

# CMPSCI 240: Fourth Summary

## 1 Game Theory

### 1.1 Zero-Sum Games

A *two-player, simultaneous-move, zero-sum game* consists of a set of options  $\{a_1, \dots, a_k\}$  for player  $A$ , a set of options  $\{b_1, \dots, b_l\}$  for player  $B$ , and a  $k \times l$  *payoff matrix*  $P$ . If  $A$  chooses  $a_i$  and  $B$  chooses  $b_j$  then  $A$  gets  $\$P_{ij}$  and  $B$  gets  $-\$P_{ij}$ .

For example, for two-finger Morra, the payoff matrix is

	1 B Finger	2 B Finger
1 A Finger	+2	-3
2 A Finger	-3	+4

To find the optimal strategies for  $A$  and  $B$ , consider  $A$  announcing her strategy, i.e., the probabilities  $(r_1, r_2, \dots, r_k)$  with which she'll play each strategy.  $B$  then picks his option such that  $A$ 's expected winnings are minimized. Let  $f(r_1, r_2, \dots, r_k)$  be this value.  $A$  picks her distribution  $r_1, \dots, r_k$  such that  $f(r_1, r_2, \dots, r_k)$  is maximized. This is  $A$ 's optimal strategy. To find  $B$ 's optimal reverse their roles.

### 1.2 Non Zero-Sum Games and Nash Equilibrium

In a non zero-sum game,  $B$ 's winning aren't necessary the negative of  $A$ 's winnings. For example, if two birds meet over a piece of food and have to decide whether to act aggressive (*hawkish*) or passive (*dovish*):

- If a hawk meets a dove, the hawk gets the food worth 50 points
- If two hawks meet they both loose -25 points
- If two doves meet, they both get 15 points

We can represent this as:

	B is a Hawk	B is a Dove
A is a Hawk	-25, -25	50, 0
A is a Dove	0, 50	15, 15

**Theorem 1.** *A Nash Equilibrium is a pair of strategies for the players where no change by one player alone can improve his outcome. Every game where each player has a finite number of options, has at least one Nash equilibrium.*

A Nash Equilibrium is *pure* if the pair of strategies involves each player playing a single option. A Nash Equilibrium is *mixed* if the strategies involve playing different options with certain probabilities. In the above example, one pure Nash Equilibrium is if  $A$  plays Dove and  $B$  plays Hawk.

To find a mixed Nash Equilibrium, suppose  $A$  play Hawk with probability  $0 < q < 1$  and  $B$  plays Hawk with probability  $0 < p < 1$ . Then  $A$ 's expected reward is

$$-25pq + 50q(1 - p) + 15(1 - p)(1 - q) = q(35 - 60p) + 15 - 15p.$$

So if  $p = 7/12$  then  $A$  can't increase her expected reward by changing  $q$ . Similar reasons implies that  $B$  can't change his expected reward by changing  $p$  if  $q = 7/12$ . Hence  $p = q = 7/12$  is the only Nash Equilibrium with mixed strategies.

## 2 Information Theory

In this final section of the course we considered encoding messages as binary strings. A natural way of doing this is to index the possible messages and let the corresponding binary string be the binary expansion of the index with leading zero's such that all binary strings are the same length. For example,

“a”	→	000
“b”	→	001
“c”	→	010
“d”	→	011
“e”	→	100

### 2.1 Coding for Transmission

However, sometimes we want to use more bits than necessary so that if we can still work out the message even if some of the bits in the binary string gets flipped (error correction) or at least identify that bits have been flipped (error detection). We defined the Hamming distance between two binary strings to be the number of positions in which they differed. We considered two examples in class. In the first example, we added a parity bit and this guaranteed that all strings were at least Hamming distance 2 apart. Hence, we'd detect an error if an odd number of errors occurred. In the second example, we added three bits and this guaranteed that all strings were at least Hamming distance 3 apart. Hence, we'd be able to guess the correct message as long as at most 1 error occurred since there'd be at most one valid string that is at most Hamming distance 1 from the received binary string.

### 2.2 Coding for Compression

Sometimes, we want to use as few bits as possible and furthermore, if we know that some messages are more likely than other messages, we want to use shorter binary strings for the more likely messages. However, in this case we insist that the set of binary strings we construct is *prefix-free*, i.e., one string can't be a prefix of another.

We construct a good set of prefix-free binary strings for our messages as follows. If there are  $k$  messages, we consider a binary tree with  $k$  leaves where each leaf is labeled by one of the messages.

For each leaf  $u$ , consider the path from root to leaf. We associate the path with a binary string in the natural way, e.g., if the path goes left-left-right then this corresponds to the binary string 001. Then the *average encoding length* is  $\sum_u d(u)P(u)$  where  $P(u)$  is the probability of the message associated with leaf  $u$  and  $d(u)$  is the length of the path from the root to leaf  $u$ . A way of constructing a tree that minimizes the average encoding length is the Huffman encoding.