15-251: Great Theoretical Ideas In Computer Science

Recitation 9

Bad Days

Suppose transitions between good days at work and bad days at work can be modeled as a Markov chain. A good day follows another good day with probability 0.7, whereas a bad day follows another bad day with probability 0.6.

(a) Find the transition probability matrix of the Markov chain.

(b) Suppose a worker works for a long time. Approximately what proportion of his days will be good?

Threshold Queue

We define a threshold queue with parameter T as follows: When the number of jobs is $\leq T$, then the number of jobs decreases by 1 with probability 0.4 and increases by 1 with probability 0.6 at each time step. However, when the number of jobs increases to $> T$, then the reverse is true, and the number of jobs increases by 1 with probability 0.4 and decreases by 1 with probability 0.6 at each time step.

(a) Draw the Markov chain for this process.

(b) Assuming that the limiting probabilities exist, derive the system of stationary equations.

(c) Compute the mean number of jobs in a threshold queue as a function of T.

Code Distance

The **distance** of a code is the minimum Hamming distance (number of differing bits) between any two codewords. For example, the Hamming(7, 4) code from lecture has distance 3, because if 4-bit messages a, b differ, then the 7-bit transmitted messages $G'a$ and $G'b$ will always differ in at least three places.

If we send a message with a parity check bit, it has distance 2. This is easy to prove:

- If two bare messages differ in exactly one bit, they differ in parity, so the message sent differs in two bits.
- Otherwise, the messages already differ in more than one bit, and we are done.

Our code with distance 2 can detect one error and correct none. Our code with distance 3 can detect two errors and correct one. Can this be generalized?

(a) Prove that a code with distance d can detect up to $d - 1$ errors.

(b) Prove that a code with distance d can correct up to $(d - 1)/2$ errors.

(c) We'd like to improve the Hamming(7, 4) code without changing it very much. One potential way is to add an 8th bit corresponding to the parity of the 7-bit codeword. Does this improve error detection? How about correction?

Correction with Polynomials

These questions also appear as warm-ups to homework 8. Note that unlike in lecture, the message is encoded as values taken by the function, not as coefficients. Is either scheme better?

(a) Suppose Alice wants to send Bob 3 numbers between 0 and 6 inclusive and she wants to guard against 1 corrupted packet.

- What's the smallest prime field Alice can use?

- Suppose Alice wants to send Bob $m = ((1, m_1), (2, m_2), (3, m_3))$. what is the maximum degree for which a unique polynomial fits these points?

- What is the minimum number of extra points Alice must send Bob so that he can correctly reconstruct her message m?

- Bob receive a message $r = (3, 3, 3, 2, 0)$. In order to check whether the message is corrupted, Bob needs to solve $N(x) = r_i E(x)$, where $N(x) = P(x)E(x)$, $P(x)$ is the original polynomial used to send the message, and $E(x)$ is the error-locator polynomial from the Berlekamp-Welch algorithm. What are the degrees of $N(x)$ and $E(x)$?

- What is the solution to the corresponding system of linear equations:

$$
\begin{aligned}
a_3 + a_2 + a_1 + a_0 &= 3 + 3b_0 \\
a_3 + 4a_2 + 2a_1 + a_0 &= 6 + 3b_0 \\
6a_3 + 2a_2 + 3a_1 + a_0 &= 2 + 3b_0 \\
a_3 + 2a_2 + 4a_1 + a_0 &= 1 + 2b_0 \\
6a_3 + 4a_2 + 5a_1 + a_0 &= 0
\end{aligned}
$$

- What is the original polynomial $P(x) = ax^2 + bx + c$?

- Which packet is corrupted, and what is the original value?