded bit error has fallen to about 3%; the rate has fallen to 1/3. [Dilbert image Copyright©1997 United Feature Syndicate, Inc., used with permission.]

Not all errors are corrected, however. If we are unlucky and two errors fall in a single block, as in the fifth triplet, then the decoding rule gets the wrong answer.



s	0	0	1	0	1	1	0
t	000	000	111	000	111	111	000
n	000	001	000	000	101	000	000
r	000	001	111	000	010	111	000
ŝ	0	0	1	0	0	1	0
corrected errors		⋆					
undetected errors					⋆		

Thus the error probability is reduced by the use of this code. It is easy to compute the error probability.

Compute the error probability of R3 for a binary symmetric channel with noise level *f*.

Clearly the error probability is dominated by the probability of two bits in a block of three being flipped, which scales as f^2. In the case of our binary symmetric channel with *f*=0.1, the R3 code has a probability of error, after decoding, of $p_b = 0.03$ per bit. Figure 1.6 shows the result of transmitting a binary image over a binary symmetric channel using the repetition code.

The repetition code R3 has therefore improved our probability of error, as desired. At the same time, we have lost something: our *rate* of information transfer has reduced by a factor of three. So if we use a repetition code to communicate over a telephone line, it will reduce the error rate, but it will also reduce our communication rate. We will have to pay three times as much for each phone call. As for our disc drive, we will need three noisy gigabyte disc drives in order to create a single gigabyte disc drive with $p_b = 0.03$.

Can we push the error probability lower to the sort of values required from a quality disk drive (10^{-15})? Well, we can achieve lower error probabilities by using repetition codes with more repetitions.

Show that it takes a repetition code with rate about 1/60 to get the probability of error down to 10^{-15}. This means that to build a *single* gigabyte disk drive with the required reliability from noisy gigabyte drives with f=0.1, we would need *sixty* of the noisy disk drives. The tradeoff between error probability and rate for repetition codes is shown in figure 1.7.

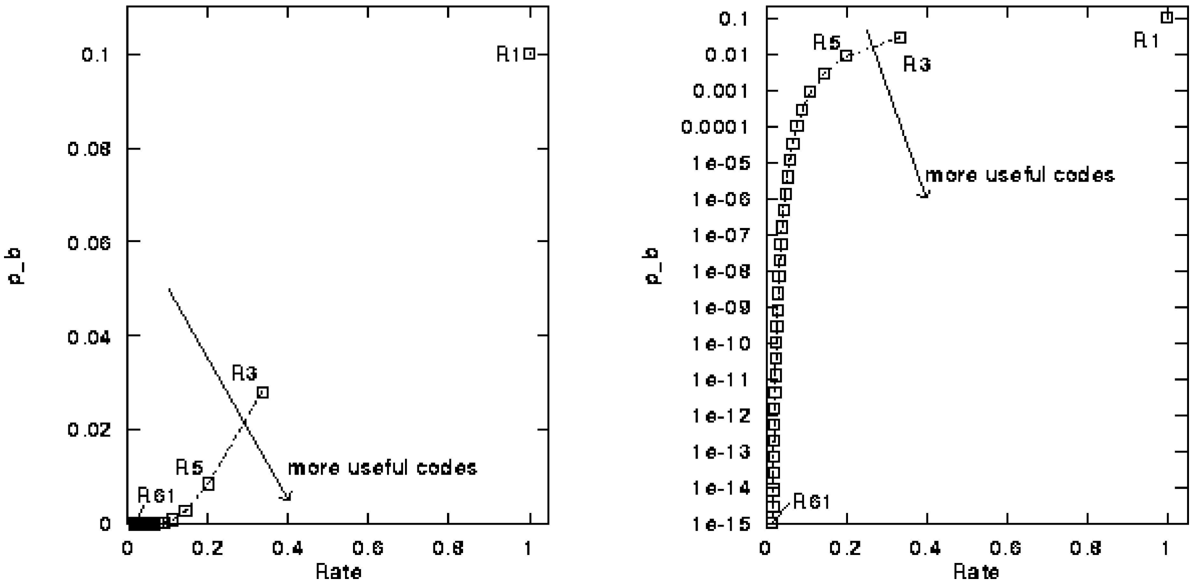


Figure: Error probability p_b versus rate for repetition codes over a binary symmetric channel with f=0.1. The right hand figure shows p_b on a logarithmic scale. We would like the rate to be large and p_b to be small.

Next Up Previous

Next: Block codes -- the **Up:** Error correcting codes for **Previous:** Error correcting codes for

David J.C. MacKay
Sat May 10 23:05:10 BST 1997