BILL, RECORD LECTURE!!!!

BILL RECORD LECTURE!!!

The Shift Cipher

Shift Cipher: Encryption, Decryption, Cracking

The Shift Cipher

The Shift Cipher

- Consider encrypting English text.

The Shift Cipher

- Consider encrypting English text.
- Associate 'a' with 0; 'b' with 1; . . . ; 'z' with 25.

The Shift Cipher

- Consider encrypting English text.
- Associate ‘a’ with 0; ‘b’ with 1; ...; ‘z’ with 25.
- $s \in \{0, \ldots, 25\}$ (or could think of $s \in \{a, \ldots, z\}$).

The Shift Cipher

- Consider encrypting English text.
- Associate ‘a’ with 0; ‘b’ with 1; . . . ; ‘z’ with 25.
- $s \in \{0, \ldots, 25\}$ (or could think of $s \in \{a, \ldots, z\}$).
- To encrypt using key s, shift every letter of the plaintext by s positions (with wraparound).

The Shift Cipher: Examples of Encryption

I want to encode **Bill works at a zoo!** with a shift-3.

The Shift Cipher: Examples of Encryption

I want to encode **Bill works at a zoo!** with a shift-3.

1. Do usual preprocessing: blocks of five, etc to get:
   billw orksa tazoo

The Shift Cipher: Examples of Encryption

I want to encode **Bill works at a zoo!** with a shift-3.

1. Do usual preprocessing: blocks of five, etc to get:
   billw orksa tazoo
2. Convert letters to numbers to get:
   1-8-11-11-22 14-17-10-18-0 19-0-25-14-14

The Shift Cipher: Examples of Encryption

I want to encode **Bill works at a zoo!** with a shift-3.

1. Do usual preprocessing: blocks of five, etc to get:
   billw orksa tazoo
2. Convert letters to numbers to get:
   1-8-11-11-22 14-17-10-18-0 19-0-25-14-14
3. Add three to each number (wrap around) to get:
   4-11-14-14-25 17-20-13-21-3 22-3-2-17-17

The Shift Cipher: Examples of Encryption

I want to encode **Bill works at a zoo!** with a shift-3.

1. Do usual preprocessing: blocks of five, etc to get:
   billw orksa tazoo
2. Convert letters to numbers to get:
   1-8-11-11-22 14-17-10-18-0 19-0-25-14-14
3. Add three to each number (wrap around) to get:
   4-11-14-14-25 17-20-13-21-3 22-3-2-17-17
4. Convert numbers to letters to get:
   elooz runvd wdcrr

The Shift Cipher: How do Decrypt

Bob knows Alice used shift-3. How does he decrypt?

The Shift Cipher: How do Decrypt

Bob knows Alice used shift-3. How does he decrypt?
He does shift by -3 or can view as shift by $26 - 3 = 23$.

The Shift Cipher: An Example of Decrypt

The Shift Cipher: An Example of Decrypt

Bob has to decode **mrvkx dolnh vpo** which was coded by shift-3.

The Shift Cipher: An Example of Decrypt

Bob has to decode **mrvkx dolnh vpo** which was coded by shift-3.

1. Convert letters to numbers to get:
   12-17-21-10-23 **3-14-11-13-7** **21-15-14**.

The Shift Cipher: An Example of Decrypt

Bob has to decode **mrvkx dolnh vpo** which was coded by shift-3.

1. Convert letters to numbers to get:
   12-17-21-10-23 **3-14-11-13-7** **21-15-14**.
2. Subtract 3 from each number (wrap around) to get:
   9-14-18-7-20 **0-11-8-10-4** **18-12-11**.

The Shift Cipher: An Example of Decrypt

Bob has to decode **mrvkx dolnh vpo** which was coded by shift-3.

1. Convert letters to numbers to get:
   12-17-21-10-23 3-14-11-13-7 21-15-14.
2. Subtract 3 from each number (wrap around) to get:
   9-14-18-7-20 0-11-8-10-4 18-12-11.
3. Convert numbers to letters to get: **joshu alike sml**.

The Shift Cipher: An Example of Decrypt

Bob has to decode **mrvkx dolnh vpo** which was coded by shift-3.

1. Convert letters to numbers to get:
   12-17-21-10-23 3-14-11-13-7 21-15-14.
2. Subtract 3 from each number (wrap around) to get:
   9-14-18-7-20 0-11-8-10-4 18-12-11.
3. Convert numbers to letters to get: **joshu alike sml**.
4. Figure out spacing to get: **Joshua likes ML**.

“Wrap Around” is Modular Arithmetic: Definitions

“Wrap Around” is Modular Arithmetic: Definitions

- $x \equiv y \pmod{N}$ if and only if N divides $x - y$.

“Wrap Around” is Modular Arithmetic: Definitions

- $x \equiv y \pmod{N}$ if and only if N divides $x - y$.
- $[x \bmod N]$ = the remainder when x is divided by N.

“Wrap Around” is Modular Arithmetic: Definitions

- $x \equiv y \pmod{N}$ if and only if N divides $x - y$.
- $[x \bmod N]$ = the remainder when x is divided by N.
  - i.e. the unique value $y \in \{0, \ldots, N-1\}$ such that $x \equiv y \pmod{N}$.

“Wrap Around” is Modular Arithmetic: Definitions

- $x \equiv y \pmod{N}$ if and only if N divides $x - y$.
- $[x \bmod N]$ = the remainder when x is divided by N.
  - i.e. the unique value $y \in \{0, \ldots, N-1\}$ such that $x \equiv y \pmod{N}$.
- $25 \equiv 35 \pmod{10}$

“Wrap Around” is Modular Arithmetic: Definitions

- $x \equiv y \pmod{N}$ if and only if N divides $x - y$.
- $[x \bmod N]$ = the remainder when x is divided by N.
  - i.e. the unique value $y \in \{0, \ldots, N-1\}$ such that $x \equiv y \pmod{N}$.
- $25 \equiv 35 \pmod{10}$
- $25 \neq [35 \bmod 10]$

“Wrap Around” is Modular Arithmetic: Definitions

- $x \equiv y \pmod{N}$ if and only if N divides $x - y$.
- $[x \bmod N]$ = the remainder when x is divided by N.
  - i.e. the unique value $y \in \{0, \ldots, N-1\}$ such that $x \equiv y \pmod{N}$.
- $25 \equiv 35 \pmod{10}$
- $25 \neq [35 \bmod 10]$
- $5 = [35 \bmod 10]$

Modular Arithmetic II: Convention

Common usage:

$$100 \equiv 2 \pmod{7}$$

Modular Arithmetic II: Convention

Common usage:

$$100 \equiv 2 \pmod 7$$

Commonly if we are in Mod n we have a large number on the left and then a number between 0 and $n-1$ on the right.

Modular Arithmetic II: Convention

Common usage:

$$100 \equiv 2 \pmod{7}$$

Commonly if we are in Mod n we have a large number on the left and then a number between 0 and $n-1$ on the right.

When dealing with mod n we assume the entire universe is $\{0, 1, \ldots, n-1\}$.

Modular Arithmetic: $+, -, \times$

$\equiv$ is Mod 26 for this slide.

Modular Arithmetic: $+, -, \times$

$\equiv$ is Mod 26 for this slide.

1. Addition: $x + y$ is easy: wrap around. E.g., $20 + 10 \equiv 30 \equiv 4$. Only use the number 30 as an intermediary value on the way to the real answer.

Modular Arithmetic: $+, -, \times$

$\equiv$ is Mod 26 for this slide.

1. Addition: $x + y$ is easy: wrap around. E.g., $20 + 10 \equiv 30 \equiv 4$. Only use the number 30 as an intermediary value on the way to the real answer.
2. $-7 \equiv x$ where $0 \le x \le 25$.

Modular Arithmetic: $+, -, \times$

$\equiv$ is Mod 26 for this slide.

1. Addition: $x + y$ is easy: wrap around. E.g., $20 + 10 \equiv 30 \equiv 4$. Only use the number 30 as an intermediary value on the way to the real answer.
2. $-7 \equiv x$ where $0 \le x \le 25$.
   Pedantic $-y$ is the number such that $y + (-y) \equiv 0$.

Modular Arithmetic: $+, -, \times$

$\equiv$ is Mod 26 for this slide.

1. Addition: $x + y$ is easy: wrap around. E.g., $20 + 10 \equiv 30 \equiv 4$. Only use the number 30 as an intermediary value on the way to the real answer.
2. $-7 \equiv x$ where $0 \le x \le 25$.
   Pedantic $-y$ is the number such that $y + (-y) \equiv 0$.
   $-7 \equiv 19 \pmod{26}$ because $19 + 7 \equiv 0 \pmod{26}$.

Modular Arithmetic: $+, -, \times$

$\equiv$ is Mod 26 for this slide.

1. Addition: $x + y$ is easy: wrap around. E.g., $20 + 10 \equiv 30 \equiv 4$. Only use the number 30 as an intermediary value on the way to the real answer.
2. $-7 \equiv x$ where $0 \le x \le 25$.
   Pedantic $-y$ is the number such that $y + (-y) \equiv 0$.
   $-7 \equiv 19 \pmod{26}$ because $19 + 7 \equiv 0 \pmod{26}$.
   Shortcut: $-y \equiv 26 - y$.

Modular Arithmetic: $+, -, \times$

$\equiv$ is Mod 26 for this slide.

1. Addition: $x + y$ is easy: wrap around. E.g., $20 + 10 \equiv 30 \equiv 4$. Only use the number 30 as an intermediary value on the way to the real answer.
2. $-7 \equiv x$ where $0 \le x \le 25$.
   Pedantic $-y$ is the number such that $y + (-y) \equiv 0$.
   $-7 \equiv 19 \pmod{26}$ because $19 + 7 \equiv 0 \pmod{26}$.
   Shortcut:$-y \equiv 26 - y$.
3. Mult: xy is easy: wrap around. E.g., $20 \times 10 \equiv 200 \equiv 18$.

Modular Arithmetic: $+, -, \times$

$\equiv$ is Mod 26 for this slide.

1. Addition: $x + y$ is easy: wrap around. E.g., $20 + 10 \equiv 30 \equiv 4$. Only use the number 30 as an intermediary value on the way to the real answer.
2. $-7 \equiv x$ where $0 \le x \le 25$.
   Pedantic $-y$ is the number such that $y + (-y) \equiv 0$.
   $-7 \equiv 19 \pmod{26}$ because $19 + 7 \equiv 0 \pmod{26}$.
   Shortcut:$-y \equiv 26 - y$.
3. Mult: xy is easy: wrap around. E.g., $20 \times 10 \equiv 200 \equiv 18$.
   Shortcut to avoid big numbers:

Modular Arithmetic: $+, -, \times$

$\equiv$ is Mod 26 for this slide.

1. Addition: $x + y$ is easy: wrap around. E.g., $20 + 10 \equiv 30 \equiv 4$. Only use the number 30 as an intermediary value on the way to the real answer.
2. $-7 \equiv x$ where $0 \le x \le 25$.
   Pedantic $-y$ is the number such that $y + (-y) \equiv 0$.
   $-7 \equiv 19 \pmod{26}$ because $19 + 7 \equiv 0 \pmod{26}$.
   Shortcut:$-y \equiv 26 - y$.
3. Mult: xy is easy: wrap around. E.g., $20 \times 10 \equiv 200 \equiv 18$.
   Shortcut to avoid big numbers:

$$20 \times 10 \equiv -6 \times 10 \equiv -2 \times 30 \equiv -2 \times 4 \equiv -8 \equiv 18.$$

Modular Arithmetic: $+, -, \times$

$\equiv$ is Mod 26 for this slide.

1. Addition: $x + y$ is easy: wrap around. E.g., $20 + 10 \equiv 30 \equiv 4$. Only use the number 30 as an intermediary value on the way to the real answer.
2. $-7 \equiv x$ where $0 \le x \le 25$.
   Pedantic $-y$ is the number such that $y + (-y) \equiv 0$.
   $-7 \equiv 19 \pmod{26}$ because $19 + 7 \equiv 0 \pmod{26}$.
   Shortcut:$-y \equiv 26 - y$.
3. Mult: xy is easy: wrap around. E.g., $20 \times 10 \equiv 200 \equiv 18$. Shortcut to avoid big numbers:

$$20 \times 10 \equiv -6 \times 10 \equiv -2 \times 30 \equiv -2 \times 4 \equiv -8 \equiv 18.$$

4. Division: Next Slide

Modular Arithmetic: $\div$

$\equiv$ is Mod 26 for this slide.
$\frac{1}{3} \equiv x$ where $0 \leq x \leq 25$.

Modular Arithmetic: $\div$

$\equiv$ is Mod 26 for this slide.
$\frac{1}{3} \equiv x$ where $0 \le x \le 25$.
Pedantic $\frac{1}{y}$ is the number such that $y \times \frac{1}{y} \equiv 1$.

Modular Arithmetic: $\div$

$\equiv$ is Mod 26 for this slide.
$\frac{1}{3} \equiv x$ where $0 \le x \le 25$.
Pedantic $\frac{1}{y}$ is the number such that $y \times \frac{1}{y} \equiv 1$.
$\frac{1}{3} \equiv 9$ since $9 \times 3 = 27 \equiv 1$.

Modular Arithmetic: $\div$

$\equiv$ is Mod 26 for this slide.
$\frac{1}{3} \equiv x$ where $0 \le x \le 25$.
Pedantic $\frac{1}{y}$ is the number such that $y \times \frac{1}{y} \equiv 1$.
$\frac{1}{3} \equiv 9$ since $9 \times 3 = 27 \equiv 1$.
Shortcut:

Modular Arithmetic: $\div$

$\equiv$ is Mod 26 for this slide.
$\frac{1}{3} \equiv x$ where $0 \le x \le 25$.
Pedantic $\frac{1}{y}$ is the number such that $y \times \frac{1}{y} \equiv 1$.
$\frac{1}{3} \equiv 9$ since $9 \times 3 = 27 \equiv 1$.
Shortcut: there is an algorithm that finds $\frac{1}{y}$ quickly.

Modular Arithmetic: $\div$

$\equiv$ is Mod 26 for this slide.
$\frac{1}{3} \equiv x$ where $0 \le x \le 25$.
Pedantic $\frac{1}{y}$ is the number such that $y \times \frac{1}{y} \equiv 1$.
$\frac{1}{3} \equiv 9$ since $9 \times 3 = 27 \equiv 1$.
Shortcut: there is an algorithm that finds $\frac{1}{y}$ quickly.
You will look up this algorithm—hw 00 which you do not hand in.

Modular Arithmetic: $\div$

$\equiv$ is Mod 26 for this slide.
$\frac{1}{3} \equiv x$ where $0 \le x \le 25$.
Pedantic $\frac{1}{y}$ is the number such that $y \times \frac{1}{y} \equiv 1$.
$\frac{1}{3} \equiv 9$ since $9 \times 3 = 27 \equiv 1$.
Shortcut: there is an algorithm that finds $\frac{1}{y}$ quickly.
You will look up this algorithm—hw 00 which you do not hand in.

$\frac{1}{2} \equiv x$ where $0 \le x \le 25$.

Modular Arithmetic: $\div$

$\equiv$ is Mod 26 for this slide.
$\frac{1}{3} \equiv x$ where $0 \le x \le 25$.
Pedantic $\frac{1}{y}$ is the number such that $y \times \frac{1}{y} \equiv 1$.
$\frac{1}{3} \equiv 9$ since $9 \times 3 = 27 \equiv 1$.
Shortcut: there is an algorithm that finds $\frac{1}{y}$ quickly.
You will look up this algorithm—hw 00 which you do not hand in.

$\frac{1}{2} \equiv x$ where $0 \le x \le 25$. Think about.

Modular Arithmetic: $\div$

$\equiv$ is Mod 26 for this slide.
$\frac{1}{3} \equiv x$ where $0 \le x \le 25$.
Pedantic $\frac{1}{y}$ is the number such that $y \times \frac{1}{y} \equiv 1$.
$\frac{1}{3} \equiv 9$ since $9 \times 3 = 27 \equiv 1$.
Shortcut: there is an algorithm that finds $\frac{1}{y}$ quickly.
You will look up this algorithm—hw 00 which you do not hand in.

$\frac{1}{2} \equiv x$ where $0 \le x \le 25$. Think about.
No such x exists.

Modular Arithmetic: $\div$

$\equiv$ is Mod 26 for this slide.
$\frac{1}{3} \equiv x$ where $0 \le x \le 25$.
Pedantic $\frac{1}{y}$ is the number such that $y \times \frac{1}{y} \equiv 1$.
$\frac{1}{3} \equiv 9$ since $9 \times 3 = 27 \equiv 1$.
Shortcut: there is an algorithm that finds $\frac{1}{y}$ quickly.
You will look up this algorithm—hw 00 which you do not hand in.

$\frac{1}{2} \equiv x$ where $0 \le x \le 25$. Think about.
No such x exists.
Fact A number y has an inverse mod 26 if y and 26 have no common factors. Numbers that have an inverse mod 26:

$$\{1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25\}$$

Proof is not hard but I won't be doing it **Proof is not hard but I won't be doing it**

The Shift Cipher, Formally

- $\mathcal{M} = \{\text{all texts in lowercase English alphabet}\}$
  $\mathcal{M}$ for **Message space**.
  All arithmetic mod 26.

- Choose uniform $s \in \mathcal{K} = \{0, \ldots, 25\}$. $\mathcal{K}$ for **Keyspace**.

- Encode $(m_1 \ldots m_t)$ as $(m_1 + s \ldots m_t + s)$.

- Decode $(c_1 \ldots c_t)$ as $(c_1 - s \ldots c_t - s)$.

- Can verify that correctness holds.

Cracking the Shift Cipher

Is the Shift Cipher Secure?

Is the Shift Cipher Secure?

Discuss

Is the Shift Cipher Secure?

Discuss Since this is a slide show I have to guess what the discussion lead to; however I suspect that you said or thought **There are only 26 possible keys, and 26 is small, so crackable.**

Is the Shift Cipher Secure?

Discuss Since this is a slide show I have to guess what the discussion lead to; however I suspect that you said or thought **There are only 26 possible keys, and 26 is small, so crackable.**

That is correct but incomplete.

Is the Shift Cipher Secure?

Discuss Since this is a slide show I have to guess what the discussion lead to; however I suspect that you said or thought **There are only 26 possible keys, and 26 is small, so crackable.**

That is correct but incomplete.

Here is the algorithm that reasoning leads to

Is the Shift Cipher Secure?

Discuss Since this is a slide show I have to guess what the discussion lead to; however I suspect that you said or thought **There are only 26 possible keys, and 26 is small, so crackable.**

That is correct but incomplete.

Here is the algorithm that reasoning leads to

1. Input T a text.

Is the Shift Cipher Secure?

Discuss Since this is a slide show I have to guess what the discussion lead to; however I suspect that you said or thought **There are only 26 possible keys, and 26 is small, so crackable.**

That is correct but incomplete.

Here is the algorithm that reasoning leads to

1. Input T a text.
2. For $s = 0$ to 25 generate T_s (T shifted by s)

Is the Shift Cipher Secure?

Discuss Since this is a slide show I have to guess what the discussion lead to; however I suspect that you said or thought **There are only 26 possible keys, and 26 is small, so crackable.**

That is correct but incomplete.

Here is the algorithm that reasoning leads to

1. Input T a text.
2. For $s = 0$ to 25 generate T_s (T shifted by s)
3. **Look** at each T_s. One will **look like** English.

Is the Shift Cipher Secure?

Discuss Since this is a slide show I have to guess what the discussion lead to; however I suspect that you said or thought **There are only 26 possible keys, and 26 is small, so crackable.**

That is correct but incomplete.

Here is the algorithm that reasoning leads to

1. Input T a text.
2. For $s = 0$ to 25 generate T_s (T shifted by s)
3. **Look** at each T_s. One will **look like** English.

This works. There will only be one text that **looks like** English.

Is the Shift Cipher Secure?

Discuss Since this is a slide show I have to guess what the discussion lead to; however I suspect that you said or thought **There are only 26 possible keys, and 26 is small, so crackable.**

That is correct but incomplete.

Here is the algorithm that reasoning leads to

1. Input T a text.
2. For $s = 0$ to 25 generate T_s (T shifted by s)
3. **Look** at each T_s. One will **look like** English.

This works. There will only be one text that **looks like** English. But do you really want to **look at 26 texts**? No you do not.

Is the Shift Cipher Secure?

Discuss Since this is a slide show I have to guess what the discussion lead to; however I suspect that you said or thought **There are only 26 possible keys, and 26 is small, so crackable.**

That is correct but incomplete.

Here is the algorithm that reasoning leads to

1. Input T a text.
2. For $s = 0$ to 25 generate T_s (T shifted by s)
3. **Look** at each T_s. One will **look like** English.

This works. There will only be one text that **looks like** English. But do you really want to **look at 26 texts**? No you do not. So what to do? Discuss.

One way to use Letter Freq

One way to use Letter Freq

Idea Assume the most common letter in T maps to e. Use this to determine shift.

One way to use Letter Freq

Idea Assume the most common letter in T maps to e. Use this to determine shift.

Slight Modification Check by hand if it looks like English. If not then look at second most common letter. For a long text of normal English you will get the right shift very soon.

One way to use Letter Freq

Idea Assume the most common letter in T maps to e. Use this to determine shift.

Slight Modification Check by hand if it looks like English. If not then look at second most common letter. For a long text of normal English you will get the right shift very soon.
Works for Shift but...

One way to use Letter Freq

Idea Assume the most common letter in T maps to e. Use this to determine shift.

Slight Modification Check by hand if it looks like English. If not then look at second most common letter. For a long text of normal English you will get the right shift very soon.
Works for Shift but. . .
We will do more complicated ciphers where nothing like that works.

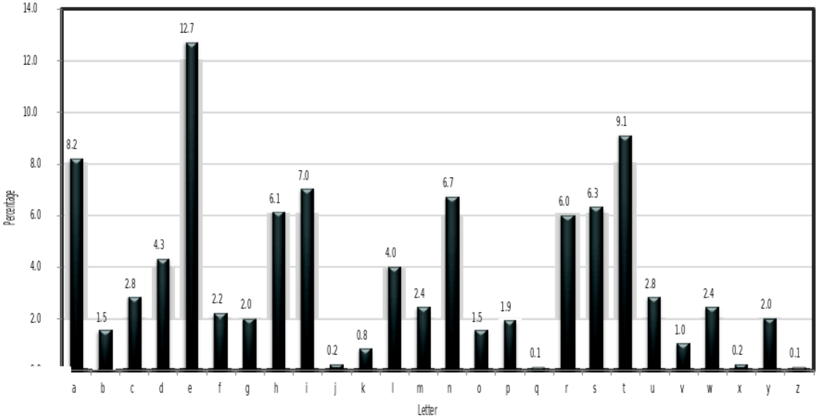
One way to use Letter Freq

Idea Assume the most common letter in T maps to e. Use this to determine shift.

Slight Modification Check by hand if it looks like English. If not then look at second most common letter. For a long text of normal English you will get the right shift very soon.
Works for Shift but...
We will do more complicated ciphers where nothing like that works.

We want a way for **a program** to tell us if a text **looks like English**.

Letter Frequencies



Freq Vectors

Let T be a long text. Length N. May or may not be coded.

Let N_a be the number of $a's$ in T.
Let N_b be the number of $b's$ in T.
$\vdots$

Freq Vectors

Let T be a long text. Length N. May or may not be coded.

Let N_a be the number of $a's$ in T.
Let N_b be the number of $b's$ in T.
$\vdots$

The **Freq Vector of T** is

$$\vec{f_T} = \left(\frac{N_a}{N}, \frac{N_b}{N}, \cdots, \frac{N_z}{N} \right)$$

How to Tell Is-English

Given a Text T you want to tell if it's **English** or a **Shift of English**. You do not want to **read** all 26 possible shifts of T.

How to Tell Is-English

Given a Text T you want to tell if it's **English** or a **Shift of English**. You do not want to **read** all 26 possible shifts of T.

Let $\vec{f_E}$ be Freq Vector for English.
Let $\vec{f_T}$ be Freq Vector for T.

How to Tell Is-English

Given a Text T you want to tell if it's **English** or a **Shift of English**. You do not want to **read** all 26 possible shifts of T.

Let $\vec{f_E}$ be Freq Vector for English.
Let $\vec{f_T}$ be Freq Vector for T.
How to tell if $\vec{f_T}$ is **close to** $\vec{f_E}$?

How to Tell Is-English

Given a Text T you want to tell if it's **English** or a **Shift of English**. You do not want to **read** all 26 possible shifts of T.

Let $\vec{f_E}$ be Freq Vector for English.
Let $\vec{f_T}$ be Freq Vector for T.
How to tell if $\vec{f_T}$ is **close to** $\vec{f_E}$? Ideas?

How to Tell Is-English

Given a Text T you want to tell if it's **English** or a **Shift of English**. You do not want to **read** all 26 possible shifts of T.

Let $\vec{f_E}$ be Freq Vector for English.
Let $\vec{f_T}$ be Freq Vector for T.
How to tell if $\vec{f_T}$ is **close to** $\vec{f_E}$? Ideas?

- $\sum_{i=0}^{25} |f_{E,i} - f_{T,i}|$

How to Tell Is-English

Given a Text T you want to tell if it's **English** or a **Shift of English**. You do not want to **read** all 26 possible shifts of T.

Let $\vec{f_E}$ be Freq Vector for English.
Let $\vec{f_T}$ be Freq Vector for T.
How to tell if $\vec{f_T}$ is **close to** $\vec{f_E}$? Ideas?

- $\sum_{i=0}^{25} |f_{E,i} - f_{T,i}|$
- $\sum_{i=0}^{25} (f_{E,i} - f_{T,i})^2$

How to Tell Is-English

Given a Text T you want to tell if it's **English** or a **Shift of English**. You do not want to **read** all 26 possible shifts of T.

Let $\vec{f_E}$ be Freq Vector for English.
Let $\vec{f_T}$ be Freq Vector for T.
How to tell if $\vec{f_T}$ is **close to** $\vec{f_E}$? Ideas?

- $\sum_{i=0}^{25} |f_{E,i} - f_{T,i}|$
- $\sum_{i=0}^{25} (f_{E,i} - f_{T,i})^2$

These are good ideas but do not seem to work.

Vorlons Alphabet: $\{a, b, c, d\}$

- Vorlon freq shifted by 0 is $\vec{f_0} = \{0.5, 0.3, 0.1, 0.1\}$.
- Vorlon freq shifted by 1 is $\vec{f_1} = \{0.1, 0.5, 0.3, 0.1\}$.
- Vorlon freq shifted by 2 is $\vec{f_2} = \{0.1, 0.1, 0.5, 0.3\}$.
- Vorlon freq shifted by 3 is $\vec{f_3} = \{0.3, 0.1, 0.1, 0.5\}$.

Vorlons Alphabet: $\{a, b, c, d\}$

- Vorlon freq shifted by 0 is $\vec{f_0} = \{0.5, 0.3, 0.1, 0.1\}$.
- Vorlon freq shifted by 1 is $\vec{f_1} = \{0.1, 0.5, 0.3, 0.1\}$.
- Vorlon freq shifted by 2 is $\vec{f_2} = \{0.1, 0.1, 0.5, 0.3\}$.
- Vorlon freq shifted by 3 is $\vec{f_3} = \{0.3, 0.1, 0.1, 0.5\}$.

$$\vec{f_0} \cdot \vec{f_0} = 0.5^2 + 0.3^2 + 0.1^2 + 0.1^2 = 0.36$$

Vorlons Alphabet: $\{a, b, c, d\}$

- Vorlon freq shifted by 0 is $\vec{f_0} = \{0.5, 0.3, 0.1, 0.1\}$.
- Vorlon freq shifted by 1 is $\vec{f_1} = \{0.1, 0.5, 0.3, 0.1\}$.
- Vorlon freq shifted by 2 is $\vec{f_2} = \{0.1, 0.1, 0.5, 0.3\}$.
- Vorlon freq shifted by 3 is $\vec{f_3} = \{0.3, 0.1, 0.1, 0.5\}$.

$\vec{f_0} \cdot \vec{f_0} = 0.5^2 + 0.3^2 + 0.1^2 + 0.1^2 = 0.36$

$\vec{f_0} \cdot \vec{f_1} = 0.5 * 0.1 + 0.3 * 0.5 + 0.1 * 0.3 + 0.1 * 0.1 = 0.24$

Vorlons Alphabet: $\{a, b, c, d\}$

- Vorlon freq shifted by 0 is $\vec{f_0} = \{0.5, 0.3, 0.1, 0.1\}$.
- Vorlon freq shifted by 1 is $\vec{f_1} = \{0.1, 0.5, 0.3, 0.1\}$.
- Vorlon freq shifted by 2 is $\vec{f_2} = \{0.1, 0.1, 0.5, 0.3\}$.
- Vorlon freq shifted by 3 is $\vec{f_3} = \{0.3, 0.1, 0.1, 0.5\}$.

$$\vec{f_0} \cdot \vec{f_0} = 0.5^2 + 0.3^2 + 0.1^2 + 0.1^2 = 0.36$$
$$\vec{f_0} \cdot \vec{f_1} = 0.5 * 0.1 + 0.3 * 0.5 + 0.1 * 0.3 + 0.1 * 0.1 = 0.24$$
$$\vec{f_0} \cdot \vec{f_2} = 0.5 * 0.1 + 0.3 * 0.1 + 0.1 * 0.5 + 0.1 * 0.3 = 0.16$$

Vorlons Alphabet: $\{a, b, c, d\}$

- Vorlon freq shifted by 0 is $\vec{f_0} = \{0.5, 0.3, 0.1, 0.1\}$.
- Vorlon freq shifted by 1 is $\vec{f_1} = \{0.1, 0.5, 0.3, 0.1\}$.
- Vorlon freq shifted by 2 is $\vec{f_2} = \{0.1, 0.1, 0.5, 0.3\}$.
- Vorlon freq shifted by 3 is $\vec{f_3} = \{0.3, 0.1, 0.1, 0.5\}$.

$$\vec{f_0} \cdot \vec{f_0} = 0.5^2 + 0.3^2 + 0.1^2 + 0.1^2 = 0.36$$
$$\vec{f_0} \cdot \vec{f_1} = 0.5 * 0.1 + 0.3 * 0.5 + 0.1 * 0.3 + 0.1 * 0.1 = 0.24$$
$$\vec{f_0} \cdot \vec{f_2} = 0.5 * 0.1 + 0.3 * 0.1 + 0.1 * 0.5 + 0.1 * 0.3 = 0.16$$
$$\vec{f_0} \cdot \vec{f_3} = 0.5 * 0.3 + 0.3 * 0.1 + 0.1 * 0.1 + 0.1 * 0.5 = 0.24$$

Vorlons Alphabet: $\{a, b, c, d\}$

- Vorlon freq shifted by 0 is $\vec{f_0} = \{0.5, 0.3, 0.1, 0.1\}$.
- Vorlon freq shifted by 1 is $\vec{f_1} = \{0.1, 0.5, 0.3, 0.1\}$.
- Vorlon freq shifted by 2 is $\vec{f_2} = \{0.1, 0.1, 0.5, 0.3\}$.
- Vorlon freq shifted by 3 is $\vec{f_3} = \{0.3, 0.1, 0.1, 0.5\}$.

$\vec{f_0} \cdot \vec{f_0} = 0.5^2 + 0.3^2 + 0.1^2 + 0.1^2 = 0.36$

$\vec{f_0} \cdot \vec{f_1} = 0.5 * 0.1 + 0.3 * 0.5 + 0.1 * 0.3 + 0.1 * 0.1 = 0.24$

$\vec{f_0} \cdot \vec{f_2} = 0.5 * 0.1 + 0.3 * 0.1 + 0.1 * 0.5 + 0.1 * 0.3 = 0.16$

$\vec{f_0} \cdot \vec{f_3} = 0.5 * 0.3 + 0.3 * 0.1 + 0.1 * 0.1 + 0.1 * 0.5 = 0.24$

Upshot

$\vec{f_0} \cdot \vec{f_0}$ **big**

For $i \in \{1, 2, 3\}$, $\vec{f_0} \cdot \vec{f_i}$ **small**

English Alphabet: $\{a, \ldots, z\}$

- English freq shifted by 0 is $\vec{f_0}$
- For $1 \le i \le 25$, English freq shifted by i is $\vec{f_i}$.

English Alphabet: $\{a, \ldots, z\}$

- English freq shifted by 0 is $\vec{f_0}$
- For $1 \le i \le 25$, English freq shifted by i is $\vec{f_i}$.

$\vec{f_0} \cdot \vec{f_0} \sim 0.065$

English Alphabet: $\{a, \ldots, z\}$

- English freq shifted by 0 is $\vec{f_0}$
- For $1 \le i \le 25$, English freq shifted by i is $\vec{f_i}$.

$\vec{f_0} \cdot \vec{f_0} \sim 0.065$

$\max_{1 \le i \le 25} \vec{f_0} \cdot \vec{f_i} \sim 0.038$

English Alphabet: $\{a, \ldots, z\}$

- English freq shifted by 0 is $\vec{f_0}$
- For $1 \le i \le 25$, English freq shifted by i is $\vec{f_i}$.

$\vec{f_0} \cdot \vec{f_0} \sim 0.065$

$\max_{1 \le i \le 25} \vec{f_0} \cdot \vec{f_i} \sim 0.038$

These numbers are **empirical**, not **mathematical.**

English Alphabet: $\{a, \ldots, z\}$

- English freq shifted by 0 is $\vec{f_0}$
- For $1 \le i \le 25$, English freq shifted by i is $\vec{f_i}$.

$\vec{f_0} \cdot \vec{f_0} \sim 0.065$

$\max_{1 \le i \le 25} \vec{f_0} \cdot \vec{f_i} \sim 0.038$

These numbers are **empirical**, not **mathematical.**

Upshot
$\vec{f_0} \cdot \vec{f_0}$ **big**
For $i \in \{1, \ldots, 25\}$, $\vec{f_0} \cdot \vec{f_i}$ **small**

English Alphabet: $\{a, \ldots, z\}$

- English freq shifted by 0 is $\vec{f_0}$
- For $1 \le i \le 25$, English freq shifted by i is $\vec{f_i}$.

$\vec{f_0} \cdot \vec{f_0} \sim 0.065$

$\max_{1 \le i \le 25} \vec{f_0} \cdot \vec{f_i} \sim 0.038$

These numbers are **empirical**, not **mathematical.**

Upshot
$\vec{f_0} \cdot \vec{f_0}$ **big**
For $i \in \{1, \ldots, 25\}$, $\vec{f_0} \cdot \vec{f_i}$ **small**

Henceforth $\vec{f_0}$ will be denoted $\vec{f_E}$. E is for *English*

Is English

We describe a way to tell if a text **Is English** that we will use throughout this course.

Is English

We describe a way to tell if a text **Is English** that we will use throughout this course.

1. Input(T) a text
2. Compute $\vec{f_T}$, the freq vector for T
3. Compute $\vec{f_E} \cdot \vec{f_T}$. If ≈ 0.065 then output YES, else NO

Is English

We describe a way to tell if a text **Is English** that we will use throughout this course.

1. Input(T) a text
2. Compute $\vec{f_T}$, the freq vector for T
3. Compute $\vec{f_E} \cdot \vec{f_T}$. If ≈ 0.065 then output YES, else NO

Note: What if $\vec{f_T} \cdot \vec{f_E} = 0.061$?

Is English

We describe a way to tell if a text **Is English** that we will use throughout this course.

1. Input(T) a text
2. Compute $\vec{f_T}$, the freq vector for T
3. Compute $\vec{f_E} \cdot \vec{f_T}$. If ≈ 0.065 then output YES, else NO

Note: What if $\vec{f_T} \cdot \vec{f_E} = 0.061$?

If shift cipher used, this will **never** happen. (Why Never? Next Slide.)

Is English

We describe a way to tell if a text **Is English** that we will use throughout this course.

1. Input(T) a text
2. Compute $\vec{f_T}$, the freq vector for T
3. Compute $\vec{f_E} \cdot \vec{f_T}$. If ≈ 0.065 then output YES, else NO

Note: What if $\vec{f_T} \cdot \vec{f_E} = 0.061$?

If shift cipher used, this will **never** happen. (Why Never? Next Slide.)

If 'simple' ciphers used, this will **never** happen.

Is English

We describe a way to tell if a text **Is English** that we will use throughout this course.

1. Input(T) a text
2. Compute $\vec{f_T}$, the freq vector for T
3. Compute $\vec{f_E} \cdot \vec{f_T}$. If ≈ 0.065 then output YES, else NO

Note: What if $\vec{f_T} \cdot \vec{f_E} = 0.061$?

If shift cipher used, this will **never** happen. (Why Never? Next Slide.)

If 'simple' ciphers used, this will **never** happen.

If 'difficult' cipher used, we may use different IS-ENGLISH function.

Why Never?

Let T be a text shifted by $s \in \{1, \ldots, 25\}$.

Why Never?

Let T be a text shifted by $s \in \{1, \dots, 25\}$.

Quite likely e maps to a letter that does not occur that often.

Why Never?

Let T be a text shifted by $s \in \{1, \ldots, 25\}$.

Quite likely e maps to a letter that does not occur that often.

Quite likely many high-freq letters get mapped to letters that occur less often.

Why Never?

Let T be a text shifted by $s \in \{1, \ldots, 25\}$.

Quite likely e maps to a letter that does not occur that often.

Quite likely many high-freq letters get mapped to letters that occur less often.

Quite likely many low-freq letters get mapped to letters that occur more often.

Why Never?

Let T be a text shifted by $s \in \{1, \ldots, 25\}$.

Quite likely e maps to a letter that does not occur that often.

Quite likely many high-freq letters get mapped to letters that occur less often.

Quite likely many low-freq letters get mapped to letters that occur more often.

So $\vec{f_E} \cdot \vec{f_T}$ will be low since the large components of f_E are mult by the low components of f_T and vice versa.

Why Never?

Let T be a text shifted by $s \in \{1, \ldots, 25\}$.

Quite likely e maps to a letter that does not occur that often.

Quite likely many high-freq letters get mapped to letters that occur less often.

Quite likely many low-freq letters get mapped to letters that occur more often.

So $\vec{f_E} \cdot \vec{f_T}$ will be low since the large components of f_E are mult by the low components of f_T and vice versa.

The numbers (0.065,0.038) are **not mathematical** and are the empirical parameters for English. Different languages will have different parameters, but all will have a large gap between shifted and non-shifted.

Cracking Shift Cipher

- Given T a long text that you KNOW was coded by shift.

Cracking Shift Cipher

- Given T a long text that you KNOW was coded by shift.
- For $s = 0$ to 25
  - Create T_s which is T shifted by s.

Cracking Shift Cipher

- Given T a long text that you KNOW was coded by shift.
- For $s = 0$ to 25
  - Create T_s which is T shifted by s.
  - If **Is English**(T_s)=YES then output T_s and stop. Else try next value of s.

Cracking Shift Cipher

- Given T a long text that you KNOW was coded by shift.
- For $s = 0$ to 25
  - Create T_s which is T shifted by s.
  - If **Is English**(T_s)=YES then output T_s and stop. Else try next value of s.

Note: Only one value of s will cause **Is English**(T_s) ~ 0.065

Speeding Up Cracking of Shift Cipher

In the last slide we tried *all* shifts in order.

Speeding Up Cracking of Shift Cipher

In the last slide we tried *all* shifts in order.
Can do better: Most common letter is probably e. If not then 2nd most. . . .

Speeding Up Cracking of Shift Cipher

In the last slide we tried *all* shifts in order.
Can do better: Most common letter is probably e. If not then 2nd most....

- Given T a long text that you KNOW was coded by shift.

Speeding Up Cracking of Shift Cipher

In the last slide we tried *all* shifts in order.
Can do better: Most common letter is probably e. If not then 2nd most. . ..

- Given T a long text that you KNOW was coded by shift.
- Find frequencies of all letters, form vector $\vec{f}$.

Speeding Up Cracking of Shift Cipher

In the last slide we tried *all* shifts in order.
Can do better: Most common letter is probably *e*. If not then 2nd most....

- Given T a long text that you KNOW was coded by shift.
- Find frequencies of all letters, form vector $\vec{f}$.
- Sort vector. So most common letter is σ_0, next is σ_1, etc.

Speeding Up Cracking of Shift Cipher

In the last slide we tried *all* shifts in order.
Can do better: Most common letter is probably e. If not then 2nd most. . ..

- Given T a long text that you KNOW was coded by shift.
- Find frequencies of all letters, form vector $\vec{f}$.
- Sort vector. So most common letter is σ_0, next is σ_1, etc.
- For $i = 0$ to 25

Speeding Up Cracking of Shift Cipher

In the last slide we tried *all* shifts in order.
Can do better: Most common letter is probably e. If not then 2nd most....

- Given T a long text that you KNOW was coded by shift.
- Find frequencies of all letters, form vector $\vec{f}$.
- Sort vector. So most common letter is σ_0, next is σ_1, etc.
- For $i = 0$ to 25
  - Create T_i which is T shifted as if σ_i maps to e.

Speeding Up Cracking of Shift Cipher

In the last slide we tried *all* shifts in order.
Can do better: Most common letter is probably e. If not then 2nd most. . ..

- Given T a long text that you KNOW was coded by shift.
- Find frequencies of all letters, form vector $\vec{f}$.
- Sort vector. So most common letter is σ_0, next is σ_1, etc.
- For $i = 0$ to 25
  - Create T_i which is T shifted as if σ_i maps to e.
  - Compute $\vec{g}$, the freq vector for T_i.

Speeding Up Cracking of Shift Cipher

In the last slide we tried *all* shifts in order.
Can do better: Most common letter is probably e. If not then 2nd most. . ..

- Given T a long text that you KNOW was coded by shift.
- Find frequencies of all letters, form vector $\vec{f}$.
- Sort vector. So most common letter is σ_0, next is σ_1, etc.
- For $i = 0$ to 25
  - Create T_i which is T shifted as if σ_i maps to e.
  - Compute $\vec{g}$, the freq vector for T_i.
  - Compute $\vec{g} \cdot \vec{f_E}$. If ≈ 0.065 then stop: T_i is your text. Else try next value of i.

Speeding Up Cracking of Shift Cipher

In the last slide we tried *all* shifts in order.
Can do better: Most common letter is probably e. If not then 2nd most. . ..

- Given T a long text that you KNOW was coded by shift.
- Find frequencies of all letters, form vector $\vec{f}$.
- Sort vector. So most common letter is σ_0, next is σ_1, etc.
- For $i = 0$ to 25
  - Create T_i which is T shifted as if σ_i maps to e.
  - Compute $\vec{g}$, the freq vector for T_i.
  - Compute $\vec{g} \cdot \vec{f_E}$. If ≈ 0.065 then stop: T_i is your text. Else try next value of i.

Note Quite likely to succeed in the first try, or at least very early.
Why Would it Not Succeed on First Try? Short Text, strange text, or the person encoding does not like the letter **e**.