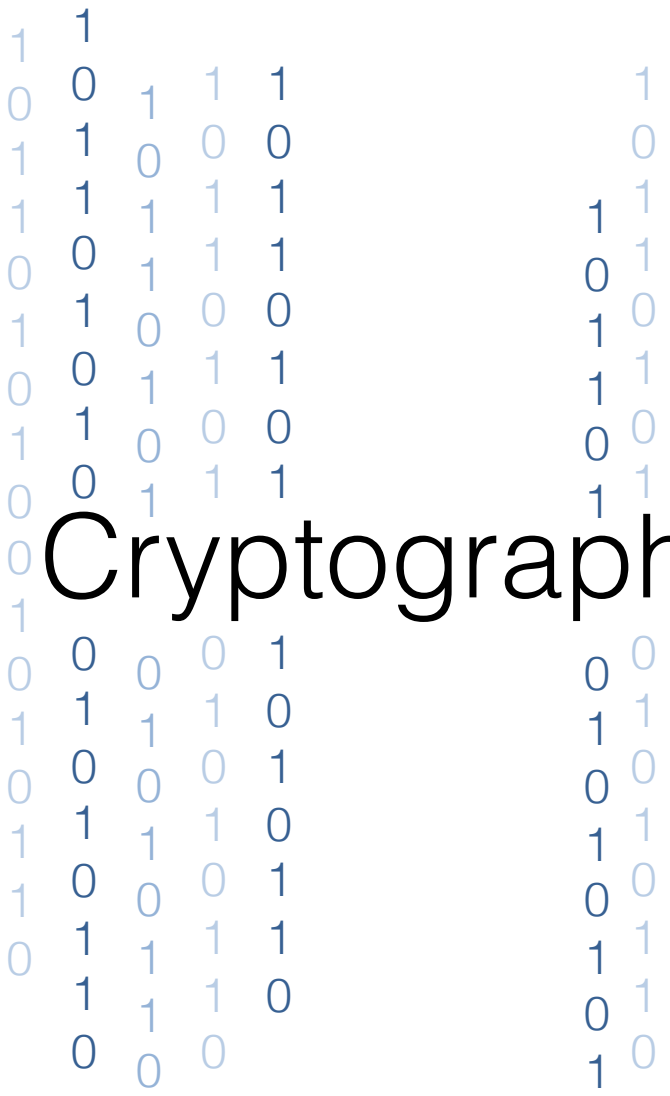

Cryptography

Project 7

Outline

1. decoderdrive
2. decoder
3. downlow
4. downlowinv
5. loglike

Outline

1. decoderdrive
   - calls decoder once for each message, displays output
2. decoder
   - decodes message using Metropolis Algorithm
3. downlow
   - letters → numbers
4. downlowinv
   - numbers → letters
5. loglike
   - computes likelihood of your guess given the inputted text (message you're trying to decode)

Metropolis Algorithm

First: letter pair probabilities (in letterprob.dat, use load) and encoded text (use fileread) need to be imported.

Generate an initial random guess:

y = randperm(27)

To compare the encoded message to your guess, you need to convert it to numbers

Metropolis Algorithm

First: letter pair probabilities (in letterprob.dat, use load) and encoded text (use fileread) need to be imported.

Generate an initial random guess, y = randperm(27).

Main loop:

-consider a potential guess, ymaybe, obtained by switching 2 random elements in y

You have two possibilities for the next guess. Either ynew = ymaybe, or no change is made (ynew = y).

Metropolis Algorithm

First: letter pair probabilities (in letterprob.dat, use load) and encoded text (use fileread) need to be imported. Generate initial random guess.

Main loop: (repeated 10^4 or 10^5 times)

-consider a potential guess, ymaybe, obtained by switching 2 random elements in y

You have two possibilities for the next guess. Either ynew = ymaybe, or no change is made (ynew = y).

-Compute log likelihood of y and ymaybe.

(a) loglike(ymaybe) > loglike(y)

→ ynew = ymaybe with 100% probability

(b) loglike(ymaybe) < loglike(y)

→ ynew = ymaybe with probability =

exp[-loglike(y) + loglike(ymaybe)],

otherwise ynew = y (guess doesn't change)

Metropolis Algorithm

First: letter pair probabilities (in letterprob.dat, use load) and encoded text (use fileread) need to be imported. Generate initial random guess.

Main loop: (repeated 10^4 or 10^5 times)

-consider a potential guess, ymaybe, obtained by switching 2 random elements in y

You have two possibilities for the next guess. Either ynew = ymaybe, or no change is made (ynew = y).

-Compute log likelihood of y and ymaybe.

(a) loglike(ymaybe) > loglike(y)

→ ynew = ymaybe with 100% probability

(b) loglike(ymaybe) < loglike(y)

→ ynew = ymaybe with probability =

exp[-loglike(y) + loglike(ymaybe)],

otherwise ynew = y (guess doesn't change)

Display encrypted & decrypted messages

loglike function

likelihood = loglike(crypt, guess, M)

$$\sum_i \log(M(t_{\sigma(i)}, t_{\sigma(i+1)}))$$

For each pair of subsequent letters in the encrypted message (‘crypt’), find your guess for those letters (in ‘guess’).

Add the log of the corresponding element in M to the sum.

downlow and lowdown

num = downlow(text) | letter= lowdowninv(num)

use “double”

```
>>s=double('hello')

 s =

    104 101 108 108 111
```

use “char”

```
>> [char(s)]

ans =

hello
```