College of Computer Science
Northeastern University
Wireless Networks CS G250

Lecture 6
September 29 2008
Lecturer: R. Sundaram

Glomosim notes, Codes, Discrete Case of Shannon's Theorem

Lecture Outline:

- Glomosim Notes
- Codes
  - The Hat Puzzle
  - Hamming Codes
  - Hamming vs. Shannon
- Shannon Theorem, Discrete Case

1 Glomosim notes:

- Linux: Compile & install
- Windows: install VC++ 6.0, compile & install
- Solaris: Get a CCIS account, compile & install

⋆ Instructions are available at
`http://www.ccs.neu.edu/course/csg250/Glomosim/howto.txt`

2 The Hat Puzzle

Suppose a university professor poses the following challenge to his n students:

> I will give you all a chance to pass this course: I will randomly place either a black or a white hat on each of your heads, and you will have to guess its color. You will not be able to see your own hat, but you will be able to see everyone else's hats. Then, when I give a signal, everyone of you must take one of the following two choices:

> 1. Speak to state what color hat you are wearing
> 2. Be silent
>
> To pass this course, those of you who speak must all guess your hat color correctly. If nobody speaks or if at least one of the speakers misses his or her hat color, then you will all fail.
>
> Those of you who choose to speak, must do so simultaneously, and you are not allowed to communicate once I have started placing hats. You are free however, to plan any strategy before the start of class.

What is the maximum chance of success for the students?

Naively, one may think that the best they can do is to choose one speaker as a strategy and let him or her decide randomly. The probability of success in this case is 0.5. Is it possible to do better than this?

As it turns out, it is possible to succeed with probability close to 1. To see how, let's look at an $n = 3$ class. Every student is assigned a color: black (represented by 0), or white (1). An outcome of the hat placement is therefore equivalent to having a vector of 3 bits, with each coordinate being a student.

Being the case that a student can't see the color of his own hat, the 8 possible outcomes can be drawn as vertices in a cube. Adjacent vertices differ only in one coordinate, yielding the problem representation shown in Figure 1. For a given outcome (or vertex), each one of the students will see two possibilities, one where the hat on his/her head is white, and the other one where it is black. Therefore every student is an edge touching the solution vertex; giving an answer is equivalent to assigning a direction to the edge; and staying silent leaves the edge undirected.

What does a successful guess look like in this scenario? It is simply one, two, or three arrows pointing towards the solution vertex, and no arrows pointing away from it. All the students need to do is maximize the number of successful guesses when building their strategy.

The strategy shown in Figure 2 maximizes the chances of winning in this setting. In the case where all hats are black and all hats are white, all three students will be wrong in their guess. However every vertex in the cube except for those two cases will have exactly one arrow pointing towards it, giving a success probability of $6/8 = 3/4$. It can be shown that as $n \to \infty$, the chance of success approaches 1!

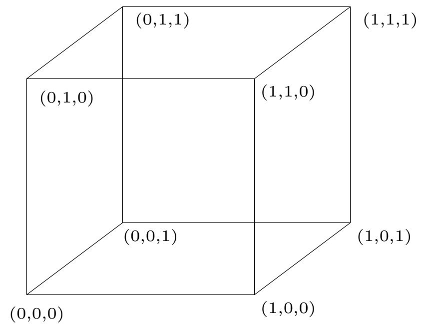

Figure 1: A representation of the hat puzzle. Every vertex represents an outcome for students (A, B, C), and vertices are joined by an edge when they differ by one bit

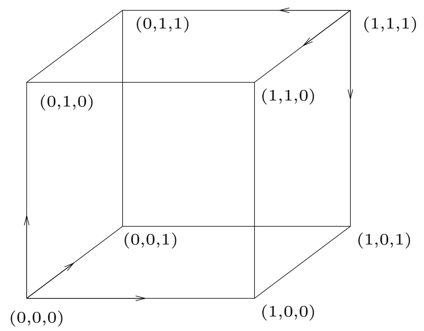

Figure 2: A strategy that maximizes chances of winning

3 Hamming Codes

But what does the Hat Puzzle have to do with Wireless Communications? The cube used in the puzzle can also be used to illustrate how to make sure bits come across a noisy channel. Suppose for instance, that a bit needs to be sent across a channel which flips bits with probability p. One possible solution is not to send only one bit, but several, to compensate for the possible errors.

For instance, the sender may send the string 000 for 0, and 111 for 1. If we know that at most one of the three bits may get flipped, then any of the possible messages received can be mapped to the actual data sent. That is to say, if the received gets the string 011, it is most likely that the sender meant to send a 1 bit.

In general, if 2^k codewords are picked from a 2^n total word space such that their pairwise distances (the number of bits which differ) $d \geq 2t+1$, it will be possible to correct up to t bit errors.

In designing coding schemes, the amount of correctable errors has to be weighed against good data rate. Intuitively, the more bits we use to represent a given signal, the more bit errors we will be able to correct, but the data rate will be reduced. There is no quick answer for this, and the different coding schemes that exist are designed to deal with other practical issues, such as the bit error rate for the given signal encoding, the transmission frequency and protocols.

Codes are characterized by three parameters $(n, k, d)_2$, meaning:

1. 2^n total words
2. 2^k codewords
3. d minimum codeword distance

Hamming codes are $(2^t - 1, 2^t - t - 1, 3)$, and are able to correct 1 bit errors. There are also perfect codes in the sense that there are no leftover codes after partitioning the total space into codewords.

4 Shannon's Theorem, Discrete Case

In a setting where the communication channel flips the bits traversing it, it is natural to ask what is the capacity of such a channel. Shannon provided a proof for the discrete version of his theorem, which uses several notions.

- The noise in the channel flips each bit independently with probability p.
- The channel capacity is $k/n = 1 - H(p)$, where $H(p)$ is the entropy of the channel (a measure of uncertainty), k is the number of bits in a codeword, and n is the number of bits for the total string space.

(continued in next lecture...)