

Homework 1

*Instructor: Anna Lysyanskaya**Due: Jan 31, 2007***Problem 1: Multi-Party Computation**

Let's design a simple multiparty computation protocol for a specific task. Suppose we have a room full of students who have just received scores (1-10 points) for their last exam. They would like to compute the average score, but of course no student wants anyone else to find out his score.

The students are all honest – they will follow whatever instructions they are given, and they will not team up to share information with any other student. However, they are curious about their fellow students, so they will try to discover as much as possible from the information they are given.

One student proposes that all the students sit in a circle. This student will whisper a number to the student next to him, who will compute another number and whisper it to the next student and so on. When this process reaches the first student again, he will announce the answer.

Suggest a protocol in this framework, and explain how it fulfills these goals.

Problem 2: Probability Practice

Suppose we are given a machine which distinguishes male faces from female faces. For both male and female faces, with probability 70% it will correctly answer “male” or “female”. However, on 30%, of the faces, it will fail and output “Error”. (If the machine is run multiple times on the same face, it will always give the same result.)

- Suppose we are given a set of images with equal numbers of male and female faces. Construct an algorithm which can correctly identify male and female faces with probability 85%.
- What if we want to run the algorithm on CS concentrators, 80% of whom are male? Can we create a more successful algorithm in this case?
- Now, suppose we have n such machines, each with a 70% success rate. The event that one machine is successful is independent of the success of others. Can we generate an algorithm that will be correct 90% of the time? What about 60% of the time? For what values of k (in terms of n) can you achieve success with probability $1 - 1/2^k$? Is it possible to create an algorithm which is successful exactly 100% of the time?
- What if the machine could not identify its errors? (i.e. on 30% of the inputs it simply outputs the incorrect answer.) Given a single such machine, could we construct an algorithm that correctly determines the gender of CS concentrators 80% of the time? What if we are given several such machines? Propose an algorithm which uses several machines to get a success probability higher than 80%.

Problem 3: Cryptanalysis

A very basic cipher simply shifts each letter in the message forward in the alphabet. If we shift each letter forward by 2, CRYPTANALYSIS becomes ETARVCPNAUKU. This is relatively simple to break (ex just try all 26 possible shifts). What if instead we use a key to determine how far to shift each letter? We can represent a key using letters of the alphabet so that A represents a 1 letter shift,

B represents a 2 letter shift, etc. If the message is longer than the key, we repeat the key. Then, using the key SAMPLEKEY, if we encrypt the word CRYPTANALYSIS, we get VSLFFFYFKRTVI.

- Suppose we are given the length of the key, and some ciphertext which we know is the encryption of an English text. Describe an algorithm that can discover the key given a long enough ciphertext. (Hint: see letter frequency charts below.)
- Suppose we are not given the length of the key. Is it possible to determine it given only a long ciphertext?
- We have posted a file on the website which contains the encryption of an English text using this scheme. (The encryption ignores spaces, punctuation, and capitalization, so that these will all be the same as in the original text). What text is it, and what is the key?
- What if we are given the encryption of a long text which is not English, but generated from some other unknown language. If we are given the length of the key, could we determine anything about this language? (Hint: Perhaps about letter frequencies?)

Note if the key were as long as the message, this would be a *one-time pad*.

English Letter Frequencies:

letter	frequency	letter	frequency
A	.082	N	.067
B	.015	O	.075
C	.028	P	.019
D	.043	Q	.001
E	.127	R	.060
F	.022	S	.063
G	.020	T	.091
H	.061	U	.028
I	.070	V	.010
J	.002	W	.023
K	.008	X	.001
L	.040	Y	.020
M	.024	Z	.001