Some Statistical Properties of Languages

Klaus Pommerening
Fachbereich Physik, Mathematik, Informatik
der Johannes-Gutenberg-Universität
Saarstraße 21
D-55099 Mainz

November 25, 1999
Expanded English version October 20, 2013
Last change August 25, 2014

In this section we study certain statistical properties of texts and languages. These help to answer questions such as:

- Does a given text belong to a certain language? Can we derive an algorithm for automatically distinguishing valid plaintext from random noise? This is one of the central problems of cryptanalysis.

- Do two given texts belong to the same language?

- Can we decide these questions also for encrypted texts? Which properties of texts are invariant under certain encryption procedures? Can we distinguish encrypted plaintext from random noise?

- Is a given ciphertext monoalphabetically encrypted? Or polyalphabetically with periodic repetition of alphabets? If so, what is the period?

- How to adjust the alphabets in the columns of a periodic cipher? Or of several ciphertexts encrypted with the same key and correctly aligned in depth?

To get useful information on these questions we define some statistical reference numbers and analyze the distributions of these numbers. The main methods for determining reference values are:

- **Exact calculation.** This works for artificial languages with exact descriptions and for simple distributions, but for natural languages it is hopeless.

- **Modelling.** We try to build a simplified model of a language, based on letter frequencies etc. and hope that the model on the one hand approximates the statistical properties of the language closely enough, and on the other hand is simple enough that it allows the calculation of the relevant statistics. The two most important models are:

  - the computer scientific model that regards a language as a fixed set of strings with certain statistical properties,
  - the stochastic model that regards a language as a finite stationary MARKOV process. This essentially goes back to SHANNON in the 1940s after at least 20 years of naive but successful use by the FRIEDMAN school.

- **Simulation.** We take a large sample of texts from a language and determine the characteristic reference numbers by counting. In this way we find empirical approximations to the distributions and their characteristic properties.

The main results of this section go back to FRIEDMAN, KULLBACK, and SINKOV in the 1920s and 1930s. However the statistical methodology has since developed and now provides a uniform conceptual framework for statistical tests and decisions.

For a systematic treatment of the first two questions above a good reference is [4, 5]. An elementary but mathematically sound introduction to probability and statistics is [6], whereas [7] and [8] use an elementary "naive" approach to probability theory.

1 Recognizing Plaintext: Friedman's Most-Frequent-Letters Test

We begin with the first question: *Does a given text belong to a certain language?* Friedman gave a quite simple procedure for distinguishing valid text from random noise that works surprisingly well, even for short texts. Besides it makes a smooth introduction to statistical test theory.

Friedman's **Procedure**

Assume we are given a string of letters and want to decide whether it is a part of a meaningful text (in a given language, say English), or whether it is random gibberish. Our first contact with this problem was the exhaustion attack against the simple shift cipher that produced 26 strings, exactly one of which represented the correct solution. Cherry-picking it was easy by visual inspection. But for automating this decision procedure we would prefer a quantitative criterion.

Such a criterion was proposed by Friedman in Riverbank Publication No. 16 from 1918 [3]. The procedure is

1. Identify a set of most frequent letters from the target language. For English take `ETOANIRSHD` that make up 73.9% of an average English text but only $10/26 \approx 38.5\%$ of a random text.

2. Count the cumulative frequencies of these most-frequent letters for each of the candidate strings.

3. Pick the string with the highest score. If this doesn't work, also consider the next highest scores.

Example. For the Caesar example in Section 1.3 the scores are in Table 1. We immediately see that the correct solution `CAESAR` has the highest score (even if this is not a genuine English word).

The example shows that Friedman's procedure seems to work well even for quite short strings. To confirm this observation we analyze the distribution of the Most-Frequent-Letters scores—in short **MFL scores**—for strings of natural languages and for random strings. First we consider this task from a theoretic viewpoint, then we also perform some empirical evaluations.

The distribution of MFL Scores

Consider strings of length r over an alphabet Σ whose letters are independently drawn with certain probabilities, the letter $s \in \Sigma$ with probability p_s. Let $\mathcal{M} \subseteq \Sigma$ be a subset and $p = \sum_{s\in\mathcal{M}} p_s$ be the cumulative probability

Table 1: FRIEDMAN *scores for the exhausion of a shift cipher*

FDHVDU	3		OMQEMD	3		XVZNVM	1	
GEIWEV	3		PNRFNE	4	<---	YWAOWN	3	
HFJXFW	1		QOSGOF	3		ZXBPXO	1	
IGKYGX	1		RPTHPG	3		AYCQYP	1	
JHLZHY	2		SQUIQH	3		BZDRZQ	2	
KIMAIZ	3		TRVJRI	4	<---	CAESAR	5	<===
LJNBJA	2		USWKSJ	2		DBFTBS	3	
MKOCKB	1		VTXLTK	2		ECGUCT	2	
NLPDLC	2		WUYMUL	0				

of the letters in $\mathcal{M}$. The **MFL score** of a string $a = (a_1, \ldots, a_r) \in \Sigma^r$ with respect to $\mathcal{M}$ is

$$N_{\mathcal{M}}(a) = \#\{i \mid a_i \in \mathcal{M}\}.$$

To make the scores for different lengths comparable we also introduce the **MFL rate**

$$\nu_{\mathcal{M}}(a) = \frac{N_{\mathcal{M}}(a)}{r}.$$

The MFL rate defines a function

$$\nu_{\mathcal{M}} \colon \Sigma^* \longrightarrow \mathbb{Q}.$$

(Set $\nu_{\mathcal{M}}(\emptyset) = 0$ for the empty string $\emptyset$ of length 0.)

The distribution of scores is binomial, that is the probability that a string $a \in \Sigma^r$ contains exactly k letters from $\mathcal{M}$ is given by the binomial distribution

$$P(a \in \Sigma^r \mid N_{\mathcal{M}}(a) = k) = B_{r,p}(k) = \binom{r}{k} \cdot p^k \cdot (1-p)^{r-k}.$$

Random strings. We take the 26 letter alphabet `A...Z` and pick a subset $\mathcal{M}$ of 10 elements. Then $p = 10/26 \approx 0.385$, and this is also the expected value of the MFL rate $\nu_{\mathcal{M}}(a)$ for $a \in \Sigma^*$. For strings of length 10 we get the two middle columns of Table 2.

English strings. Assuming that the letters of an English string are independent is certainly only a rough approximation to the truth, but the best we can do for the moment, and, as it turns out, not too bad. Then we take $\mathcal{M} = \{\texttt{ETOANIRSHD}\}$ and $p = 0.739$ and get the rightmost two columns of Table 2.

Table 2: *Binomial distribution for $r = 10$. The columns headed "Total" contain the accumulated probabilities.*

		$p = 0.385$ **(Random)**		$p = 0.739$ **(English)**	
Score	Coefficient	Probability	Total	Probability	Total
0	$B_{10,p}(0)$	0.008	0.008	0.000	0.000
1	$B_{10,p}(1)$	0.049	0.056	0.000	0.000
2	$B_{10,p}(2)$	0.137	0.193	0.001	0.001
3	$B_{10,p}(3)$	0.228	0.422	0.004	0.005
4	$B_{10,p}(4)$	0.250	**0.671**	0.020	**0.024**
5	$B_{10,p}(5)$	0.187	0.858	0.067	0.092
6	$B_{10,p}(6)$	0.097	0.956	0.159	0.250
7	$B_{10,p}(7)$	0.035	0.991	0.257	0.507
8	$B_{10,p}(8)$	0.008	0.999	0.273	0.780
9	$B_{10,p}(9)$	0.001	1.000	0.172	0.951
10	$B_{10,p}(10)$	0.000	1.000	0.049	1.000

A Statistical Decision Procedure

What does this table tell us? Let us interpret the cryptanalytic task as a decision problem: We set a threshold value T and decide:

- A string with score $\leq T$ is probably random. We discard it.
- A string with score $> T$ could be true plaintext. We keep it for further examination.

There are two kinds of possible errors in this decision:

1. A true plaintext has a low score. We miss it.
2. A random string has a high score. We keep it.

Example. Looking at Table 2 we are tempted to set the threshold value at $T = 4$. Then (in the long run) we'll miss 2.4% of all true plaintexts because the probability for an English 10 letter text string having an MFL score ≤ 4 is 0.024. On the other hand we'll discard only 67.1% of all random strings and erroneously keep 32.9% of them.

The lower the threshold T, the more unwanted random strings will be selected. But the higher the threshold, the more true plaintext strings will be missed. Because the distributions of the MFL scores for "Random" and "English" overlap there is no clear cutpoint that always gives the correct decision.

This is a typical situation for statistical decision problems (or **tests**). The statistician usually bounds one of the two errors by a fixed amount, usually 5% or 1%, and calls this the **error of the first kind**, denoted by α. (The complementary value $1 - \alpha$ is called the sensitivity of the test.) Then she tries to minimize the other error, the **error of the second kind**, denoted by β. The complementary value $1 - \beta$ is called the **power** (or specifity) of the test. FRIEDMAN's MFL-method, interpreted as a statistical test (for the "null hypothesis" of English text against the "alternative hypothesis" of random text), has a power of $\approx 67\%$ for English textstrings of length 10 and $\alpha = 2.4\%$. This α-value was chosen because it is the largest one below 5% that really occurs in the sixth column of Table 2.

To set up a test the statistician faces two choices. First she has to choose between "first" and "second" kind depending on the severity of the errors in the actual context. In our case she wants to bound the number of missed true plaintexts at a very low level—a missed plaintext renders the complete cryptanalysis obsolete. On the other hand keeping too many random strings increases the effort of the analysis, but this of somewhat less concern.

The second choice is the error level α. By these two choices the statistician adapts the test to the context of the decision problem.

Remark. We won't discuss the trick of raising the power by exhausting the α-level, randomizing the decision at the threshold value.

Note. There is another ("BAYESian") way to look at the decision problem. The **predictive values** give the probabilities that texts are actually what we decide them to be. If we decide "random" for texts with MFL score ≤ 4, we'll be correct for about 671 of 1000 random texts and err for 24 of 1000 English texts. This makes 695 decisions for random of which 671 are correct. The predictive value of our "random" decision is $96.5\% \approx 671/695$. The decision "English" for an MFL score > 4 will be correct for 976 of 1000 English texts and false for 329 of 1000 random texts. Hence the predictive value of the decision "English" is about $75\% \approx 976/1305$. That means that if we pick up texts (of length 10) with a score of at least 5, then (in the long run) one out of four selected texts will be random.

Other Languages: German and French

German: The ten most frequent letters are `ENIRSATDHU`. They make up 75.1% of an average German text.

French: The ten most frequent letters are `EASNTIRULO`. They make up 79.1% of an average French text.

With these values we supplement Table 2 by Table 3.

As before for English we get as conclusions for textstrings of length 10:

Table 3: *Distribution of MFL scores for $r = 10$*

	$p = 0.751$ **(German)**		$p = 0.791$ **(French)**	
Score	Probability	Total	Probability	Total
0	0.000	0.000	0.000	0.000
1	0.000	0.000	0.000	0.000
2	0.000	0.000	0.000	0.000
3	0.003	0.003	0.001	0.001
4	0.016	**0.019**	0.007	0.008
5	0.058	0.077	0.031	**0.039**
6	0.145	0.222	0.098	0.137
7	0.250	0.471	0.212	0.350
8	0.282	0.754	0.301	0.651
9	0.189	0.943	0.253	0.904
10	0.057	1.000	0.096	1.000

German: With a threshold of $T = 4$ and $\alpha = 1.9\%$ the MFL-test has a power of 67%. The predictive value for "German" is 75% $\approx 981/1310$.

French: With a threshold of $T = 5$ and $\alpha = 3.9\%$ the MFL-test has a power of 86%. The predictive value for "French" is 87% $\approx 961/1103$.

Textstrings of length 20

The distribution is given in Table 4. We conclude:

English: With a threshold of $T = 10$ and $\alpha = 1.9\%$ the MFL-test has a power of 90% and a predictive value of 91% $\approx 981/1081$.

German: With a threshold of $T = 11$ and $\alpha = 4.0\%$ the MFL-test has a power of 96% and a predictive value of 96% $\approx 960/1002$.

French: With a threshold of $T = 12$ and $\alpha = 4.1\%$ the MFL-test has a power of 98.5% and a predictive value of 98.5% $\approx 959/974$.

Table 4: *Distribution of MFL scores for $r = 20$*

	Random		**English**		**German**		**French**	
Score	Prob	Total	Prob	Total	Prob	Total	Prob	Total
0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000
2	0.005	0.005	0.000	0.000	0.000	0.000	0.000	0.000
3	0.017	0.022	0.000	0.000	0.000	0.000	0.000	0.000
4	0.045	0.067	0.000	0.000	0.000	0.000	0.000	0.000
5	0.090	0.157	0.000	0.000	0.000	0.000	0.000	0.000
6	0.140	0.297	0.000	0.000	0.000	0.000	0.000	0.000
7	0.175	0.472	0.000	0.000	0.000	0.000	0.000	0.000
8	0.178	0.650	0.001	0.001	0.001	0.001	0.000	0.000
9	0.148	0.798	0.004	0.006	0.003	0.004	0.001	0.001
10	0.102	**0.900**	0.013	**0.019**	0.010	0.013	0.003	0.004
11	0.058	**0.958**	0.034	0.053	0.026	**0.040**	0.010	0.013
12	0.027	**0.985**	0.072	0.125	0.060	0.100	0.028	**0.041**
13	0.010	0.996	0.125	0.250	0.111	0.211	0.064	0.105
14	0.003	0.999	0.178	0.428	0.168	0.379	0.121	0.226
15	0.001	1.000	0.201	0.629	0.202	0.581	0.184	0.410
16	0.000	1.000	0.178	0.807	0.191	0.772	0.217	0.627
17	0.000	1.000	0.119	0.925	0.135	0.907	0.193	0.820
18	0.000	1.000	0.056	0.981	0.068	0.975	0.122	0.942
19	0.000	1.000	0.017	0.998	0.022	0.997	0.049	0.991
20	0.000	1.000	0.002	1.000	0.003	1.000	0.009	1.000

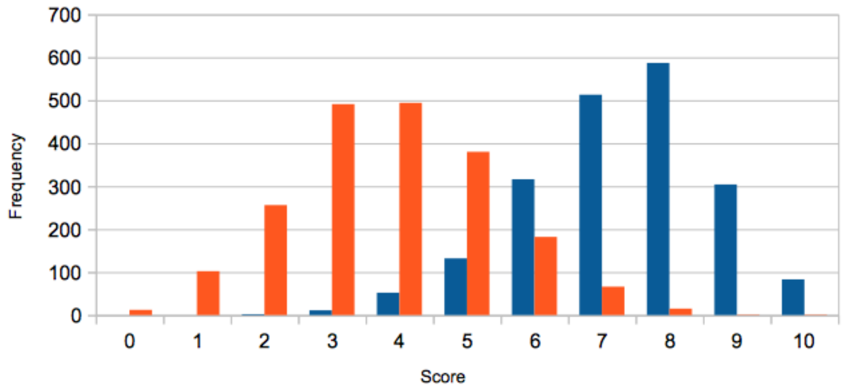

Figure 1: *MFL scores for 2000 English (blue) and random (red) text chunks of 10 letters each*

2 Empirical Results on MFL Scores

The power calculations for the tests—not the tests themselves!—relied on the independency of the letters in a string. This assumption is clearly false for natural languages. Therefore getting experimental results for the distributions of the MFL scores makes sense.

For English we take a text of 20000 letters, an extract from the Project Gutenberg etext of *Kim*, by Rudyard Kipling, `http://www.gutenberg.org/ebooks/2226`. The partial 20000 letter text is at `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Files/Kim20K.txt`. We divide this text into 2000 substrings of 10 letters each. To this set of substrings we apply the Perl script `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Perl/fritestE.pl`. The results are collected and evaluated in a spreadsheet, found at `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Files/statFriE.xls`.

We do the same for random text, constructed by taking 20000 random numbers between 0 and 25 from `random.org`, see `.../Files/rnd10E.txt`. The Perl script `.../Perl/RandOrg.pl` transforms the random numbers to text.

Figure 1 shows some characteristics of the distribution. Table 5 compares the expected and observed distributions. For random texts they match well, taking into account variations caused by drawing a sample. Also for English the observations seem to match the predicted values. The empirical values amount to a power of 68% (instead of 67%) and a predictive value of 75% (75%).

We repeat this procedure for German and French. As texts we

Table 5: *Expected and observed frequencies of MFL scores for 2000 English and 2000 random text chunks of 10 letters*

	Random		**English**	
score	expected	observed	expected	observed
0	16	12	0	0
1	98	102	0	0
2	274	256	2	2
3	456	491	8	11
4	500	494	40	52
5	374	380	134	132
6	194	182	318	316
7	70	66	514	513
8	16	15	546	587
9	2	1	344	304
10	0	1	98	83

take *Schachnovelle* by Stefan Zweig, `http://gutenberg.spiegel.de/buch/7318/1`, and *De la Terre à la Lune* by Jules Verne, `http://www.gutenberg.org/ebooks/799`. The 20000 letter extracts are in `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Files/Schach20K.txt` and `.../Files/Lune20K.txt`. We generate independent random texts, see `.../Files/rnd10D.txt` and `.../Files/rnd10F.txt`. (The random texts being independent, the observed values for random texts differ.) The Perl scripts, adapted to the differing collections of most-frequent letters, are `.../Perl/fritestD.pl` and `.../Perl/fritestF.pl`.

The results are in Figures 2 and 3, and Tables 6 and 7. The comprehensive evaluation is in the spreadsheets `.../Files/statFriD.xls` and `.../Files/statFriF.xls`.

The empirical values amount to a power of 63% (theory: 67%) and a predictive value of 75% (75%) for German, and a power of 87% (86%) and a predictive value of 88% (87%).

Exercise. Verify the calculations of powers and predictive values.

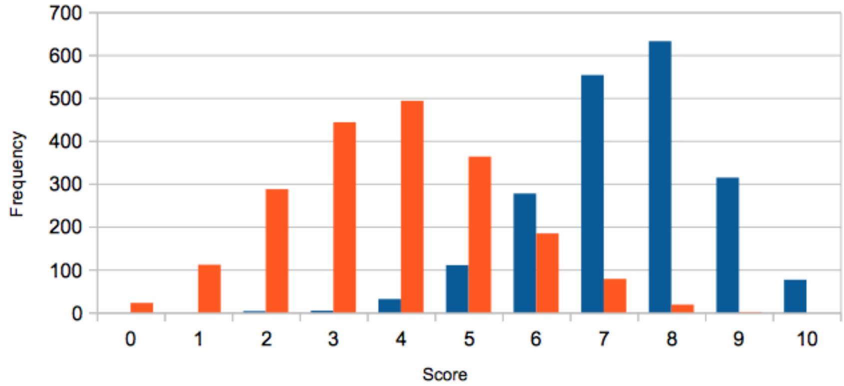

Figure 2: *MFL scores for 2000 German (blue) and random (red) text chunks of 10 letters each*

Table 6: *Expected and observed frequencies of MFL scores for 2000 German and 2000 random text chunks of 10 letters*

	Random		**German**	
score	expected	observed	expected	observed
0	16	22	0	0
1	98	111	0	0
2	274	287	0	3
3	456	443	6	4
4	500	493	32	31
5	374	363	116	110
6	194	184	290	277
7	70	78	500	553
8	16	18	564	632
9	2	1	378	314
10	0	0	114	76

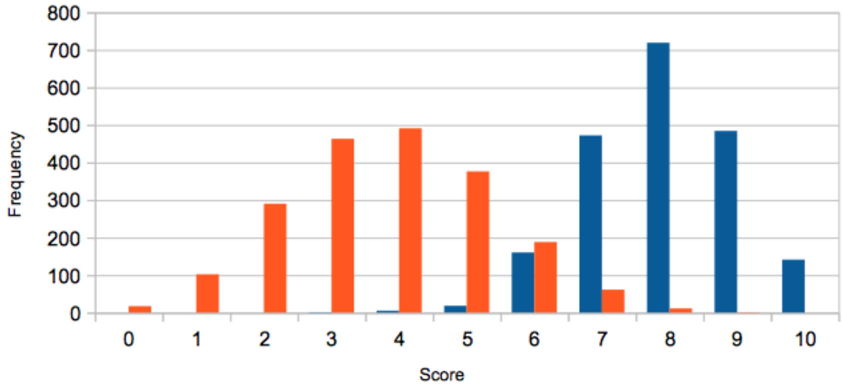

Figure 3: *MFL scores for 2000 French (blue) and random (red) text chunks of 10 letters each*

Table 7: *Expected and observed frequencies of MFL scores for 2000 French and 2000 random text chunks of 10 letters*

	Random		**French**	
score	expected	observed	expected	observed
0	16	17	0	0
1	98	102	0	0
2	274	290	0	0
3	456	463	2	1
4	500	491	14	5
5	374	376	62	18
6	194	188	196	160
7	70	61	424	472
8	16	11	602	719
9	2	1	506	484
10	0	0	192	141

3 Application to the Cryptanalysis of the Belaso Cipher

The Friedman procedure doesn't need contiguous plaintext. It also works when we pick out isolated letters from a meaningful text. In particular it works in a (semi-) automated approach to adjusting the columns of a Belaso ciphertext.

As an example we consider the ciphertext

```
UMHOD BLRHT SCWWJ NHZWB UWJCP ICOLB AWSWK CLJDO WWJOD L
```

We assume a Belaso cipher with period 4. (The Kasiski analysis yields a single significant repetition `WWJ` at a distance of 28.) The four columns (written horizontally) are

```
UDHWHUPLSLWD  MBTWZWIBWJWL  HLSJWJCAKDJ  ORCNBCOWCOO
```

For an exhaustion attack we complete the alphabets (i. e. we increment the letters step by step) and count the MFL scores for letter combinations in each row, see Table 8.

We pick up the most promising result for each column:

```
Column 1: RAETERMIPITA
Column 2: ETLOROATOBOD
Column 3: PTARERKISLR
Column 4: ADOZNOAIOAA or EHSDRSEMSEE
```

Only for column 4 we have more than one choice. However the first choice yields an ugly "plaintext". We drop it and keep

```
Col 1: RAETERMIPITA
Col 2: ETLOROATOBOD
Col 3: PTARERKISLR
Col 4: EHSDRSEMSEE
```

From this scheme we read the solution columnwise:

Repeat the last order. Errors make it impossible to read.

Exercise. What was the encryption key used in this example?

Remark. Friedman in his Riverbank Publication No. 16 [3] uses the MLF method also for polyalphabetic ciphers with non-standard, but known, primary alphabets.

Table 8: *MFL scores for the example*

UDHWHUPLSLWD	5	MBTWZWIBWJWL	2	HLSJWJCAKDJ	4	ORCNBCOWCOO	6
VEIXIVQMTMXE	5	NCUXAXJCXKXM	2	IMTKXKDBLEK	4	PSDOCDPXDPP	5
WFJYJWRNUNYF	3	ODVYBYKDYLYN	4	JNULYLECMFL	2	QTEPDEQYEQQ	5
XGKZKXSOVOZG	3	PEWZCZLEZMZO	3	KOVMZMFDNGM	3	RUFQEFRZFRR	5
YHLALYTPWPAH	5	QFXADAMFANAP	6	LPWNANGEOHN	7	SVGRFGSAGSS	6
ZIMBMZUQXQBI	2	RGYBEBNGBOBQ	4	MQXOBOHFPIO	5	TWHSGHTBHTT	8*
AJNCNAVRYRCJ	6	SHZCFCOHCPCR	5	NRYPCPIGQJP	3	UXITHIUCIUU	5
BKODOBWSZSDK	6	TIADGDPIDQDS	9*	OSZQDQJHRKQ	5	VYJUIJVDJVV	2
CLPEPCXTATEL	5	UJBEHEQJERET	7	PTARERKISLR	8*	WZKVJKWEKWW	1
DMQFQDYUBUFM	2	VKCFIFRKFSFU	3	QUBSFSLJTMS	4	XALWKLXFLXX	1
ENRGREZVCVGN	6	WLDGJGSLGTGV	3	RVCTGTMKUNT	5	YBMXLMYGMYY	0
FOSHSFAWDWHO	8*	XMEHKHTMHUHW	6	SWDUHUNLVOU	5	ZCNYMNZHNZZ	4
GPTITGBXEXIP	5	YNFILIUNIVIX	6	TXEVIVOMWPV	4	ADOZNOAIOAA	10*
HQUJUHCYFYJQ	2	ZOGJMJVOJWJY	2	UYFWJWPNXQW	1	BEPAOPBJPBB	3
IRVKVIDZGZKR	5	APHKNKWPKXKZ	3	VZGXKXQOYRX	2	CFQBPQCKQCC	0
JSWLWJEAHALS	6	BQILOLXQLYLA	3	WAHYLYRPZSY	4	DGRCQRDLRDD	7
KTXMXKFBIBMT	3	CRJMPMYRMZMB	2	XBIZMZSQATZ	4	EHSDRSEMSEE	10*
LUYNYLGCJCNU	2	DSKNQNZSNANC	8*	YCJANATRBUA	6	FITESTFNTFF	7
MVZOZMHDKDOV	5	ETLOROATOBOD	10*	ZDKBOBUSCVB	3	GJUFTUGOUGG	2
NWAPANIELEPW	7	FUMPSPBUPCPE	2	AELCPCVTDWC	4	HKVGUVHPVHH	4
OXBQBOJFMFQX	2	GVNQTQCVQDQF	3	BFMDQDWUEXD	4	ILWHVWIQWII	5
PYCRCPKGNGRY	3	HWORURDWRERG	8*	CGNEREXVFYE	5	JMXIWXJRXJJ	2
QZDSDQLHOHSZ	7	IXPSVSEXSFSH	7	DHOFSFYWGZF	4	KNYJXYKSYKK	2
RAETERMIPITA	10*	JYQTWTFYTGTI	5	EIPGTGZXHAG	5	LOZKYZLTZLL	2
SBFUFSNJQJUB	3	KZRUXUGZUHUJ	2	FJQHUHAYIBH	5	MPALZAMUAMM	3
TCGVGTOKRKVC	4	LASVYVHAVIVK	5	GKRIVIBZJCI	4	NQBMABNVBNN	5

4 Recognizing Plaintext: Sinkov's Log-Weight Test

The MFL-test is simple and efficient. Sinkov in [8] proposed a more refined test that uses the information given by all single letter frequencies, not just by separating the letters into two classes. We won't explore the power of this method but treat it only as a motivation for Section 5.

As in Section 1 we assign a probability p_s to each letter s of the alphabet Σ. We enumerate the alphabet as $(s_1, \ldots, s_n)$ and write $p_i := p_{s_i}$. For a string $a = (a_1, \ldots, a_r) \in \Sigma^r$ we denote by $N_i(a) = \#\{j \mid a_j = s_i\}$ the multiplicity of the letter s_i in a. Then for an n-tuple $k = (k_1, \ldots, k_n) \in \mathbb{N}^n$ of natural numbers the probability for a string a to have multiplicities exactly given by k follows the multinomial distribution:

$$P(a \in \Sigma^r \mid N_i(a) = k_i \text{ for all } i = 1, \ldots, n) = \frac{r!}{k_1! \cdots k_n!} \cdot p_1^{k_1} \cdots p_n^{k_n}.$$

The Log-Weight (LW) Score

A heuristic derivation of the LW-score of a string $a \in \Sigma^r$ considers the "null hypothesis" (H_0): *a belongs to a given language with letter probabilities p_i*, and the "alternative hypothesis" (H_1): *a is a random string.* The probabilities for a having k as its set of multiplicities if (H_1) or (H_0) is true, are (in a somewhat sloppy notation)

$$P(k \mid \mathrm{H}_1) = \frac{r!}{k_1! \cdots k_n!} \cdot \frac{1}{n^r}, \quad P(k \mid \mathrm{H}_0) = \frac{r!}{k_1! \cdots k_n!} \cdot p_1^{k_1} \cdots p_n^{k_n}.$$

The quotient of these two values, the "likelihood ratio"

$$\lambda(k) = \frac{P(k \mid \mathrm{H}_0)}{P(k \mid \mathrm{H}_1)} = n^r \cdot p_1^{k_1} \cdots p_n^{k_n} ,$$

makes a good score for deciding between (H_0) and (H_1).

> Usually one takes the reciprocal value, that is H_1 in the numerator, and H_0 in the denominator. We deviate from this convention because we want to have the score large for true texts and small for random texts.

For convenience one considers the logarithm (to any base) of this number:

$$\log \lambda(k) = r \log n + \sum_{i=1}^{n} k_i \cdot \log p_i.$$

Table 9: *Log weights of the letters for English (base-10 logarithms)*

s	A	B	C	D	E	F	G
$1000p_s$	82	15	28	43	127	22	20
Log weight	1.9	1.2	1.4	1.6	2.1	1.3	1.3
s	H	I	J	K	L	M	N
$1000p_s$	61	70	2	8	40	24	67
Log weight	1.8	1.8	0.3	0.9	1.6	1.4	1.8
s	O	P	Q	R	S	T	U
$1000p_s$	75	19	1	60	63	91	28
Log weight	1.9	1.3	0.0	1.8	1.8	1.9	1.4
s	V	W	X	Y	Z		
$1000p_s$	10	23	1	20	1		
Log weight	1.0	1.4	0.0	1.3	0.0		

(We assume all $p_i > 0$, otherwise we would omit s_i from our alphabet.) Noting that the summand $r \log n$ is the same for all $a \in \Sigma^r$ one considers

$$\log \lambda(k) - r \log n = \sum_{i=1}^{n} k_i \cdot \log p_i = \sum_{j=1}^{r} \log p_{a_j}.$$

Because $0 < p_i < 1$ the summands are negative. Adding a constant doesn't affect the use of this score, so finally we define Sinkov's **Log-Weight (LW) score** as

$$S_1(a) := \sum_{i=1}^{n} k_i \cdot \log(1000 \cdot p_i) = \sum_{j=1}^{r} \log(1000 \cdot p_{a_j}) = r \cdot \log 1000 + \sum_{j=1}^{r} \log p_{a_j}.$$

The numbers $\log(1000 \cdot p_i)$ are the "log weights". More frequent letters have higher weights. Table 9 gives the weights for the English alphabet with base-10 logarithms (so $\log 1000 = 3$). The MFL-method in contrast uses the weights 1 for `ETOANIRSHD`, and 0 else.

Note that the definition of the LW score doesn't depend on its heuristic motivation. Just take the weights given in Table 9 and use them for the definition of S_1.

Examples

We won't analyze the LW-method in detail, but rework the examples from Section 1. The LW scores for the Caesar example are in Table 10.

The correct solution stands out clearly, the order of the non-solutions is somewhat permuted compared with the MFL score.

Table 10: *LW scores for the exhausion of a shift cipher*

FDHVDU	8.7		OMQEMD	8.4		XVZNVM	5.2	
GEIWEV	9.7		PNRFNE	10.1	<---	YWAOWN	9.7	
HFJXFW	6.1		QOSGOF	8.2		ZXBPXO	4.4	
IGKYGX	6.6		RPTHPG	9.4		AYCQYP	7.2	
JHLZHY	6.8		SQUIQH	6.8		BZDRZQ	4.6	
KIMAIZ	7.8		TRVJRI	8.6		CAESAR	10.9	<===
LJNBJA	7.1		USWKSJ	7.6		DBFTBS	9.0	
MKOCKB	7.7		VTXLTK	7.3		ECGUCT	9.5	
NLPDLC	9.3		WUYMUL	8.5				

For the period-4 example the LW scores are in Tables 11 to 14. The method unambiguously picks the correct solution except for column 3 where the top score occurs twice.

In summary the examples show no clear advantage of the LW-method over the MFL-method, notwithstanding the higher granularity of the information used to compute the scores.

As for MFL scores we might define the LW rate as the quotient of the LW score be the length of the string. This makes the values for strings of different lengths comparable.

Table 11: *LW scores for column 1 of a period 4 cipher*

UDHWHUPLSLWD	18.7		DMQFQDYUBUFM	13.9		MVZOZMHDKDOV	14.5	
VEIXIVQMTMXE	14.5		ENRGREZVCVGN	17.4		NWAPANIELEPW	20.4	<--
WFJYJWRNUNYF	15.4		FOSHSFAWDWHO	19.9		OXBQBOJFMFQX	10.5	
XGKZKXSOVOZG	11.0		GPTITGBXEXIP	15.9		PYCRCPKGNGRY	16.9	
YHLALYTPWPAH	19.1		HQUJUHCYFYJQ	12.3		QZDSDQLHOHSZ	13.9	
ZIMBMZUQXQBI	10.2		IRVKVIDZGZKR	13.9		RAETERMIPITA	21.7	<==
AJNCNAVRYRCJ	16.7		JSWLWJEAHALS	17.9		SBFUFSNJQJUB	13.8	
BKODOBWSZSDK	16.2		KTXMXKFBIBMT	13.9		TCGVGTOKRKVC	16.7	
CLPEPCXTATEL	18.5		LUYNYLGCJCNU	16.6				

Exercise. Give a more detailed analysis of the distribution of the LW scores for English and for random texts (with “English” weights). You may use the Perl script `LWscore.pl` in the directory `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Perl/`.

Table 15 gives log weights for German and French.

Table 12: *LW scores for column 2 of a period 4 cipher*

MBTWZWIBWJWL	15.0		VKCFIFRKFSFU	16.2		ETLOROATOBOD	21.6	<==
NCUXAXJCXKXM	10.5		WLDGJGSLGTGV	16.4		FUMPSPBUPCPE	17.2	
ODVYBYKDYLYN	16.8		XMEHKHTMHUHW	17.7		GVNQTQCVQDQF	11.3	
PEWZCZLEZMZO	13.2		YNFILIUNIVIX	17.4		HWORURDWRERG	20.1	<--
QFXADAMFANAP	16.3		ZOGJMJVOJWJY	11.4		IXPSVSEXSFSH	16.5	
RGYBEBNGBOBQ	16.3		APHKNKWPKXKZ	13.1		JYQTWTFYTGTI	16.3	
SHZCFCOHCPCR	17.3		BQILOLXQLYLA	14.5		KZRUXUGZUHUJ	11.7	
TIADGDPIDQDS	18.2		CRJMPMYRMZMB	14.7		LASVYVHAVIVK	17.0	
UJBEHEQJERET	17.1		DSKNQNZSNANC	16.6				

Table 13: *LW scores for column 3 of a period 4 cipher*

HLSJWJCAKDJ	13.3		QUBSFSLJTMS	14.5		ZDKBOBUSCVB	13.6	
IMTKXKDBLEK	14.3		RVCTGTMKUNT	16.7		AELCPCVTDWC	17.0	
JNULYLECMFL	15.8		SWDUHUNLVOU	17.1		BFMDQDWUEXD	13.6	
KOVMZMFDNGM	14.0		TXEVIVOMWPV	14.8		CGNEREXVFYE	16.2	
LPWNANGEOHN	18.7	<-	UYFWJWPNXQW	11.6		DHOFSFYWGZF	15.0	
MQXOBOHFPIO	14.5		VZGXKXQOYRX	8.2		EIPGTGZXHAG	14.7	
NRYPCPIGQJP	13.6		WAHYLYRPZSY	15.5		FJQHUHAYIBH	14.6	
OSZQDQJHRKQ	10.1		XBIZMZSQATZ	10.0		GKRIVIBZJCI	13.3	
PTARERKISLR	18.7	<-	YCJANATRBUA	16.8				

Table 14: *LW scores for column 4 of a period 4 cipher*

ORCNBCOWCOO	18.0		XALWKLXFLXX	10.3		GJUFTUGOUGG	14.8	
PSDOCDPXDPP	15.1		YBMXLMYGMYY	13.5		HKVGUVHPVHH	15.1	
QTEPDEQYEQQ	12.4		ZCNYMNZHNZZ	11.3		ILWHVWIQWII	15.8	
RUFQEFRZFRR	14.6		ADOZNOAIOAA	18.5		JMXIWXJRXJJ	7.6	
SVGRFGSAGSS	17.1		BEPAOPBJPBB	14.9		KNYJXYKSYKK	11.4	
TWHSGHTBHTT	18.7	<-	CFQBPQCKQCC	10.3		LOZKYZLTZLL	12.4	
UXITHIUCIUU	16.1		DGRCQRDLRDD	16.1		MPALZAMUAMM	15.6	
VYJUIJVDJVV	11.0		EHSDRSEMSEE	20.4	<=	NQBMABNVBNN	15.1	
WZKVJKWEKWW	11.7		FITESTFNTFF	18.4				

Table 15: *Log weights of the letters for German and French (base-10 logarithms)*

s	A	B	C	D	E	F	G
German	1.8	1.3	1.4	1.7	2.2	1.2	1.5
French	1.9	1.0	1.5	1.6	2.2	1.1	1.0

s	H	I	J	K	L	M	N
German	1.6	1.9	0.5	1.2	1.5	1.4	2.0
French	0.8	1.8	0.5	0.0	1.8	1.4	1.9

s	O	P	Q	R	S	T	U
German	1.5	1.0	0.0	1.9	1.8	1.8	1.6
French	1.7	1.4	1.0	1.8	1.9	1.9	1.8

s	V	W	X	Y	Z		
German	1.0	1.2	0.0	0.0	1.0		
French	1.2	0.0	0.6	0.3	0.0		

5 Recognizing Plaintext: The Log-Weight Method for Bigrams

In the last four sections we used only the single letter frequencies of a natural language. In other words, we treated texts as sequences of independent letters. But a characteristic aspect of every natural language is how letters are combined as bigrams (letter pairs). We may hope to get good criteria for recognizing a language by evaluating the bigrams in a text. Of course this applies to contiguous text only, in particular it is useless for the polyalphabetic example of Sections 3 and 4.

In analogy with the LW score we define a **Bigram Log-Weight (BLW) score** for a string. Let p_{ij} be the probability (or average relative frequency) of the bigram $s_i s_j$ in the base language. Because these numbers are small we multiply them by 10000.

Tables containing these bigram frequencies for English, German, and French are in `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/8_Transpos/Bigrams.html`

In contrast to the single letter case we cannot avoid the case $p_{ij} = 0$: some letter pairs never occur as bigrams in a meaningful text. Therefore we count the frequencies k_{ij} of the bigrams $s_i s_j$ in a string $a \in \Sigma^r$, and define the BLW-score by the formula

$$S_2(a) := \sum_{i,j=1}^{n} k_{ij} \cdot w_{ij} \quad \text{where } w_{ij} = \begin{cases} \log(10000 \cdot p_{ij}) & \text{if } 10000 \cdot p_{ij} > 1, \\ 0 & \text{otherwise.} \end{cases}$$

Note. We implicitly set $\log 0 = 0$. This convention is not as strange as it may look at first sight: For $p_{ij} = 0$ we'll certainly have $k_{ij} = 0$, and setting $0 \cdot \log 0 = 0$ is widespread practice.

To calculate the BLW score we go through the bigrams $a_t a_{t+1}$ for $t = 1, \ldots, r-1$ and add the log weight $w_{ij} = \log(10000 \cdot p_{ij})$ of each bigram. This approach is somewhat naive because it implicitly considers the bigrams—even the overlapping ones!—as independent. This criticism doesn't mean that we are doing something mathematically wrong, but only that the usefulness of the score might be smaller than expected.

We prepare matrices for English, German, and French that contain the relative frequencies of the bigrams in the respective language. These are in the files `eng_rel.csv`, `ger_rel.csv`, `fra_rel.csv` in the directory `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Files/` as comma-separated tables. The corresponding bigram log-weights are in the files `eng_blw.csv`, `ger_blw.csv`, `fra_blw.csv`. Programs that compute BLW scores for English, German, or French are `BLWscE.pl`, `BLWscD.pl`, and `BLWscF.pl` in the Perl directory.

As an example we compute the scores for the Caesar example, see Table 16. The correct solution is evident in all three languages.

Table 16: *BLW scores for the exhaustion of a* Caesar *cipher*

BLW scores	English	German	French
FDHVDU	1.4	3.1	2.2
GEIWEV	5.8 <---	7.3 <===	4.3
HFJXFW	0.9	0.3	0.0
IGKYGX	2.2	2.1	1.3
JHLZHY	0.5	1.9	0.3
KIMAIZ	5.9 <---	5.2	4.9
LJNBJA	1.1	2.4	0.9
MKOCKB	2.7	4.2	0.8
NLPDLC	3.0	2.8	1.4
OMQEMD	3.5	3.8	3.6
PNRFNE	3.6	4.7	3.6
QOSGOF	5.8 <---	4.0	3.4
RPTHPG	4.5	2.6	2.7
SQUIQH	2.3	0.6	6.3 <---
TRVJRI	4.1	4.3	4.9
USWKSJ	3.3	3.7	2.0
VTXLTK	1.3	2.0	1.1
WUYMUL	3.1	2.9	2.7
XVZNVM	0.6	1.3	1.0
YWAOWN	5.5	2.3	0.0
ZXBPXO	0.0	0.0	0.0
AYCQYP	3.2	0.0	0.3
BZDRZQ	1.0	2.1	1.1
CAESAR	7.7 <===	7.5 <===	8.4 <===
DBFTBS	4.7	3.5	0.6
ECGUCT	5.5	3.6	5.5

6 Empirical Results on BLW Scores

The heuristic motivation of the BLW score, like for all the scores in this chapter, relies on independence assumptions that are clearly violated by natural languages. Therefore again it makes sense to get empirical results by analyzing a large sample of concrete texts. We extract 20000 letters from each of the texts *Kim*, *Schachnovelle*, and *De la Terre à la Lune*, and decompose them into 2000 chunks à 10 letters, see the files `eng10a.txt`, `ger10a.txt`, and `fra10a.txt` in the directory `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Files/`. Likewise we generate random texts, see `rnd10Ea.txt`, `rnd10Da.txt`, and `rnd10Fa.txt`. We collect the results in the spreadsheets `ER10res.xls`, `DR10res.xls`, and `FR10res.xls`.

The results are summarized in Tables 17, 18, 19, and Figures 4, 5, 6

Table 17: *Frequencies of BLW scores for English vs. random 10 letter texts*

Score	Random	English
$0 \leq x \leq 1$	32	0
$1 < x \leq 2$	97	0
$2 < x \leq 3$	187	0
$3 < x \leq 4$	254	0
$4 < x \leq 5$	324	3
$5 < x \leq 6$	301	1
$6 < x \leq 7$	271	4
$7 < x \leq 8$	216	1
$8 < x \leq 9$	156	8
$9 < x \leq 10$	77	18
$10 < x \leq 11$	49	51
$11 < x \leq 12$	25	120
$12 < x \leq 13$	6	196
$13 < x \leq 14$	3	322
$14 < x \leq 15$	2	413
$15 < x \leq 16$	0	406
$16 < x \leq 17$	0	255
$17 < x \leq 18$	0	157
$18 < x \leq 19$	0	40
$19 < x < \infty$	0	5

The empirical results for the 5%-level of the error of the first kind are as follows.

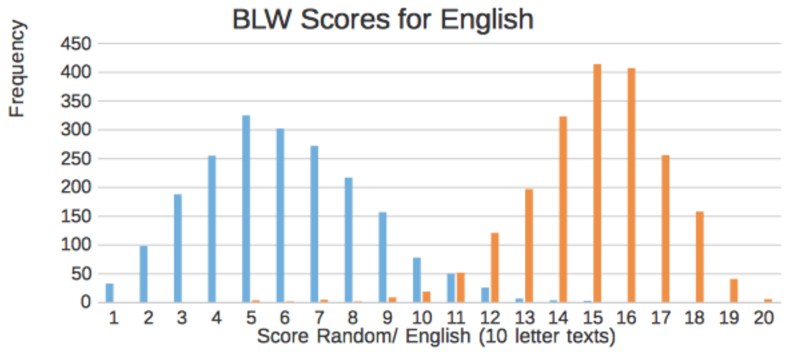

Figure 4: *BLW scores for 2000 English (red) and random (blue) text chunks of 10 letters each*

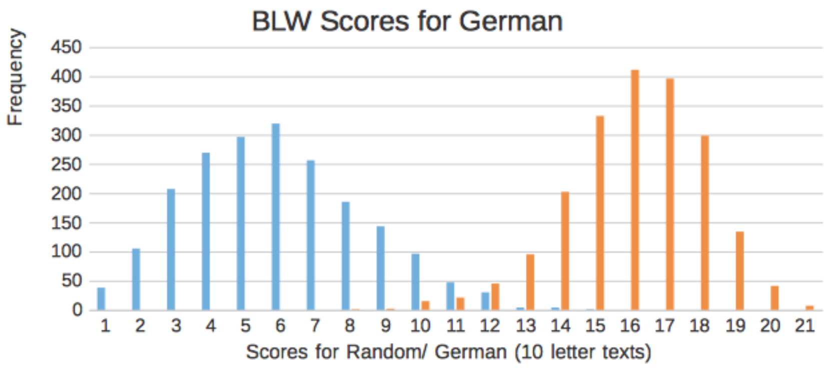

Figure 5: *BLW scores for 2000 German (red) and random (blue) text chunks of 10 letters each*

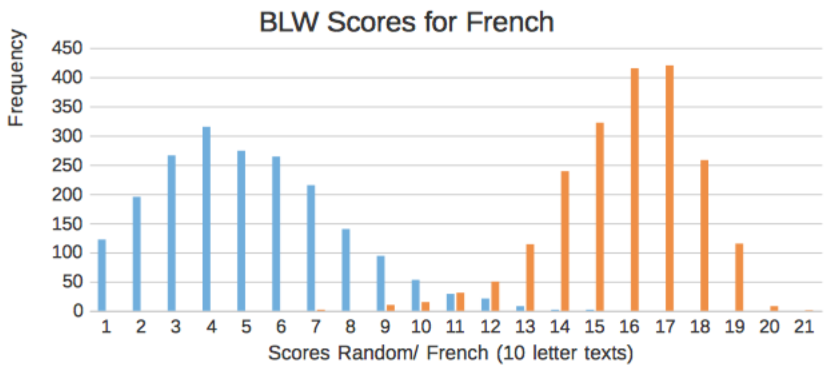

Figure 6: *BLW scores for 2000 French (red) and random (blue) text chunks of 10 letters each*

Table 18: *Frequencies of BLW scores for German vs. random texts*

Score	Random	German
$0 \le x \le 1$	38	0
$1 < x \le 2$	105	0
$2 < x \le 3$	207	0
$3 < x \le 4$	269	0
$4 < x \le 5$	296	0
$5 < x \le 6$	319	0
$6 < x \le 7$	256	0
$7 < x \le 8$	185	1
$8 < x \le 9$	143	2
$9 < x \le 10$	96	15
$10 < x \le 11$	47	21
$11 < x \le 12$	30	45
$12 < x \le 13$	4	95
$13 < x \le 14$	4	202
$14 < x \le 15$	1	332
$15 < x \le 16$	0	411
$16 < x \le 17$	0	396
$17 < x \le 18$	0	298
$18 < x \le 19$	0	134
$19 < x \le 20$	0	41
$20 < x < \infty$	0	7

English. We take the threshold value $T = 11$ for English texts. Then 86 of 2000 English scores are $\le T$, the error of the first kind is $\alpha = 86/2000 = 4.2\%$. For random texts 1964 of 2000 scores are $\le T$, the power is $1964/2000 = 99.5\%$. There are 36 random scores and 1914 English scores $> T$, the predictive value for English is $1914/1950 = 98.2\%$.

German. We take the threshold value $T = 12$ for German texts. Then 84 of 2000 German scores are $\le T$, the error of the first kind is $\alpha = 84/2000 = 4.2\%$. For random texts 1991 of 2000 scores are $\le T$, the power is $1991/2000 = 99.6\%$. There are 9 random scores and 1916 German scores $> T$, the predictive value for German is $1916/1925 = 99.5\%$.

French. We take the threshold value $T = 11$ for French texts. Then 58 of 2000 French scores are $\le T$, the error of the first kind is $\alpha = 58/2000 = 2.9\%$. For random texts 1967 of 2000 scores are $\le T$, the power is $1967/2000 = 98.3\%$. There are 33 random scores and 1942 French

Table 19: *Frequencies of BLW scores for French vs. random texts*

Score	Random	French
$0 \le x \le 1$	122	0
$1 < x \le 2$	195	0
$2 < x \le 3$	266	0
$3 < x \le 4$	315	0
$4 < x \le 5$	274	0
$5 < x \le 6$	264	0
$6 < x \le 7$	215	2
$7 < x \le 8$	140	0
$8 < x \le 9$	94	10
$9 < x \le 10$	53	15
$10 < x \le 11$	29	31
$11 < x \le 12$	21	50
$12 < x \le 13$	8	114
$13 < x \le 14$	2	239
$14 < x \le 15$	2	322
$15 < x \le 16$	0	415
$16 < x \le 17$	0	420
$17 < x \le 18$	0	258
$18 < x \le 19$	0	115
$19 < x \le 20$	0	8
$20 < x < \infty$	0	1

scores $> T$, the predictive value for French is $1942/1975 = 98.3\%$.

The BLW score is significantly stronger than the MFL score.

7 Coincidences of Two Texts

The first six sections of this chapter introduced efficient methods for recognizing plaintext in comparison with noise. These methods break down for encrypted texts because they ignore properties that remain invariant under encryption. One such invariant property—at least for monoalphabetic substitution—is the equality of two letters, no matter what the concrete value of these letters is.

This is the main idea that we work out in the next sections: Look for identical letters in one or more texts, or in other words, for coincidences.

Definition

Let Σ be a finite alphabet. Let $a = (a_0, \ldots, a_{r-1})$ and $b = (b_0, \ldots, b_{r-1}) \in \Sigma^r$ be two texts of the same length $r \geq 1$. Then

$$\kappa(a, b) := \frac{1}{r} \cdot \#\{j \mid a_j = b_j\} = \frac{1}{r} \cdot \sum_{j=0}^{r-1} \delta_{a_j b_j}$$

is called **coincidence index** of a and b (where $\delta =$ Kronecker symbol).

For each $r \in \mathbb{N}_1$ this defines a map

$$\kappa \colon \Sigma^r \times \Sigma^r \longrightarrow \mathbb{Q} \subseteq \mathbb{R}.$$

The scaling factor $\frac{1}{r}$ makes results for different lengths comparable.

A Perl program is in the Web: `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Perl/kappa.pl`.

Remarks

1. Always $0 \leq \kappa(a, b) \leq 1$.
2. $\kappa(a, b) = 1 \Longleftrightarrow a = b$.
3. By convention $\kappa(\emptyset, \emptyset) = 1$ (where $\emptyset$ denotes the nullstring by abuse of notation).
4. Note that up to scaling the coincidence index is a converse of the Hamming distance that counts non-coincidences.

Example 1: Two English Texts

We compare the first four verses (text 1) of the poem "If ..." by Rudyard Kipling and the next four verses (text 2). (The lengths differ, so we crop the longer one.)

```
IFYOU CANKE EPYOU RHEAD WHENA LLABO UTYOU ARELO OSING THEIR
IFYOU CANMA KEONE HEAPO FALLY OURWI NNING SANDR ISKIT ONONE
||||| |||                                        |
SANDB LAMIN GITON YOUIF YOUCA NTRUS TYOUR SELFW HENAL LMEND
TURNO FPITC HANDT OSSAN DLOOS EANDS TARTA GAINA TYOUR BEGIN
                                  | |
OUBTY OUBUT MAKEA LLOWA NCEFO RTHEI RDOUB TINGT OOIFY OUCAN
NINGS ANDNE VERBR EATHE AWORD ABOUT YOURL OSSIF YOUCA NFORC
                                                 |

WAITA NDNOT BETIR EDBYW AITIN GORBE INGLI EDABO UTDON TDEAL
EYOUR HEART ANDNE RVEAN DSINE WTOSE RVEYO URTUR NLONG AFTER
          |                       |
INLIE SORBE INGHA TEDDO NTGIV EWAYT OHATI NGAND YETDO NTLOO
THEYA REGON EANDS OHOLD ONWHE NTHER EISNO THING INYOU EXCEP
                                             |
KTOOG OODNO RTALK TOOWI SEIFY OUCAN DREAM ANDNO TMAKE DREAM
TTHEW ILLWH ICHSA YSTOT HEMHO LDONI FYOUC ANTAL KWITH CROWD
 |                       |                ||           |
SYOUR MASTE RIFYO UCANT HINKA NDNOT MAKET HOUGH TSYOU RAIMI
SANDK EEPYO URVIR TUEOR WALKW ITHKI NGSNO RLOOS ETHEC OMMON
|                          |
FYOUC ANMEE TWITH TRIUM PHAND DISAS TERAN DTREA TTHOS ETWOI
TOUCH IFNEI THERF OESNO RLOVI NGFRI ENDSC ANHUR TYOUI FALLM
         |  |                                   |
MPOST ORSAS THESA MEIFY OUCAN BEART OHEAR THETR UTHYO UVESP
ENCOU NTWOR THYOU BUTNO NETOO MUCHI FYOUC ANFIL LTHEU NFORG
            ||                                   ||
OKENT WISTE DBYKN AVEST OMAKE ATRAP FORFO OLSOR WATCH THETH
IVING MINUT EWITH SIXTY SECON DSWOR THOFD ISTAN CERUN YOURS
   |   |                               |
INGSY OUGAV EYOUR LIFEF ORBRO KENAN DSTOO PANDB UILDE MUPWI
ISTHE EARTH ANDEV ERYTH INGTH ATSIN ITAND WHICH ISMOR EYOUL
|                                 |
THWOR NOUTT OOLS
LBEAM ANMYS ON
            |
```

In these texts of length 562 we find 35 coincidences, the coincidence index is $\frac{35}{562} = 0.0623$.

Invariance

The coincidence index of two texts is an invariant of polyalphabetic substitution (the keys being equal):

Proposition 1 (Invariance) *Let $f\colon \Sigma^* \longrightarrow \Sigma^*$ be a polyalphabetic encryption function. Then*

$$\kappa(f(a), f(b)) = \kappa(a, b)$$

for all $a, b \in \Sigma^*$ of the same length.

Note that Proposition 1 doesn't need any assumptions on periodicity or on relations between the alphabets used. It only assumes that the encryption function uses the same alphabets at the corresponding positions in the texts.

Mean Values

For a fixed $a \in \Sigma^r$ we determine the mean value of $\kappa(a, b)$ taken over all $b \in \Sigma^r$:

$$\begin{aligned}
\frac{1}{n^r} \cdot \sum_{b\in\Sigma^r} \kappa(a,b) &= \frac{1}{n^r} \cdot \sum_{b\in\Sigma^r} \left[\frac{1}{r} \cdot \sum_{j=0}^{r-1} \delta_{a_j b_j}\right] \\
&= \frac{1}{rn^r} \cdot \sum_{j=0}^{r-1} \underbrace{\left[\sum_{b\in\Sigma^r} \delta_{a_j b_j}\right]}_{n^{r-1}} \\
&= \frac{1}{rn^r} \cdot r \cdot n^{r-1} = \frac{1}{n},
\end{aligned}$$

because, if $b_j = a_j$ is fixed, there remain n^{r-1} possible values for b.

In an analogous way we determine the mean value of $\kappa(a, f_\sigma(b)$ for fixed $a, b \in \Sigma^r$ over all permutations $\sigma \in \mathcal{S}(\Sigma)$:

$$\begin{aligned}
\frac{1}{n!} \cdot \sum_{\sigma\in\mathcal{S}(\Sigma)} \kappa(a, f_\sigma(b)) &= \frac{1}{n!} \cdot \frac{1}{r} \sum_{\sigma\in\mathcal{S}(\Sigma)} \#\{j \mid \sigma b_j = a_j\} \\
&= \frac{1}{rn!} \cdot \#\{(j,\sigma) \mid \sigma b_j = a_j\} \\
&= \frac{1}{rn!} \cdot \sum_{j=0}^{r-1} \#\{\sigma \mid \sigma b_j = a_j\} \\
&= \frac{1}{rn!} \cdot r \cdot (n-1)! = \frac{1}{n},
\end{aligned}$$

because exactly $(n-1)!$ permutations map a_j to b_j.

Note that this conclusion also works for $a = b$.

This derivation shows:

Proposition 2 (i) *The mean value of $\kappa(a,b)$ over all texts $b \in \Sigma^*$ of equal length is $\frac{1}{n}$ for all $a \in \Sigma^*$.*

(ii) *The mean value of $\kappa(a,b)$ over all $a, b \in \Sigma^r$ is $\frac{1}{n}$ for all $r \in \mathbb{N}_1$.*

(iii) *The mean value of $\kappa(a, f_\sigma(b))$ over all monoalphabetic substitutions with $\sigma \in \mathcal{S}(\Sigma)$ is $\frac{1}{n}$ for each pair $a, b \in \Sigma^*$ of texts of equal length.*

(iv) *The mean value of $\kappa(f_\sigma(a), f_\tau(b))$ over all pairs of monoalphabetic substitutions, with $\sigma, \tau \in \mathcal{S}(\Sigma)$, is $\frac{1}{n}$ for each pair $a, b \in \Sigma^*$ of texts of equal length.*

Interpretation

- For a given text a and a "random" text b of the same length $\kappa(a,b) \approx \frac{1}{n}$.
- For "random" texts a and b of the same length $\kappa(a,b) \approx \frac{1}{n}$.
- For given texts a and b of the same length and a "random" monoalphabetic substitution f_σ we have $\kappa(a, f_\sigma(b)) \approx \frac{1}{n}$. This remark justifies treating a nontrivially monoalphabetically encrypted text as random with respect to κ and plaintexts.
- For given texts a and b of the same length and two "random" monoalphabetic substitutions f_σ, f_τ we have $\kappa(f_\sigma(a), f_\tau(b)) \approx \frac{1}{n}$.
- The same holds for "random" polyalphabetic substitutions because counting the coincidences is additive with respect to arbitrary decompositions of texts.

Values that significantly differ from these mean values are suspicious for the cryptanalyst, they could have a *non-random cause*. For more precise statements we should assess the variances (or standard deviations) or, more generally, the distribution of κ-values in certain "populations" of texts.

Variance

First fix $a \in \Sigma^r$ and vary b over all of Σ^r. Using the mean value $\frac{1}{n}$ we calculate the variance:

$$\begin{aligned} V_{\Sigma^r}(\kappa, a) &= \frac{1}{n^r} \cdot \sum_{b \in \Sigma^r} \kappa(a,b)^2 - \frac{1}{n^2} \\ &= \frac{1}{n^r} \cdot \sum_{b \in \Sigma^r} \left[\frac{1}{r} \cdot \sum_{j=0}^{r-1} \delta_{a_j b_j} \right]^2 - \frac{1}{n^2} \end{aligned}$$

Evaluating the square of the sum in brackets we get the quadratic terms

$$\sum_{j=0}^{r-1} \delta_{a_j b_j}^2 = \sum_{j=0}^{r-1} \delta_{a_j b_j} = r \cdot \kappa(a,b) \quad \text{because} \quad \delta_{a_j b_j} = 0 \text{ or } 1$$

$$\sum_{b\in\Sigma^r}\sum_{j=0}^{r-1}\delta_{a_jb_j}^2 = r\cdot\sum_{b\in\Sigma^r}\kappa(a,b) = r\cdot n^r\cdot\frac{1}{n} = r\cdot n^{r-1}$$

and the mixed terms

$$2\cdot\sum_{j=0}^{r-1}\sum_{k=j+1}^{r-1}\delta_{a_jb_j}\delta_{a_kb_k} \quad \text{where} \quad \delta_{a_jb_j}\delta_{a_kb_k} = \begin{cases}1 & \text{if } a_j = b_j \text{ and } a_k = b_k\\ 0 & \text{else}\end{cases}$$

If we fix two letters b_j and b_k, we are left with n^{r-2} different b's that give the value 1. The total sum over the mixed terms evaluates as

$$\sum_{b\in\Sigma^r}\left(2\cdot\sum_{j=0}^{r-1}\sum_{k=j+1}^{r-1}\delta_{a_jb_j}\delta_{a_kb_k}\right) = 2\cdot\sum_{j=0}^{r-1}\sum_{k=j+1}^{r-1}\underbrace{\sum_{b\in\Sigma^r}\delta_{a_jb_j}\delta_{a_kb_k}}_{n^{r-2}}$$

Substituting our intermediary results we get

$$\begin{aligned} V_{\Sigma^r}(\kappa,a) &= \frac{1}{n^r r^2}\left(r\cdot n^{r-1} + r\cdot(r-1)\cdot n^{r-2}\right) - \frac{1}{n^2}\\ &= \frac{1}{rn} + \frac{r-1}{rn^2} - \frac{1}{n^2} = \frac{1}{rn} - \frac{1}{rn^2} = \frac{1}{r}\left(\frac{1}{n} - \frac{1}{n^2}\right)\end{aligned}$$

Next we let a and b vary and calculate the variance of κ:

$$\begin{aligned} V_{\Sigma^r}(\kappa) &= \frac{1}{n^{2r}}\sum_{a,b\in\Sigma^r}\kappa(a,b)^2 - \frac{1}{n^2}\\ &= \frac{1}{n^r}\sum_{a\in\Sigma^r}\underbrace{\left(\frac{1}{n^r}\sum_{b\in\Sigma^r}\kappa(a,b)^2\right)}_{\frac{1}{r}\left(\frac{1}{n}-\frac{1}{n^2}\right)+\frac{1}{n^2}} - \frac{1}{n^2}\\ &= \frac{1}{r}\left(\frac{1}{n} - \frac{1}{n^2}\right) + \frac{1}{n^2} - \frac{1}{n^2} = \frac{1}{r}\left(\frac{1}{n} - \frac{1}{n^2}\right)\end{aligned}$$

We have shown:

Proposition 3 (i) *The mean value of $\kappa(a,b)$ over all texts b of equal length $r\in\mathbb{N}_1$ is $\frac{1}{n}$ with variance $\frac{1}{r}\left(\frac{1}{n}-\frac{1}{n^2}\right)$ for all $a\in\Sigma^r$.*

(ii) *The mean value of $\kappa(a,b)$ over all $a,b\in\Sigma^r$ is $\frac{1}{n}$ with variance $\frac{1}{r}\left(\frac{1}{n}-\frac{1}{n^2}\right)$ for all $r\in\mathbb{N}_1$.*

For the 26 letter alphabet A...Z we have the mean value $\frac{1}{26}\approx 0.0385$, independently from the text length r. The variance is $\approx\frac{0.03370}{r}$, the standard deviation $\approx\frac{0.19231}{\sqrt{r}}$. From this we get the second row of Table 20.

For statistical tests (one-sided in this case) we would like to know the 95% quantiles. If we take the values for a normal distribution as approximations,

Table 20: *Standard deviations and 95% quantiles of κ for random text pairs of length r*

r	10	40	100	400	1000	10000
Std dev	0.0608	0.0304	0.0192	0.0096	0.0061	0.0019
95% quantile	0.1385	0.0885	0.0700	0.0543	0.0485	0.0416

that is "mean value + 1.645 times standard deviation", we get the values in the third row of Table 20. These raw estimates show that the κ-statistic in this form is weak in distinguishing "meaningful" texts from random texts, even for text lengths of 100 letters, and strong only for texts of several thousand letters.

Distinguishing meaningful plaintext from random noise is evidently not the main application of the κ-statistic. The next section will show the true relevancy of the coincidence index.

8 Empirical Values for Natural Languages

Examples

Here are some additional explicit examples.

Example 2: Two German Texts

We compare the two poems “Berg und Burgen schaun herunter” by Heinrich Heine and “Vor Jahren waren wir mal entzweit” by Wilhelm Busch.

```
BERGU NDBUR GENSC HAUNH ERUNT ERIND ENSPI EGELH ELLEN RHEIN
VORJA HRENW ARENW IRMAL ENTZW EITUN DTATE NUNSM ANCHE SZUMT
  |                     |     |
UNDME INSCH IFFCH ENSEG ELTMU NTERR INGSU MGLAE NZTVO NSONN
ORTEW IRSAG TENUN SBEID EZUJE NERZE ITVIE LBITT ERBOE SEWOR
      | |               |     |     |
ENSCH EINRU HIGSE HICHZ UDEMS PIELE GOLDN ERWEL LENKR AUSBE
TEDRA UFHAB ENWIR UNSIN EINAN DERGE SCHIC KTWIR SCHLO SSENF
                                  |         |
WEGTS TILLE RWACH ENDIE GEFUE HLEDI EICHT IEFIM BUSEN HEGTF
RIEDE NUNDH ABEND IEBIT TERBO ESENW ORTEE RSTIC KTUND FESTU
                     |   |      |            |         | |
REUND LICHG RUESS ENDUN DVERH EISSE NDLOC KTHIN ABDES STROM
NDTIE FBEGR ABENJ ETZTI STESW IRKLI CHREC HTFAT ALDAS SWIED
              |   |       |             |  |    | | | |
ESPRA CHTDO CHICH KENNI HNOBE NGLEI SSEND BIRGT SEINI NNRES
EREIN ZWIST NOTWE NDIGO WEHDI EWORT EVOND AZUMA LDIEW ERDEN
|                                      ||         |      |
TODUN DNACH TOBEN LUSTI MBUSE NTUEC KENST ROMDU BISTD ERLIE
NUNWI EDERL EBEND IGDIE KOMME NNUNE RSTIN OFFNE NSTRE ITUND
                            | | |
BSTEN BILDD IEKAN NAUCH SOFRE UNDLI CHNIC KENLA ECHEL TAUCH
FLIEG ENAUF ALLED AECHE RNUNB RINGE NWIRS IEINE WIGKE ITNIC
   |                                       |
SOFRO MMUND MILD
HTWIE DERIN IHREL OECHE R
```

The (common) text length is 414, we find 35 coincidences, the coincidence index is $\frac{35}{414} = 0.0845$.

Example 3: German Text and English Text

We compare the poems by Heine and Kipling (truncated).

```
BERGU NDBUR GENSC HAUNH ERUNT ERIND ENSPI EGELH ELLEN RHEIN
IFYOU CANKE EPYOU RHEAD WHENA LLABO UTYOU ARELO OSING THEIR
    |                                       ||         |||
UNDME INSCH IFFCH ENSEG ELTMU NTERR INGSU MGLAE NZTVO NSONN
SANDB LAMIN GITON YOUIF YOUCA NTRUS TYOUR SELFW HENAL LMEND
                              ||                          |
ENSCH EINRU HIGSE HICHZ UDEMS PIELE GOLDN ERWEL LENKR AUSBE
OUBTY OUBUT MAKEA LLOWA NCEFO RTHEI RDOUB TINGT OOIFY OUCAN
                          |                             |
WEGTS TILLE RWACH ENDIE GEFUE HLEDI EICHT IEFIM BUSEN HEGTF
WAITA NDNOT BETIR EDBYW AITIN GORBE INGLI EDABO UTDON TDEAL
|  |              |                                 |
REUND LICHG RUESS ENDUN DVERH EISSE NDLOC KTHIN ABDES STROM
INLIE SORBE INGHA TEDDO NTGIV EWAYT OHATI NGAND YETDO NTLOO
                    |         |                         | |
ESPRA CHTDO CHICH KENNI HNOBE NGLEI SSEND BIRGT SEINI NNRES
KTOOG OODNO RTALK TOOWI SEIFY OUCAN DREAM ANDNO TMAKE DREAM
          |           |               |
TODUN DNACH TOBEN LUSTI MBUSE NTUEC KENST ROMDU BISTD ERLIE
SYOUR MASTE RIFYO UCANT HINKA NDNOT MAKET HOUGH TSYOU RAIMI
   |                          |         |  |
BSTEN BILDD IEKAN NAUCH SOFRE UNDLI CHNIC KENLA ECHEL TAUCH
FYOUC ANMEE TWITH TRIUM PHAND DISAS TERAN DTREA TTHOS ETWOI
                                              |   |
SOFRO MMUND MILD
MPOST ORSAS THES
```

Text length 414, number of coincidences 28, coincidence index $\frac{28}{414} = 0.0676$.

Example 4: Plaintext and Monoalphabetic Ciphertext

We compare the poem by Heine with a monoalphabetically encrypted version of the poem by Busch:

```
BERGU NDBUR GENSC HAUNH ERUNT ERIND ENSPI EGELH ELLEN RHEIN
UINBG PNZHV GNZHV FNEGD ZHRYV ZFRSH TRGRZ HSHQE GHAPZ QYSER
            |
UNDME INSCH IFFCH ENSEG ELTMU NTERR INGSU MGLAE NZTVO NSONN
INRZV FNQGO RZHSH QLZFT ZYSBZ HZNYZ FRUFZ DLFRR ZNLIZ QZVIN
 |     |        |                                         |
ENSCH EINRU HIGSE HICHZ UDEMS PIELE GOLDN ERWEL LENKR AUSBE
RZTNG SKPGL ZHVFN SHQFH ZFHGH TZNOZ QAPFA CRVFN QAPDI QQZHK
                                           |
```

```
WEGTS TILLE RWACH ENDIE GEFUE HLEDI EICHT IEFIM BUSEN HEGTF
NFZTZ HSHTP GLZHT FZLFR RZNLI ZQZHV INRZZ NQRFA CRSHT KZQRS
   |                                              |

REUND LICHG RUESS ENDUN DVERH EISSE NDLOC KTHIN ABDES STROM
HTRFZ KLZON GLZHB ZRYRF QRZQV FNCDF APNZA PRKGR GDTGQ QVFZT

ESPRA CHTDO CHICH KENNI HNOBE NGLEI SSEND BIRGT SEINI NNRES
ZNZFH YVFQR HIRVZ HTFOI VZPTF ZVINR ZUIHT GYSEG DTFZV ZNTZH
                      |                                |

TODUN DNACH TOBEN LUSTI MBUSE NTUEC KENST ROMDU BISTD ERLIE
HSHVF ZTZND ZLZHT FOTFZ CIEEZ HHSHZ NQRFH IKKHZ HQRNZ FRSHT
                                                       |

BSTEN BILDD IEKAN NAUCH SOFRE UNDLI CHNIC KENLA ECHEL TAUCH
KDFZO ZHGSK GDDZT GZAPZ NHSHL NFHOZ HVFNQ FZFHZ VFOCZ FRHFA

SOFRO MMUND MILD
PRVFZ TZNFH FPNZ
```

Text length 414, number of coincidences 11, coincidence index $\frac{11}{414} = 0.0266$.

Example 5: Two Independent Polyalphabetic Ciphertexts

We encrypt the poems by Heine and by Busch with different polyalphabetic substitutions and compare the resulting ciphertexts.

```
YSPHK CBZNS TSKIU XTYUG XCSBJ YSJUB XTQDX YFDRG XXIWB IGDOO
MMIWH HZRCW UNMPD WHJUY MPNLO NZCNP MDSSY ANPEA SLWUM VHFBS
                     |

PTTZW NOVGG ZUZUE YOVJF XXRZK CVDHS ZTBIK BFMDC GMRLC CUQUO
FNEDD WHRUS EDYFC RVQRC OLLGY AMUHR TSOVM MJWJS YNIQO CXWFN
                                 |                    |

XTQUE YIPHW AYBIW XIBMN PRAZI FIDRC TAIVB YSEJL DSKTG SWVFC
EDMBS UKUHN OTOFI DXVST XFDLX COBKN JOQIL YJWZN ZBRKD RJQXF
         |                                |

RSBJI KIMRC LJEUE YOCOC TSZKW XLDII XYNEJ NCFOM YGQWB XCGXD
ZWXEY ANPMV STYAL IOOTS LQTNK RINDG YUNRX QJCRB VDLLX RMVNF
                   |

LSSBV ZIBMF LGAII YOCNO EIAGE YIVEC GRICU AVIOO WPTWI JVUVM
CELVM FJRKQ UMMPU RJKLV ZWOCO FIXVI LVHNW UEFID SIXLZ VDWXE
                               |

XDMGR VGWIP HWDUE ACPUI ATLSW CFMJI MDABV UIULV MSDBX COUJU
YNMIY LOFJC XQNHX LXVPQ DRAEZ QCQZD XVFAL EHFBA BPRDD RHEYA
  |
```

```
OATKB WOAGG OAXWB ZWVXI FPSIW CVYJZ CSKIJ IPOIW YYQJV YSMOC
XXYHT NXQTM OOXLX VPCSR EMCKM PYFCN IBEIY ZYBDQ XVNBX FLDXC
            |                          |                  |
YDRWB UIMIB ZSGRB CTYGG MAZGW LOCRI HWKXU ACPRT XQCWA KTYGG
PKTNA QXEBS SIBQL EOPAN IANPJ BTLAQ XKSBI FYVXD DWKHY VEPSP
                         |
MAZGC BMYUB FYIV
ASPVM COBTL ZUTD
```

Text length 414, number of coincidences 12, coincidence index $\frac{12}{414} = 0.0290$.

Example 6: Two Polyalphabetic Ciphertexts With the Same Key

Now we encrypt the poems by Heine and by Busch with the same polyalphabetic substitution and compare the resulting ciphertexts

```
UNISN PMOLQ AQXVL VSUDU MUBTJ NIVXC OTIOZ QPDWV XIBQX URRTL
MMIWH HZRCW UNMPD WHJUY MPNLO NZCNP MDSSY ANPEA SLWUM VHFBS
  |                     |     |
MALOO WCRWU RFPPA NDBMG OKJJM AEDZB TLABN OQKSN DJEYK TIMDA
FNEDD WHRUS EDYFC RVQRC OLLGY AMUHR TSOVM MJWJS YNIQO CXWFN
      | |               |     |     |
MPEPA NZATX RWKRY URBRL LEYKZ RSRNN ATVCY RHWYY VDYYH AMBID
EDMBS UKUHN OTOFI DXVST XFDLX COBKN JOQIL YJWZN ZBRKD RJQXF
                                  |         |
DRKSJ CRMWR HWUOQ DYQTN AQOXO VNNXV MILVJ FYRRO JFIND UMGNS
ZWXEY ANPMV STYAL IOOTS LQTNK RINDG YUNRX QJCRB VDLLX RMVNF
                     |   |      |            |         | |
HNMAL MSPAC IDMVE RCEMA LYOBA NZBZD YQNMW XEHST STXQZ VNBDJ
CELVM FJRKQ UMMPU RJKLV ZWOCO FIXVI LVHNW UEFID SIXLZ VDWXE
              |   |       |             |  |    | | | |
YBKUI PASXT JHSPA HYAXI RTDTY APMOW IRYAL NSBKS JQRPS TCQYB
YNMIY LOFJC XQNHX LXVPQ DRAEZ QCQZD XVFAL EHFBA BPRDD RHEYA
|                                      ||         |      |
EQMFC EDLJH NZUND YNVNW BTMBM PNFXZ NQXVN BDJXD IIEDW NIYRD
XXYHT NXQTM OOXLX VPCSR EMCKM PYFCN IBEIY ZYBDQ XVNBX FLDXC
                            | | |
JCJND MRMMQ TNNLX PIFVD JTOUO FCEBV JHYWV HYAVE OPANB CHXLV
PKTNA QXEBS SIBQL EOPAN IANPJ BTLAQ XKSBI FYVXD DWKHY VEPSP
   |                                       |
IMKNY OXFCE CVVC
ASPVM COBTL ZUTD
```

Text length 414, number of coincidences 35, coincidence index $\frac{35}{414} = 0.0845$—an expected result because identical plaintext letters are transformed to identical ciphertext letters.

Empirical Observations

These examples show some tendencies that will be empirically or mathematically founded later in this section:

- The typical coincidence index of two German texts is about 0.08.
- The typical coincidence index of two English texts is about 0.06.
- The typical coincidence index of a German and an English text is about 0.06 to 0.07.
- The typical coincidence index of a plaintext and ciphertext is about 0.03 to 0.05, that is near the "random" value $\frac{1}{26} \approx 0.0385$. The same is true for two independent ciphertexts.
- *If the same key is used for two polyalphabetic ciphertexts this fact reveals itself by a coincidence index that resembles that of two plaintexts.*

This latter statement is the first application of coincidence counts. No matter whether the encryption is periodic or not—if we get several ciphertexts encrypted in the same way, we can arrange them in parallel rows and get monoalphabetically encrypted columns that eventually can be decrypted.

Historical Example

The Polish cryptanalyst REJEWSKI was the first who successfully broke early military versions of the German cipher machine Enigma, see Chapter 6. He detected that ciphertexts were "in phase" by coincidence counts. It is unknown whether he knew FRIEDMAN's approach, or whether he found it for himself. FRIEDMAN's early publications were not classified and published even in France.

For example REJEWSKI noted that the two ciphertexts

```
RFOWL DOCAI HWBGX EMPTO BTVGG INFGR OJVDD ZLUWS JURNK KTEHM
RFOWL DNWEL SCAPX OAZYB BYZRG GCJDX NGDFE MJUPI MJVPI TKELY
```

besides having the initial six letters identical also had a suspicious number of coincidences between the remaining 44 letters ($5/44 \approx 0.114$).

Exercise. How many coincidences among 44 letters would you expect for independently encrypted texts?

REJEWSKI assumed that the first six letters denoted a "message key" that was identical for the two messages, and from this, that the Enigma operators prefixed their messages by a six letter message key. (Later on he even detected that in fact they used a repeated three letter key.)

Source: F. L. Bauer: Mathematik besiegte in Polen die unvernünftig gebrauchte ENIGMA. Informatik Spektrum 1. Dezember 2005, 493–497.]

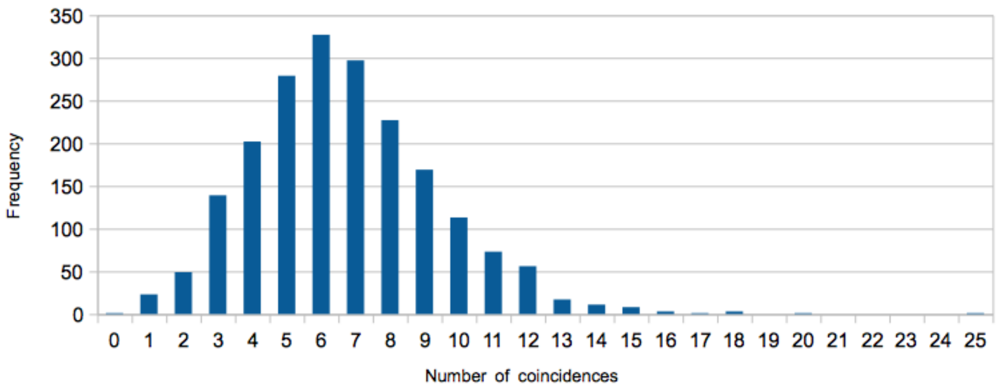

Figure 7: *Frequency of coincidence counts for 2000 English text pairs of 100 letters—to get coincidence indices divide x-values by* 100

The Kappa Distribution for English Texts

We want to learn more about the distribution of coincidence indices $\kappa(a, b)$ for English texts (or text chunks) a and b. To this end we take a large English text—in this case the book *The Poisoned Pen* by Arthur B. Reeve (that by the way contains a cryptogram) from Project Gutenberg—and chop it into chunks $a, b, c, d, \ldots$ of r letters each. Then we count $\kappa(a, b)$, $\kappa(c, d)$, $\ldots$ and list the values in the first column of a spreadsheet for easy evaluation. See the Perl program `kapstat.pl` in `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Perl/` and the spreadsheet `EnglKap.xls` in `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Files/`

> In fact we also record the pure incidence counts as integers. This makes it easier drawing a histogram without generating discretization artefacts.

The text has 449163 letters. Taking $r = 100$ we get 2245 text pairs. We take the first 2000 of them. Table 21 and Figure 7 show some characteristics of the distribution.

The Kappa Distribution for German Texts

We repeat this procedure for German texts, using *Scepter und Hammer* by Karl May from the web page of the Karl-May-Gesellschaft. We take the first 2000 text pairs. The results are in Table 22 and Figure 8.

Table 21: *Distribution of κ for 2000 English text pairs of 100 letters*

Minimum:	0.00		
Median:	0.06	Mean value:	0.0669
Maximum:	0.25	Standard dev:	0.0272
1st quartile:	0.05	5% quantile:	0.0300
3rd quartile:	0.08	95% quantile:	0.1200

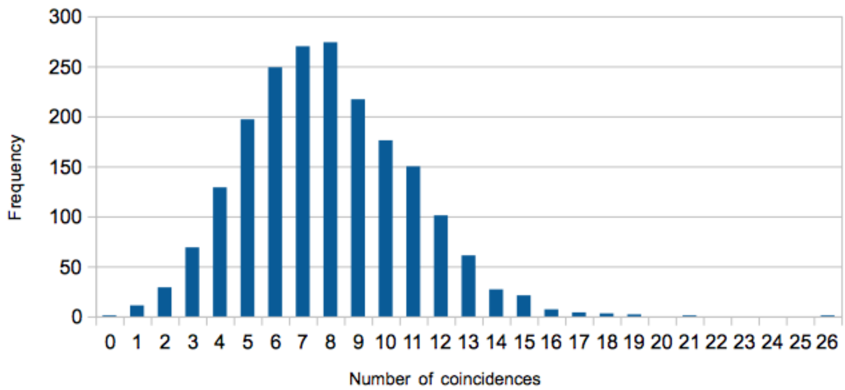

Figure 8: *Frequency of coincidence counts for 2000 German text pairs of 100 letters—to get coincidence indices divide x-values by* 100

Table 22: *Distribution of κ for 2000 German text pairs of 100 letters*

Minimum:	0.00		
Median:	0.08	Mean value:	0.0787
Maximum:	0.26	Standard dev:	0.0297
1st quartile:	0.06	5% quantile:	0.0300
3rd quartile:	0.10	95% quantile:	0.1300

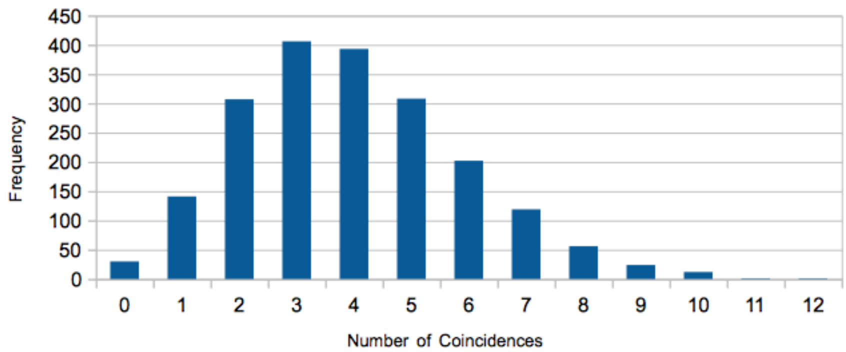

Figure 9: *Frequency of coincidence counts for 2000 random text pairs of 100 letters—to get coincidence indices divide x-values by* 100

Table 23: *Distribution of κ for 2000 random text pairs of 100 letters*

Minimum:	0.00		
Median:	0.04	Mean value:	0.040
Maximum:	0.12	Standard dev:	0.020
1st quartile:	0.03	5% quantile:	0.010
3rd quartile:	0.05	95% quantile:	0.070

The Kappa Distribution for Random Texts

Finally the same procedure for random texts. To this end we generate a 400000 character text by the built-in (pseudo-) random generator of Perl. Since the simulation might depend on the quality of the random generator we enhance the random text in the following way: We generate 8132 random letters by the cryptographically strong BBS-generator and use them as key for a Belaso encryption of our random text, repeating the key several times. In spite of this periodicity we may assume that the result gives a 400000 character random text of good quality. This provides us with 2000 text pairs of length 100. The results are in Table 23 and Figure 9. Note that the values fit the theoretical values almost perfectly.

Applications

To test whether a text a belongs to a certain language we would take one (or maybe several) fixed texts of the language and would test a against them. Because the values for natural languages are quite similar this test would only make sense for testing against random. This test is much weaker then the MFL, LW and BLW tests.

Also adjusting the columns of a disk cipher could be tested this way: If two alphabets are relatively shifted, the corresponding columns behave like random texts with respect to each other. If the alphabets are properly adjusted, the columns represent meaningful texts encrypted by the same monoalphabetic substitution, therefore they belong to the same language and show the typical coincidence index—up to statistical noise. Note that we need quite long columns for this test to work in a sensible way!

In the following sections we'll see some better tests for these problems. The main application of the coincidence index in its pure form is detecting identically encrypted polyalphabetic ciphertexts. Moreover it is the basis of some refined methods.

9 Autoincidence of a Text

Introduction

For the cryptanalysis of periodic polyalphabetic ciphers the following construction is of special importance: Let $a \in \Sigma^*$, and let $a_{(q)}$ and $a_{(-q)}$ be the cyclic shifts of a by q positions to the right resp. to the left. That is

$$\begin{array}{lllllllllll} a & = & a_0 & a_1 & a_2 & \ldots & a_{q-1} & a_q & a_{q+1} & \ldots & a_{r-1} \\ a_{(q)} & = & a_{r-q} & a_{r-q+1} & a_{r-q+2} & \ldots & a_{r-1} & a_0 & a_1 & \ldots & a_{r-q-1} \\ a_{(-q)} & = & a_q & a_{q+1} & a_{q+2} & \ldots & a_{2q-1} & a_{2q} & a_{2q+1} & \ldots & a_{q-1} \end{array}$$

Clearly $\kappa(a, a_{(q)}) = \kappa(a, a_{(-q)})$.

Definition. For a text $a \in \Sigma^*$ and a natural number $q \in \mathbb{N}$ the number $\kappa_q(a) := \kappa(a, a_{(q)})$ is called the q-th **autocoincidence index** of a.

Note. This is not a common notation. Usually this concept is not given an explicit name.

Example. We shift a text by 6 positions to the right:

```
COINCIDENCESBETWEENTHETEXTANDTHESHIFTEDTEXT <-- original text
EDTEXTCOINCIDENCESBETWEENTHETEXTANDTHESHIFT <-- shifted by 6
             |  |       | |           |    | <-- 6 coincidences
```

Properties

The q-th autocoincidence index κ_q defines a map

$$\kappa_q \colon \Sigma^* \longrightarrow \mathbb{Q}.$$

Clearly $\kappa_q(a) = \kappa_{r-q}(a)$ for $a \in \Sigma^r$ and $0 < q < r$, and κ_0 is a constant map.

Application

Take a ciphertext c that is generated by a periodic polyalphabetic substitution. If we determine $\kappa_q(c)$, we encounter two different situations: In the general case q is not a multiple of the period l. Counting the coincidences we encounter letter pairs that come from independent monoalphabetic substitutions. By the results of Section 7 we expect an index $\kappa_q(c) \approx \frac{1}{n}$.

In the special case where $l|q$ however we encounter the situation

$$\begin{array}{llllll} \sigma_0 a_0 & \sigma_1 a_1 & \ldots & \sigma_0 a_q & \sigma_1 a_{q+1} & \ldots \\ & & & \sigma_0 a_0 & \sigma_1 a_1 & \ldots \end{array}$$

where the letters below each other come from the same monoalphabetic substitution. Therefore they coincide if and only if the corresponding plaintext letters coincide. Therefore we expect an index $\kappa_q(c)$ near the coincidence index κ_M that is typical for the plaintext language M.

More precisely for a polyalphabetic substitution f of period l, plaintext a, and ciphertext $c = f(a)$:

1. For l not a divisor of q or $r - q$ we expect $\kappa_q(c) \approx \frac{1}{n}$.
2. For $l|q$ and q small compared with r we expect $\kappa_q(c) \approx \kappa_q(a)$, and this value should be near the typical coincidence index κ_M.

This is the second application of coincidence counts, detecting the period of a polyalphabetic substitution by looking at the autocoincidence indices of the ciphertext. Compared with the search for repetitions after Kasiski this method also takes account of repetitions of length 1 or 2. In this way we make much more economical use of the traces that the period leaves in the ciphertext.

Example

We want to apply these considerations to the autocoincidence analysis of a polyalphabetic ciphertext using the Perl program `coinc.pl` from `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Perl/`. We start with the cryptogram that we already have solved in Chapter 2 by repetition analysis:

```
        00    05    10    15    20    25    30    35    40    45

0000    AOWBK NLRMG EAMYC ZSFJO IYYVS HYQPY KSONE MDUKE MVEMP JBBOA
0050    YUHCB HZPYW MOOKQ VZEAH RMVVP JOWHR JRMWK MHCMM OHFSE GOWZK
0100    IKCRV LAQDX MWRMH XGTHX MXNBY RTAHJ UALRA PCOBJ TCYJA BBMDU
0150    HCQNY NGKLA WYNRJ BRVRZ IDXTV LPUEL AIMIK MKAQT MVBCB WVYUX
0200    KQXYZ NFPGL CHOSO NTMCM JPMLR JIKPO RBSIA OZZZC YPOBJ ZNNJP
0250    UBKCO WAHOO JUWOB CLQAW CYTKM HFPGL KMGKH AHTYG VKBSK LRVOQ
0300    VOEQW EALTM HKOBN CMVKO BJUPA XFAVK NKJAB VKNXX IJVOP YWMWQ
0350    MZRFB UEVYU ZOORB SIAOV VLNUK EMVYY VMSNT UHIWZ WSYPG KAAIY
0400    NQKLZ ZZMGK OYXAO KJBZV LAQZQ AIRMV UKVJO CUKCW YEALJ ZCVKJ
0450    GJOVV WMVCO ZZZPY WMWQM ZUKRE IWIPX BAHZV NHJSJ ZNSXP YHRMG
0500    KUOMY PUELA IZAMC AEWOD QCHEW OAQZQ OETHG ZHAWU NRIAA QYKWX
0550    EJVUF UZSBL RNYDX QZMNY AONYT AUDXA WYHUH OBOYN QJFVH SVGZH
0600    RVOFQ JISVZ JGJME VEHGD XSVKF UKXMV LXQEO NWYNK VOMWV YUZON
0650    JUPAX FANYN VJPOR BSIAO XIYYA JETJT FQKUZ ZZMGK UOMYK IZGAW
0700    KNRJP AIOFU KFAHV MVXKD BMDUK XOMYN KVOXH YPYWM WQMZU EOYVZ
0750    FUJAB YMGDV BGVZJ WNCWY VMHZO MOYVU WKYLR MDJPV JOCUK QELKM
```

```
0800  AJBOS YXQMC AQTYA SABBY ZICOB XMZUK POOUM HEAUE WQUDX TVZCG
0850  JJMVP MHJAB VZSUM CAQTY AJPRV ZINUO NYLMQ KLVHS VUKCW YPAQJ
0900  ABVLM GKUOM YKIZG AVLZU VIJVZ OGJMO WVAKH CUEYN MXPBQ YZVJP
0950  QHYVG JBORB SIAOZ HYZUV PASMF UKFOW QKIZG ASMMK ZAUEW YNJAB
1000  VWEYK GNVRM VUAAQ XQHXK GVZHU VIJOY ZPJBB OOQPE OBLKM DVONV
1050  KNUJA BBMDU HCQNY PQJBA HZMIB HWVTH UGCTV ZDIKG OWAMV GKBBK
1100  KMEAB HQISG ODHZY UWOBR ZJAJE TJTFU K
```

The Autocoincidence Indices

This is the sequence of autocoincidence indices of our cryptogram

κ_1	κ_2	κ_3	κ_4	κ_5	κ_6	κ_7	κ_8
0.0301	0.0345	0.0469	0.0354	0.0371	0.0354	**0.0822**	0.0416
κ_9	κ_{10}	κ_{11}	κ_{12}	κ_{13}	κ_{14}	κ_{15}	κ_{16}
0.0265	0.0309	0.0416	0.0389	0.0327	**0.0787**	0.0460	0.0345
κ_{17}	κ_{18}	κ_{19}	κ_{20}	κ_{21}	κ_{22}	κ_{23}	κ_{24}
0.0460	0.0309	0.0327	0.0309	**0.0769**	0.0318	0.0309	0.0327
κ_{25}	κ_{26}	κ_{27}	κ_{28}	κ_{29}	κ_{30}	κ_{31}	κ_{32}
0.0318	0.0309	0.0416	**0.0875**	0.0477	0.0416	0.0442	0.0354
κ_{33}	κ_{34}	κ_{35}	κ_{36}				
0.0318	0.0389	**0.0610**	0.0371				

The period 7 stands out, as it did with the period analysis after KASISKI in the last chapter. This is also clearly seen in the graphical representation, see Figure 10.

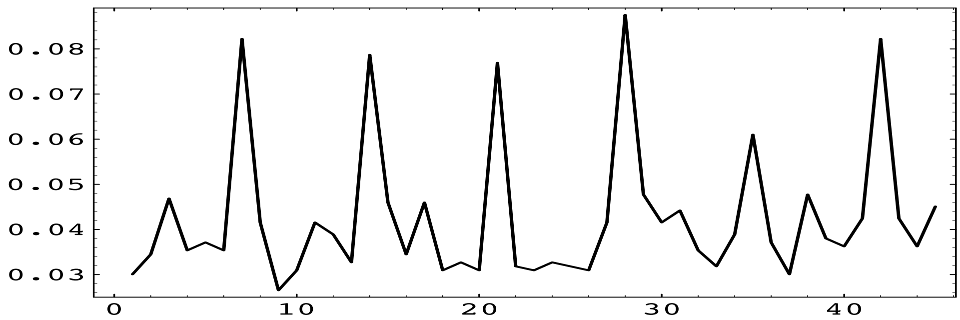

Figure 10: *Autocoincidence spectrum of a sample ciphertext*

The values other than at multiples of 7 fluctuate around the "random" value $\frac{1}{26} \approx 0.0385$ as expected. The values in the peaks fluctuate around the typical coincidence index near 0.08 of the plaintext language German, for which we gave empirical evidence in the last section. This effect has an easy explanation.

The Autocoincidence Spectrum

To analyze the effect seen in Figure 10, let c be the ciphertext from a polyalphabetic encryption of a text $a \in M$ with period l. What values can we expect for the $\kappa_q(c)$?

$c =$	c_0	$\ldots$	c_{q-1}	c_q		$\ldots$	c_{r-1}
$c_{(q)} =$	c_{r-q}	$\ldots$	c_{r-1}	c_0		$\ldots$	c_{r-q-1}
expected coinc.:	$q \cdot \kappa_M$	if	$l \mid r-q$,	$(r-q) \cdot \kappa_M$	if	$l \mid q$,	
	$q \cdot \kappa_{\Sigma^*}$	else		$(r-q) \cdot \kappa_{\Sigma^*}$	else		

Adding these up we get the following expected values for the autocoincidence spectrum:

1. case, $l|r$

$$\kappa_q(c) \approx \begin{cases} \frac{q \cdot \kappa_M + (r-q) \cdot \kappa_M}{r} = \kappa_M & \text{if } l|q, \\ \frac{q \cdot \kappa_{\Sigma^*} + (r-q) \cdot \kappa_{\Sigma^*}}{r} = \kappa_{\Sigma^*} & \text{else.} \end{cases}$$

2. case, $l \not| r$

$$\kappa_q(c) \approx \begin{cases} \frac{q \cdot \kappa_{\Sigma^*} + (r-q) \cdot \kappa_M}{r} & \text{if } l|q, \\ \frac{q \cdot \kappa_M + (r-q) \cdot \kappa_{\Sigma^*}}{r} & \text{if } l|r-q, \\ \kappa_{\Sigma^*} & \text{else.} \end{cases}$$

In particular for $q << r$

$$\kappa_q(c) \approx \begin{cases} \kappa_M & \text{if } l|q, \\ \kappa_{\Sigma^*} & \text{else.} \end{cases}$$

This explains the autocoincidence spectrum that we observed in the example. Typical autocoincidence spectra are shown in Figures 11 and 12.

Since in the second case the resulting image may be somewhat blurred, one could try to calculate autocoincidence indices not by shifting the text cyclically around but by simply cutting off the ends.

Definition. The sequence $(\kappa_1(a), \ldots, \kappa_{r-1}(a))$ of autocoincidence indices of a text $a \in \Sigma^r$ of length r is called the **autocoincidence spectrum** of a.

Note. that this notation too is not common in the literature, but seems adequate for its evident cryptanalytical importance.

Exercise 1. Determine the autocoincidence spectrum of the ciphertext that you already broke by a KASISKI analysis. Create a graphical representation of it using graphic software of your choice.

Exercise 2. Cryptanalyze the ciphertext

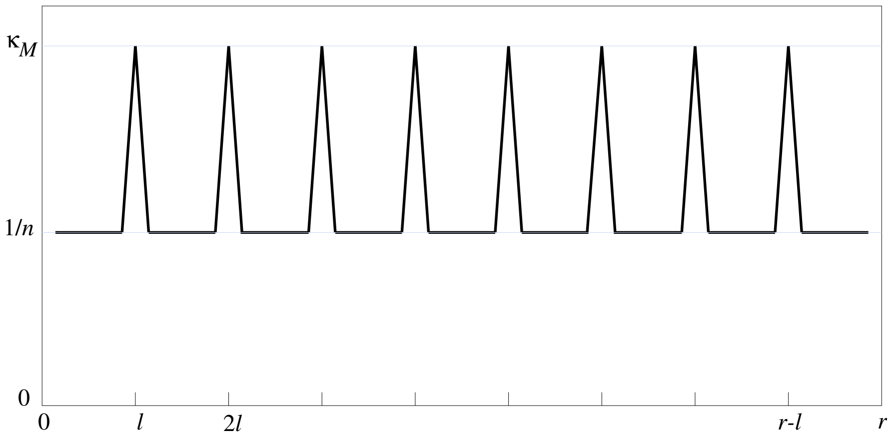

Figure 11: *Text length is multiple of period*

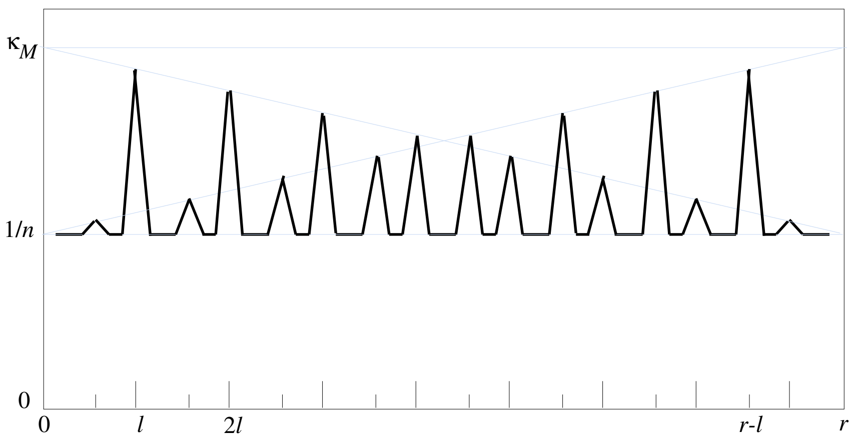

Figure 12: *Text length not multiple of period*

```
ECWUL MVKVR SCLKR IULXP FFXWL SMAEO HYKGA ANVGU GUDNP DBLCK
MYEKJ IMGJH CCUJL SMLGU TXWPN FQAPU EUKUP DBKQO VYTUJ IVWUJ
IYAFL OVAPG VGRYL JNWPK FHCGU TCUJK JYDGB UXWTT BHFKZ UFSWA
FLJGK MCUJR FCLCB DBKEO OUHRP DBVTP UNWPZ ECWUL OVAUZ FHNQY
XYYFL OUFFL SHCTP UCCWL TMWPB OXNKL SNWPZ IIXHP DBSWZ TYJFL
NUMHD JXWTZ QLMEO EYJOP SAWPL IGKQR PGEVL TXWPU AODGA ANZGY
BOKFH TMAEO FCFIH OTXCT PMWUO BOK
```

10 The Inner Coincidence Index of a Text

Definition

Let $a \in \Sigma^r$ ($r \geq 2$) be a text, and $(\kappa_1(a), \ldots, \kappa_{r-1}(a))$ be its autocoincidence spectrum. Then the mean value

$$\varphi(a) := \frac{1}{r-1}\left[\kappa_1(a) + \cdots + \kappa_{r-1}(a)\right]$$

is called the **(inner) coincidence index** of a.

It defines a map

$$\varphi\colon \Sigma^{(\geq 2)} \longrightarrow \mathbb{Q}.$$

See the Perl program `phi.pl` from `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Perl/`.

Another description

Pick up the letters from two random positions of a text a. How many "twins" will you find? That means the same letter $s \in \Sigma$ at the two positions, or a "coincidence"?

Let $m_s = m_s(a) = \#\{j \mid a_j = s\}$ be the number of occurrences of s in a. Then the answer is

$$\frac{m_s \cdot (m_s - 1)}{2}$$

times. Therefore the total number of coincidences is

$$\sum_{s\in\Sigma} \frac{m_s \cdot (m_s-1)}{2} = \frac{1}{2} \cdot \sum_{s\in\Sigma} m_s^2 - \frac{1}{2} \cdot \sum_{s\in\Sigma} m_s = \frac{1}{2} \cdot \sum_{s\in\Sigma} m_s^2 - \frac{r}{2}$$

We count these coincidences in another way by the following algorithm: Let z_q be the number of already found coincidences with a distance of q for $q = 1, \ldots, r-1$, and initialize it as $z_q := 0$. Then execute the nested loops

for $i = 0, \ldots, r-2$	[loop through the text a]
for $j = i+1, \ldots, r-1$	[loop through the remaining text]
if $a_i = a_j$	[coincidence detected]
increment z_{j-i}	[with distance $j - i$]
increment z_{r+i-j}	[and with distance $r + i - j$]

After running through these loops the variables $z_1, \ldots, z_{r-1}$ have values such that

Lemma 1 (i) $z_1 + \cdots + z_{r-1} = \sum_{s\in\Sigma} m_s \cdot (m_s - 1)$.

(ii) $\kappa_q(a) = \frac{z_q}{r}$ *for* $q = 1, \ldots, r-1$.

Proof. (i) We count all coincidences twice.

(ii) $\kappa_q(a) = \frac{1}{r} \cdot \#\{j \,|\, a_{j+q} = a_j\}$ by definition (where the indices are taken mod r). $\diamond$

The Kappa-Phi Theorem

Theorem 1 (Kappa-Phi Theorem) *The inner coincidence index of a text $a \in \Sigma^*$ of length $r \geq 2$ is the proportion of coincidences among all letter pairs of a.*

Proof. The last term of the equation

$$\begin{aligned}
\varphi(a) &= \frac{\kappa_1(a) + \cdots \kappa_{r-1}(a)}{r-1} = \frac{z_1 + \cdots + z_{r-1}}{r \cdot (r-1)} \\
&= \frac{\sum_{s\in\Sigma} m_s \cdot (m_s - 1)}{r \cdot (r-1)} = \frac{\sum_{s\in\Sigma} \frac{m_s \cdot (m_s-1)}{2}}{\frac{r\cdot(r-1)}{2}}
\end{aligned}$$

has the total number of coincidences in its numerator, and the total number of letter pairs in its denominator. ◇

Corollary 1 *The inner coincidence index may be expressed as*

$$\varphi(a) = \frac{r}{r-1} \cdot \sum_{s\in\Sigma} \left(\frac{m_s}{r}\right)^2 - \frac{1}{r-1}$$

Proof. This follows via the intermediate step

$$\varphi(a) = \frac{\sum_{s\in\Sigma} m_s^2 - r}{r \cdot (r-1)}$$

◇

Note that this corollary provides a much faster algorithm for determining $\varphi(a)$. The definition formula needs $r - 1$ runs through a text of length r, making $r \cdot (r-1)$ comparisons. The above algorithm reduces the costs to $\frac{r\cdot(r-1)}{2}$ comparisons. Using the formula of the corollary we need only one pass through the text, the complexity is linear in r. For a Perl program implementing this algorithm see the Perl script `coinc.pl` from the web page `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Perl/`

Corollary 2 *The inner coincidence index of a text is invariant under* **mono***alphabetic substitution.*

Proof. The number of letter pairs is unchanged. ◇

11 The Distribution of the Inner Coincidence Index

First we calculate the exact mean value of the inner coincidence index $\varphi(a)$ for $a \in \Sigma^r$. Then we determine empirical values for mean value and variance for English, German, and random texts by simulation, as we did for κ.

The exact value of the variance leads to a somewhat more complicated calculation. We omit it.

Mean Value

We calculate the mean value of the letter frequencies $m_s(a)$ over $a \in \Sigma^r$ for each $s \in \Sigma$. Because of the symmetry in s all these values are identical, therefore we have

$$n \cdot \sum_{a\in\Sigma^r} m_s(a) = \sum_{s\in\Sigma}\sum_{a\in\Sigma^r} m_s(a) = \sum_{a\in\Sigma^r}\underbrace{\sum_{s\in\Sigma} m_s(a)}_{r} = r\cdot n^r$$

This gives the mean value

$$\frac{1}{n^r}\sum_{a\in\Sigma^r} m_s(a) = \frac{r}{n}$$

for each letter $s \in \Sigma$.

Next we calculate the mean value of $\kappa_q(a)$ over $a \in \Sigma^r$. We treat the indices of the letters of the texts a as elements of the cyclic additive group $\mathbb{Z}/n\mathbb{Z}$. Then we have

$$\begin{aligned}
\sum_{a\in\Sigma^r} \kappa_q(a) &= \sum_{a\in\Sigma^r} \frac{1}{r}\#\{j \in \mathbb{Z}/n\mathbb{Z} \mid a_{j+q} = a_j\} \\
&= \frac{1}{r}\sum_{j\in\mathbb{Z}/n\mathbb{Z}}\sum_{a\in\Sigma^r} \delta_{a_{j+q},a_j} \\
&= \frac{1}{r}\sum_{j\in\mathbb{Z}/n\mathbb{Z}} \underbrace{\#\{a \in \Sigma^r \mid a_{j+q} = a_j\}}_{n^{r-1}} \\
&= n^{r-1}
\end{aligned}$$

because in the underbraced count for a we may choose $r-1$ letters freely, and then the remaining letter is fixed. This gives the mean value

$$\frac{1}{n^r}\sum_{a\in\Sigma^r} \kappa_q(a) = \frac{1}{n}$$

for each $q = 1, \ldots, r-1$.

Now for φ. We use the additivity of the mean value.

$$\begin{aligned}\frac{1}{n^r}\sum_{a\in\Sigma^r}\varphi(a) &= \frac{1}{r-1}\left[\frac{1}{n^r}\sum_{a\in\Sigma^r}\kappa_1(a)+\cdots+\frac{1}{n^r}\sum_{a\in\Sigma^r}\kappa_{r-1}(a)\right]\\ &= \frac{1}{r-1}\cdot(r-1)\cdot\frac{1}{n}=\frac{1}{n}\end{aligned}$$

We have shown:

Proposition 4 *The mean values of the q-th autocoincidence index for $q = 1,\ldots,r-1$ and of the inner coincidence index over $a\in\Sigma^r$ each are $\frac{1}{n}$.*

And for the letter frequencies we have:

Corollary 3 *The sum of the letter frequencies $m_s(a)$ over $a\in\Sigma^r$ is*

$$\sum_{a\in\Sigma^r} m_s(a) = r\cdot n^{r-1}$$

for all letters $s\in\Sigma$.

Corollary 4 *The sum of the squares $m_s(a)^2$ of the letter frequencies over $a\in\Sigma^r$ is*

$$\sum_{a\in\Sigma^r} m_s(a)^2 = r\cdot(n+r-1)\cdot n^{r-2}$$

for all letters $s\in\Sigma$.

Proof. By the Kappa-Phi Theorem we have

$$\sum_{t\in\Sigma}\left[\sum_{a\in\Sigma^r} m_s(a)^2-\sum_{a\in\Sigma^r} m_s(a)\right] = r\cdot(r-1)\cdot\sum_{a\in\Sigma^r}\varphi(a) = r\cdot(r-1)\cdot n^{r-1}$$

Substituting the result of the previous corollary and using the symmetry of the sum of squares with respect to s we get

$$n\cdot\sum_{a\in\Sigma^r} m_s(a)^2 = \sum_{t\in\Sigma}\sum_{a\in\Sigma^r} m_s(a)^2 = r\cdot(r-1)\cdot n^{r-1}+rn\cdot n^{r-1} = r\cdot n^{r-1}\cdot(r-1+n)$$

Dividing by n we get the above formula. $\diamond$

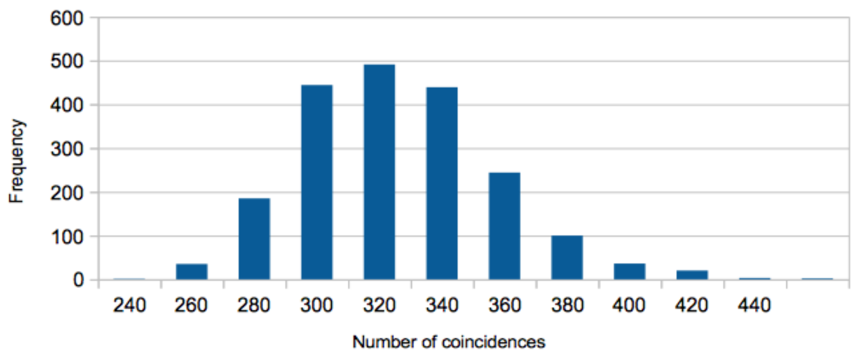

Figure 13: *Frequency of inner coincidence counts for 2000 English texts of 100 letters—to get φ values divide x-values by* 4950

Table 24: *Distribution of φ for 2000 English texts of 100 letters*

Minimum:	0.0481		
Median:	0.0634	Mean value:	0.0639
Maximum:	0.0913	Standard dev:	0.0063
1st quartile:	0.0594	5% quantile:	0.0549
3rd quartile:	0.0677	95% quantile:	0.0750

The Phi Distribution for English Texts

For empirically determining the distribution of the inner coincidence index $\varphi(a)$ we use the Perl program `phistat.pl` from `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Perl/`. For English texts (or text chunks) a, we again take a large English text—in this case the book *The Fighting Chance* by Robert W. Chambers from Project Gutenberg—and chop it into chunks $a, b, c, d, \ldots$ of r letters each. Then we count $\varphi(a)$, $\varphi(b)$, ... and list the values in the first column of a spreadsheet. See the file `EnglPhi.xls` in `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Files/`. The text has 602536 letters. We take the first 262006 of them and consider the first 2000 pieces of 100 letters each. Table 24 and Figure 13 show some characteristics of the distribution.

The Phi Distribution for German Texts

We repeat this procedure for German texts, using *Scepter und Hammer* by Karl May. We already consumed its first 400000 letters for κ. Now we take the next 200000 letters—in fact we skip 801 letters in between—and form

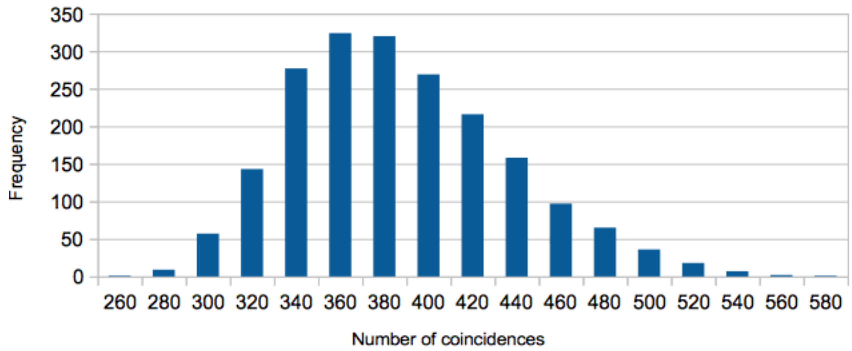

Figure 14: *Frequency of inner coincidence counts for 2000 German texts of 100 letters—to get φ values divide x-values by* 4950

Table 25: *Distribution of φ for 2000 German texts of 100 letters*

Minimum:	0.0517		
Median:	0.0752	Mean value:	0.0763
Maximum:	0.1152	Standard dev:	0.0099
1st quartile:	0.0689	5% quantile:	0.0618
3rd quartile:	0.0828	95% quantile:	0.0945

2000 text chunks with 100 letters each. The results are in Table 25 and Figure 14.

The Phi Distribution for Random Texts

And now the same procedure for random text. The results are in Table 26 and Figure 15.

Table 26: *Distribution of φ for 2000 random texts of 100 letters*

Minimum:	0.0331		
Median:	0.0398	Mean value:	0.0401
Maximum:	0.0525	Standard dev:	0.0028
1st quartile:	0.0382	5% quantile:	0.0360
3rd quartile:	0.0418	95% quantile:	0.0451

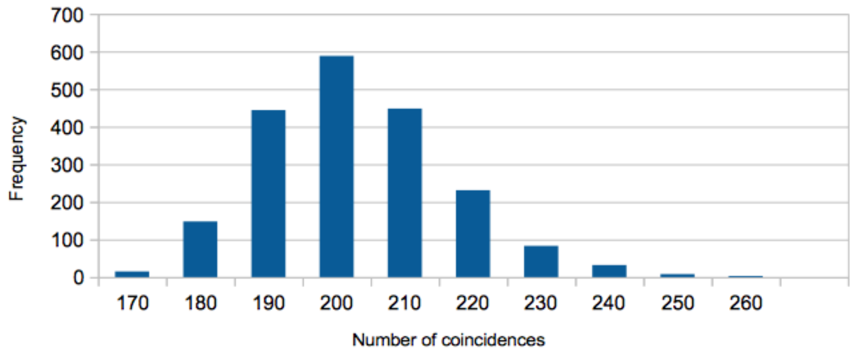

Figure 15: *Frequency of inner coincidence counts for 2000 random texts of 100 letters—to get φ values divide x-values by* 4950

Applications

To which questions from the introduction do these results apply?

We can decide whether a text is from a certain language. This includes texts that are monoalphabetically encrypted because φ is invariant under monoalphabetic substitution. And *we can recognize a monoalphabetically encrypted ciphertext.*

For both of these decision problems we calculate the coincidence index $\varphi(a)$ of our text a and decide "belongs to language" or "is monoalphabetic encrypted"—depending on our hypothesis—if $\varphi(a)$ reaches or surpasses the 95% quantile of φ for random texts of the same length—if we are willing to accept an error rate of the first kind of 5%.

For a text of 100 letters the threshold for φ is about 0.0451 by Table 26. Tables 24 and 25 show that English or German texts surpass this threshold with high probability: For both languages the test has a power of nearly 100%.

It makes sense to work with the more ambitious "significance level" of 1% = bound for the error of the first kind. For this we set the threshold to the 99% quantile of the φ distribution for random texts. Our experiment for texts of length 100 gives the empirical value of 0.0473, failing the empirical minimum for our 2000 English 100 letter texts, and sitting far below the empirical minimum for German. Therefore even at the 1%-level the test has a power of nearly 100%.

The Phi Distribution for 26 Letter Texts

Since the φ test performs so excellently for 100 letter texts we dare to look at 26 letter texts—a text length that occurs in the Meet-in-the-Middle attack against rotor machines.

Table 27: *Distribution of φ for 2000 English texts of 26 letters*

Minimum:	0.0227		
Median:	0.0585	Mean value:	0.0606
Maximum:	0.1385	Standard dev:	0.0154
1st quartile:	0.0492	5% quantile:	0.0400
3rd quartile:	0.0677	95% quantile:	0.0892

Table 28: *Distribution of φ for 2000 German texts of 26 letters*

Minimum:	0.0308		
Median:	0.0708	Mean value:	0.0725
Maximum:	0.1785	Standard dev:	0.0204
1st quartile:	0.0585	5% quantile:	0.0431
3rd quartile:	0.0831	95% quantile:	0.1108

Here we give the results as tables only.

The decision threshold on the 5%-level is 0.0585. For English texts the test has a power of only 50%, for German, near 75%. So we have a method to recognize monoalphabetic ciphertext that works fairly well for texts as short as 26 letters.

Table 29: *Distribution of φ for 2000 random texts of 26 letters*

Minimum:	0.0154		
Median:	0.0400	Mean value:	0.0401
Maximum:	0.0954	Standard dev:	0.0112
1st quartile:	0.0338	5% quantile:	0.0246
3rd quartile:	0.0462	95% quantile:	0.0585

12 Sinkov's Formula

Let's apply the approximative formulas for $\kappa_q(c)$ from Section 9 to the coincidence index of a periodically polyalphabetically encrypted text $c = f(a)$ with $a \in M$ of length r. In the case $l|r$ we get:

$$\begin{aligned}
\varphi(c) &= \frac{1}{r-1} \cdot [\kappa_1(c) + \cdots + \kappa_{r-1}(c)] \\
&\approx \frac{1}{r-1} \cdot \left[(\frac{r}{l} - 1) \cdot \kappa_M + (r - \frac{r}{l}) \cdot \kappa_{\Sigma^*} \right] \\
&= \frac{r-l}{r-1} \cdot \frac{1}{l} \cdot \kappa_M + \frac{r(l-1)}{l(r-1)} \cdot \kappa_{\Sigma^*} \\
&\approx \frac{1}{l} \cdot \kappa_M + \frac{l-1}{l} \cdot \kappa_{\Sigma^*},
\end{aligned}$$

since $\frac{r}{l} - 1$ summands scatter around κ_M, the other $r - \frac{r}{l}$ ones around κ_{Σ^*}.

In the same way for $l \not| r$ we get:

$$\begin{aligned}
\varphi(c) &\approx \frac{1}{r-1} \cdot \left[\frac{r-1}{l} \cdot \frac{q \cdot \kappa_{\Sigma^*} + (r-q) \cdot \kappa_M}{r} \right. \\
&\quad \left. + \frac{r-1}{l} \cdot \frac{q \cdot \kappa_M + (r-q) \cdot \kappa_{\Sigma^*}}{r} + (r-1) \cdot (1 - \frac{2}{l}) \cdot \kappa_{\Sigma^*} \right] \\
&= \frac{1}{l} \cdot \frac{r \cdot \kappa_{\Sigma^*} + r \cdot \kappa_M}{r} + (1 - \frac{2}{l}) \cdot \kappa_{\Sigma^*} \\
&= \frac{1}{l} \cdot \kappa_M + \frac{l-1}{l} \cdot \kappa_{\Sigma^*},
\end{aligned}$$

that is the same approximative formula in both cases. Note that this is a weighted mean.

$$\boxed{\varphi(c) \approx \frac{1}{l} \cdot \kappa_M + \frac{l-1}{l} \cdot \kappa_{\Sigma^*}}$$

For the example $M =$ "German" and $l = 7$ we therefore expect

$$\varphi(c) \approx \frac{1}{7} \cdot 0.0762 + \frac{6}{7} \cdot 0.0385 \approx 0.0439,$$

and this is in accordance with the empirical value from the former example. In general Table 30 and Figure 16 show the connection between period and expected coincidence index for a polyalphabetically encrypted German text. The situation for English is even worse.

If we solve the above formula for the period length l, we get Sinkov's formula:

$$l \cdot \varphi(c) \approx \kappa_M + (l-1) \cdot \kappa_{\Sigma^*},$$

$$l \cdot [\varphi(c) - \kappa_{\Sigma^*}] \approx \kappa_M - \kappa_{\Sigma^*},$$

Table 30: *Coincidence index and period length (for German)*

period	1	2	3	4	5
Coinc. index	0.0762	0.0574	0.0511	0.0479	0.0460
	6	7	8	9	10
	0.0448	0.0439	0.0432	0.0427	0.0423

period	10	20	30	40	50
Coinc index	0.0423	0.0404	0.0398	0.0394	0.0393
	60	70	80	90	100
	0.0391	0.0390	0.0390	0.0389	0.0389

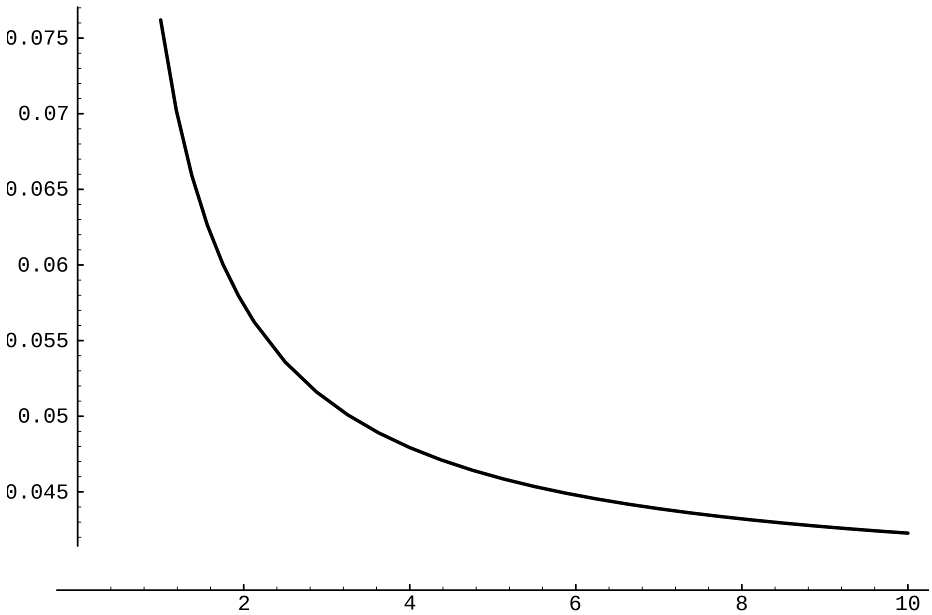

Figure 16: *Coincidence index and period length (for German)*

$$\boxed{l \approx \frac{\kappa_M - \kappa_{\Sigma^*}}{\varphi(c) - \kappa_{\Sigma^*}}.}$$

Remark. There are "more exact" versions of this formula. But these don't give better results due to the variation of $\varphi(c)$ and the numerical instability of the small denominator.

For our sample cryptanalysis we get

$$l \approx \frac{0.0762 - 0.0385}{0.0440 - 0.0385} \approx 6.85.$$

This is also evidence for 7 being the length of the period.

The problem with Sinkov's formula is the lack of numerical stability: the larger the period, the closer the coincidence index is to the value for random texts, as the table shows, that is, the closer the denominator in the formula is to 0.

Therefore the autocoincidence spectrum usually yields a better guess of the period. In fact Sinkov himself in his book [8] uses "his" formula—or rather the English equivalents of Table 30 and Figure 16—only for distinguishing between monoalphabetic and polyalphabetic ciphertexts. For determining the period he gives a very powerful test, see Section 13.

13 Sinkov's Test for the Period

We want to test a pretended period l whether it is the real period. We write the text in rows of width l and consider the columns.

- If l is the correct period, each column is monoalphabetically encrypted and has its coincidence index near the coincidence index of the plaintext language.
- Otherwise the columns are random garbage and have coincidence indices near the random value $\frac{1}{n}$. Or rather near the value for a polyalphabetic ciphertext of period (the true) l.

Maybe the columns are quite short, thus their coincidence indices are diffuse and give no clear impression. However we can put all the indices together without bothering about the different monoalphabets, and get a much more precise value, based on all the letters of the text.

Definition For a text $a \in \Sigma^*$ and $l \in \mathbb{N}_1$ the mean value

$$\bar{\varphi}_l(a) := \frac{1}{l} \cdot \sum_{i=0}^{l-1} \varphi(a_i a_{i+l} a_{i+2l} \ldots)$$

is called the Sinkov **statistic** of a of order l.

Note that $\bar{\varphi}_1 = \varphi$.

A Perl program, `phibar.pl`, is in `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Perl/`.

Example

Let us again examine the ciphertext from Section 9. We get the values:

$\bar{\varphi}_1(a)$	0.0442	$\bar{\varphi}_7(a)$	**0.0829**	$\bar{\varphi}_{13}(a)$	0.0444
$\bar{\varphi}_2(a)$	0.0439	$\bar{\varphi}_8(a)$	0.0443	$\bar{\varphi}_{14}(a)$	**0.0839**
$\bar{\varphi}_3(a)$	0.0440	$\bar{\varphi}_9(a)$	0.0427	$\bar{\varphi}_{15}(a)$	0.0432
$\bar{\varphi}_4(a)$	0.0438	$\bar{\varphi}_{10}(a)$	0.0421	$\bar{\varphi}_{16}(a)$	0.0439
$\bar{\varphi}_5(a)$	0.0430	$\bar{\varphi}_{11}(a)$	0.0426	$\bar{\varphi}_{17}(a)$	0.0444
$\bar{\varphi}_6(a)$	0.0435	$\bar{\varphi}_{12}(a)$	0.0432	$\bar{\varphi}_{18}(a)$	0.0419

The period 7 is overwhelmingly evident. The values other than at the multiples of 7 are in almost perfect compliance with a (German) ciphertext of period around 7.

A Short Ciphertext

Our example ciphertext was quite long, and it is no surprise that the statistical methods perform very well. To get a more realistic picture let us examine the following ciphertext of length 148:

```
MDJJL DSKQB GYMZC YKBYT ZVRYU PJTZN WPZXS KCHFG EFYFS ENVFW
KORMX ZQGYT KEDIQ WRVPM OYMQV DQWDN UBQQM XEQCA CXYLP VUOSG
EJYDS PYYNA XOREC YJAFA MFCOF DQKTA CBAHW FYJUI LXBYA DTT
```

The KASISKI test finds no reptitions of length 3 or more. It finds 16 repetitions of length 2 and no eye-catching pattern. The common factors 10 or 20 could be a hint at the correct period, but repetitions of length 2 are not overly convincing.

Repetition:	DS	SK	GY	YM	CY	BY	YT	TZ
Distance:	98	28	47	60	100	125	40	8
Repetition:	GE	FY	OR	MX	QW	DQ	AC	YJ
Distance:	60	94	60	31	12	50	40	21

The coincidence index of the text is 0.0386 and doesn't distinguish the ciphertext from random text. The first 40 values of the autocoincidence spectrum are

κ_1	κ_2	κ_3	κ_4	κ_5	κ_6	κ_7	κ_8
0.0270	0.0203	0.0541	0.0405	0.0405	0.0338	0.0405	**0.0676**
κ_9	κ_{10}	κ_{11}	κ_{12}	κ_{13}	κ_{14}	κ_{15}	κ_{16}
0.0270	0.0473	0.0270	**0.0676**	0.0405	0.0473	0.0541	0.0541
κ_{17}	κ_{18}	κ_{19}	κ_{20}	κ_{21}	κ_{22}	κ_{23}	κ_{24}
0.0203	0.0203	**0.0608**	0.0473	0.0473	0.0135	0.0541	0.0270
κ_{25}	κ_{26}	κ_{27}	κ_{28}	κ_{29}	κ_{30}	κ_{31}	κ_{32}
0.0338	0.0405	0.0541	**0.0811**	0.0338	0.0338	0.0405	0.0203
κ_{33}	κ_{34}	κ_{35}	κ_{36}	κ_{37}	κ_{38}	κ_{39}	κ_{40}
0.0068	0.0473	0.0473	0.0270	0.0405	0.0066	0.0203	0.0473

Values above 0.06 occur for shifts of 8, 12, 19, 28, the latter being the largest one. This makes a diffuse picture, giving slight evidence for a period of 28. Finally let's try SINKOV's test. It gives as its first 40 values:

$\bar{\varphi}_1$ 0.0386	$\bar{\varphi}_2$ 0.0413	$\bar{\varphi}_3$ 0.0386	$\bar{\varphi}_4$ 0.0492	$\bar{\varphi}_5$ 0.0421	$\bar{\varphi}_6$ 0.0441	$\bar{\varphi}_7$ 0.0433	$\bar{\varphi}_8$ 0.0471
$\bar{\varphi}_9$ 0.0330	$\bar{\varphi}_{10}$ 0.0505	$\bar{\varphi}_{11}$ 0.0265	$\bar{\varphi}_{12}$ **0.0591**	$\bar{\varphi}_{13}$ 0.0333	$\bar{\varphi}_{14}$ 0.0486	$\bar{\varphi}_{15}$ 0.0444	$\bar{\varphi}_{16}$ 0.0410
$\bar{\varphi}_{17}$ 0.0280	$\bar{\varphi}_{18}$ 0.0395	$\bar{\varphi}_{19}$ 0.0439	$\bar{\varphi}_{20}$ **0.0589**	$\bar{\varphi}_{21}$ 0.0357	$\bar{\varphi}_{22}$ 0.0264	$\bar{\varphi}_{23}$ 0.0476	$\bar{\varphi}_{24}$ 0.0548
$\bar{\varphi}_{25}$ 0.0507	$\bar{\varphi}_{26}$ 0.0359	$\bar{\varphi}_{27}$ 0.0444	$\bar{\varphi}_{28}$ 0.0488	$\bar{\varphi}_{29}$ 0.0368	$\bar{\varphi}_{30}$ **0.0622**	$\bar{\varphi}_{31}$ 0.0312	$\bar{\varphi}_{32}$ 0.0323
$\bar{\varphi}_{33}$ 0.0091	$\bar{\varphi}_{34}$ 0.0294	$\bar{\varphi}_{35}$ 0.0429	$\bar{\varphi}_{36}$ **0.0611**	$\bar{\varphi}_{37}$ 0.0541	$\bar{\varphi}_{38}$ 0.0307	$\bar{\varphi}_{39}$ 0.0256	$\bar{\varphi}_{40}$ 0.0542

The values for 12, 20, 30, and 36 stand somewhat out, followed by the values for 24, 37, and 40, then 10 and 25—again there is no clear favorite. Let's discuss the candidate values for the period and rate each criterion as "good", "weak", or "prohibitive".

Period?	Pros and cons
8	$\varphi(c)$ should be slightly larger (weak). Only 3 repetition distances are multiples of 8 (weak). κ_8 and κ_{16} are good, κ_{40} is weak, κ_{24} and κ_{32} are prohibitive. $\bar{\varphi}_8$ is weak, $\bar{\varphi}_{16}$ and $\bar{\varphi}_{32}$ are prohibitive, $\bar{\varphi}_{24}$ and $\bar{\varphi}_{40}$ are good.
10	$\varphi(c)$ should be slightly larger (weak). 7 repetition distances are multiples of 10 (good). κ_{10}, κ_{20}, and κ_{40} are weak, κ_{30} is prohibitive. $\bar{\varphi}_{10}$, $\bar{\varphi}_{20}$, $\bar{\varphi}_{30}$, and $\bar{\varphi}_{40}$ are good.
12	$\varphi(c)$ should be slightly larger (weak). 4 repetition distances are multiples of 12 (good). κ_{12} is good, κ_{24} and κ_{36} are prohibitive. $\bar{\varphi}_{12}$, $\bar{\varphi}_{24}$, and $\bar{\varphi}_{36}$ are good.
19	0 repetition distances are multiples of 19 (prohibitive). κ_{19} is good, κ_{38} is prohibitive. $\bar{\varphi}_{19}$ and $\bar{\varphi}_{38}$ are prohibitive.
20	6 repetition distances are multiples of 20 (good). κ_{20} and κ_{40} are weak. $\bar{\varphi}_{20}$ and $\bar{\varphi}_{40}$ are good.
24	0 repetition distances are multiples of 24 (prohibitive). κ_{24} is prohibitive. $\bar{\varphi}_{24}$ is good.
28	Only 1 repetition distance is a multiple of 28 (weak). κ_{28} is good. $\bar{\varphi}_{28}$ is weak.
30	3 repetition distances are multiples of 30 (good). κ_{30} is prohibitive. $\bar{\varphi}_{30}$ is good.
36	0 repetition distances are multiples of 36 (prohibitive). κ_{36} is prohibitive. $\bar{\varphi}_{36}$ is good.
37	0 repetition distances are multiples of 37 (prohibitive). κ_{37} is prohibitive. $\bar{\varphi}_{37}$ is good.

To assess these findings let us score the values "good" as $+1$, "weak" as 0, and "prohibitive" as -1. Note that 3 repetitions for period 8 are weaker than 3 repetitions for period 30. The candidates 19, 24, 36, and 37 have negative weights, the candidates 8 and 28, zero weights. We skip them in the first round. Positive weights have 10 (3 of 9), 12 (3 of 8), 20 (3 of 5), and 30 (1 of 3). We rank them by their relative weights: 20 with score $0.6 = 3/5$, then 12 with score 0.375, then 10 and 30 with scores 0.333.

The most promising approach to further cryptanalysis starts from the hypothetical period 20, see Section 15.

14 Kullback's Cross-Product Sum Statistic

For a decision whether two texts $a \in \Sigma^r$, $b \in \Sigma^q$ belong to the same language we could consider $\varphi(a||b)$, the coincidence index of the concatenated string $a||b$. It should approximately equal the coincidence index of the language, or—in the negative case—be significantly smaller. This index evaluates as

$$(q+r)(q+r-1)\cdot\varphi(a||b) = \sum_{s\in\Sigma}[m_s(a)+m_s(b)]\,[m_s(a)+m_s(b)-1]$$

$$= \sum_{s\in\Sigma} m_s(a)^2 + \sum_{s\in\Sigma} m_s(b)^2 + 2\cdot\sum_{s\in\Sigma} m_s(a)m_s(b) - r - q$$

In this expression we consider terms depending on only one of the texts as irrelevant for the decision problem. Omitting them we are left with the "cross-product sum"

$$\sum_{s\in\Sigma} m_s(a)m_s(b)$$

From another viewpoint we could consider the "Euclidean distance" of a and b in the n-dimensional space of single letter frequencies

$$d(a,b) = \sum_{s\in\Sigma}[m_s(a)-m_s(b)]^2 = \sum_{s\in\Sigma} m_s(a)^2 + \sum_{s\in\Sigma} m_s(b)^2 - 2\cdot\sum_{s\in\Sigma} m_s(a)m_s(b)$$

and this also motivates considering the cross-product sum. It should be large for texts from the same language, and small otherwise.

Definition

Let Σ be a finite alphabet. Let $a \in \Sigma^r$ and $b \in \Sigma^q$ be two texts of lengths $r, q \geq 1$. Then

$$\chi(a,b) := \frac{1}{rq}\cdot\sum_{s\in\Sigma} m_s(a)m_s(b),$$

where m_s denotes the frequency of the letter s in a text, is called **cross-product sum** of a and b.

For each pair $r, q \in \mathbb{N}_1$ this defines a map

$$\chi\colon \Sigma^r \times \Sigma^q \longrightarrow \mathbb{Q}.$$

A Perl program, `chi.pl`, is in `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Perl/`.

Transforming a and b by the same monoalphabetic substitution permutes the summands of $\chi(a,b)$. Therefore χ is invariant under monoalphabetic substitution.

Lemma 2 *Always $\chi(a,b) \leq 1$. Equality holds if and only if a and b consist of repetitions of the same single letter.*

Proof. We use the CAUCHY-SCHWARTZ inequality:

$$\begin{aligned}\chi(a,b)^2 = \left(\sum_{s\in\Sigma} \frac{m_s(a)}{r}\frac{m_s(b)}{q}\right)^2 \quad &\leq \quad \sum_{s\in\Sigma}\left(\frac{m_s(a)}{r}\right)^2 \cdot \sum_{s\in\Sigma}\left(\frac{m_s(b)}{q}\right)^2 \\ &\leq \quad \sum_{s\in\Sigma}\frac{m_s(a)}{r} \cdot \sum_{s\in\Sigma}\frac{m_s(b)}{q} = 1\end{aligned}$$

Equality holds if and only if

- $m_s(a) = c \cdot m_s(b)$ for all $s \in \Sigma$ with a fixed $c \in \mathbb{R}$,
- *and* all $\frac{m_s(a)}{r}$ and $\frac{m_s(b)}{q}$ are 0 or 1.

These two conditions together are equivalent with both of a and b consisting of only one—the same—repeated letter. $\diamond$

Considering the quantity $\psi(a) := \chi(a,a) = \sum_s m_s(a)^2/r^2$ doesn't make much sense for Corollary 1 of the Kappa-Phi-Theorem gives a linear (more exactly: affine) relation between ψ and φ:

Lemma 3 *For all $a \in \Sigma^r$, $r \geq 2$,*

$$\varphi(a) = \frac{r}{r-1} \cdot \psi(a) - \frac{1}{r-1}$$

Side Remark: COHEN's Kappa

In statistical texts one often encounters a related measure of coincidence between two series of observations: COHEN's kappa. It combines FRIEDMAN's kappa and KULLBACK's chi. Let $a = (a_1, \ldots, a_r)$, $b = (b_1, \ldots, b_r) \in \Sigma^r$ be two texts over the alphabet Σ (or two series of observations of data of some type). Then consider the matrix of frequencies

$$m_{st}(a,b) = \#\{i \mid a_i = s, b_i = t\} \quad \text{for } s,t \in \Sigma.$$

Its row sums are

$$m_s(a) = \#\{i \mid a_i = s\} = \sum_{t\in\Sigma} m_{st}(a,b),$$

its column sums are

$$m_t(b) = \#\{i \mid b_i = t\} = \sum_{s\in\Sigma} m_{st}(a,b),$$

its diagonal sum is

$$\sum_{s\in\Sigma} m_{ss}(a,b) = \sum_{s\in\Sigma} \#\{i \mid a_i = b_i = s\} = \#\{i \mid a_i = b_i\}.$$

The intermediate values from which COHEN's kappa is calculated are

$$p_0 = \frac{1}{r} \cdot \sum_{s\in\Sigma} m_{ss}(a,b) = \kappa(a,b) \quad \text{and} \quad p_e = \frac{1}{r^2} \cdot \sum_{s\in\Sigma} m_s(a)m_s(b) = \chi(a,b)$$

COHEN's kappa is defined for $a \neq b$ by

$$\mathsf{K}(a,b) := \frac{p_0 - p_e}{1 - p_e} = \frac{\kappa(a,b) - \chi(a,b)}{1 - \chi(a,b)}$$

If a and b are random strings with not necessarily uniform letter probabilities p_s, then K is asymptotically normally distributed with expectation 0 and variance

$$\frac{p_0 \cdot (1 - p_0)}{r \cdot (1 - p_0)^2}$$

Therefore its use is convenient for large series of observations—or large strings—but in cryptanalysis we mostly have to deal with short strings, and considering κ and χ separately may retain more information.

Mean Values

For a fixed $a \in \Sigma^r$ we determine the mean value of $\kappa(a,b)$ taken over all $b \in \Sigma^q$:

$$\begin{aligned}
\frac{1}{n^q} \cdot \sum_{b\in\Sigma^q} \chi(a,b) &= \frac{1}{n^q} \cdot \sum_{b\in\Sigma^q} \left[\frac{1}{rq} \cdot \sum_{s\in\Sigma} m_s(a)m_s(b) \right] \\
&= \frac{1}{rqn^q} \cdot \sum_{s\in\Sigma} m_s(a) \underbrace{\sum_{b\in\Sigma^q} m_s(b)}_{q\cdot n^{q-1}} \\
&= \frac{1}{rqn^q} \cdot r \cdot q \cdot n^{q-1} = \frac{1}{n}
\end{aligned}$$

where we used the corollary of Proposition 4.

In an analogous way we determine the mean value of $\chi(a, f_\sigma(b))$ for fixed $a, b \in \Sigma^r$ over all permutations $\sigma \in \mathcal{S}(\Sigma)$:

$$\frac{1}{n!} \cdot \sum_{\sigma\in\mathcal{S}(\Sigma)} \chi(a, f_\sigma(b)) = \frac{1}{rqn!} \cdot \sum_{\sigma\in\mathcal{S}(\Sigma)} \sum_{s\in\Sigma} m_s(a)m_s(f_\sigma(b))$$

As usual we interchange the order of summation, and evaluate the sum

$$\begin{aligned}
\sum_{\sigma\in\mathcal{S}(\Sigma)} m_s(f_\sigma(b)) &= \frac{1}{n} \cdot \sum_{t\in\Sigma} \sum_{\sigma\in\mathcal{S}(\Sigma)} m_t(f_\sigma(b)) \\
&= \frac{1}{n} \cdot \sum_{\sigma\in\mathcal{S}(\Sigma)} \underbrace{\sum_{t\in\Sigma} m_t(f_\sigma(b))}_{q} = \frac{1}{n} \cdot n! \cdot q = (n-1)! \cdot q
\end{aligned}$$

using the symmetry with respect to s. Therefore

$$\begin{aligned}\frac{1}{n!}\cdot\sum_{\sigma\in\mathcal{S}(\Sigma)}\chi(a,f_\sigma(b)) &= \frac{1}{rqn!}\cdot\sum_{s\in\Sigma}m_s(a)\cdot\sum_{\sigma\in\mathcal{S}(\Sigma)}m_s(f_\sigma(b))\\ &= \frac{1}{rqn!}\cdot r\cdot(n-1)!\cdot q=\frac{1}{n}\end{aligned}$$

Note that this conclusion also holds for $a = b$.

This derivation shows:

Proposition 5 (i) *The mean value of* $\chi(a, b)$ *over all texts* $b \in \Sigma^*$ *of a fixed length* q *is* $\frac{1}{n}$ *for all* $a \in \Sigma^*$.

(ii) *The mean value of* $\chi(a, b)$ *over all* $a \in \Sigma^r$ *and* $b \in \Sigma^q$ *is* $\frac{1}{n}$ *for all* $r, q \in \mathbb{N}_1$.

(iii) *The mean value of* $\chi(a, f_\sigma(b))$ *over all monoalphabetic substitutions with* $\sigma \in \mathcal{S}(\Sigma)$ *is* $\frac{1}{n}$ *for each pair* $a, b \in \Sigma^*$.

(iv) *The mean value of* $\chi(f_\sigma(a), f_\tau(b))$ *over all pairs of monoalphabetic substitutions, with* $\sigma, \tau \in \mathcal{S}(\Sigma)$, *is* $\frac{1}{n}$ *for each pair* $a, b \in \Sigma^*$.

Interpretation

- For a given text a and a “random” text b we have $\chi(a, b) \approx \frac{1}{n}$.
- For “random” texts a and b we have $\chi(a, b) \approx \frac{1}{n}$.
- For given texts a and b and a “random” monoalphabetic substitution f_σ we have $\chi(a, f_\sigma(b)) \approx \frac{1}{n}$. This remark justifies treating a nontrivially monoalphabetically encrypted text as random with respect to χ and plaintext.
- For given texts a and b and two “random” monoalphabetic substitutions f_σ, f_τ we have $\chi(f_\sigma(a), f_\tau(b)) \approx \frac{1}{n}$.

Empirical Results

We collect empirical results for 2000 pairs of 100 letter texts using `chistat.pl`, from `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Perl/`. For English we use the book *Dr Thorndyke Short Story Omnibus* by R. Austin Freeman from Project Gutenberg. We extract a first part of 402347 letters (`Thorn1.txt`) and take the first 400000 of them for our statistic. In the same way for German we use *Die Juweleninsel* by Karl May from Karl-May-Gesellschaft (`Juwelen1.txt`, 434101 letters). For random texts we generate 400000 letters by Perl's random generator (`RndT400K.txt`). (All texts in `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Files/`.)

The results are in Tables 31, 32, and 33. We see that χ—in contrast with the coincidence index κ—performs extremely well, in fact in our experiments it even completely separates English and German texts from random texts of length 100. It is a test with power near 100% and error probability near 0%. The χ test even distinguishes between English and German texts at the 5% error level with a power of almost 75%. For this assertion compare the 95% quantile for English with the first quartile for German.

Table 31: *Distribution of χ for 2000 English text pairs of 100 letters*

Minimum:	0.0500		
Median:	0.0660	Mean value:	0.0663
Maximum:	0.0877	Standard dev:	0.0049
1st quartile:	0.0630	5% quantile:	0.0587
3rd quartile:	0.0693	95% quantile:	0.0745

The results for 100 letter texts encourage us to try 26 letter texts. To this end we need 104000 letters for each language. We extract the next 104009 letters from *Dr Thorndyke Short Story Omnibus* (`Thorn2.txt`), and the next 104293 letters from *Die Juweleninsel* (`Juwelen2.txt`). We construct random text by taking 104000 random numbers between 0 and 25 from `random.org` (`RndT104K.txt`). The results are in Tables 34, 35, and 36. The χ-test is quite strong even for 26 letters: At the 5% error level its power is around 91% for English, 98% for German.

Table 32: *Distribution of χ for 2000 German text pairs of 100 letters*

Minimum:	0.0578		
Median:	0.0792	Mean value:	0.0794
Maximum:	0.1149	Standard dev:	0.0074
1st quartile:	0.0742	5% quantile:	0.0677
3rd quartile:	0.0840	95% quantile:	0.0923

Table 33: *Distribution of* χ *for 2000 random text pairs of 100 letters*

Minimum:	0.0337		
Median:	0.0400	Mean value:	0.0400
Maximum:	0.0475	Standard dev:	0.0020
1st quartile:	0.0387	5% quantile:	0.0367
3rd quartile:	0.0413	95% quantile:	0.0433

Table 34: *Distribution of* χ *for 2000 English text pairs of 26 letters*

Minimum:	0.0266		
Median:	0.0666	Mean value:	0.0666
Maximum:	0.1169	Standard dev:	0.0120
1st quartile:	0.0577	5% quantile:	0.0488
3rd quartile:	0.0740	95% quantile:	0.0873

Table 35: *Distribution of* χ *for 2000 German text pairs of 26 letters*

Minimum:	0.0325		
Median:	0.0784	Mean value:	0.0793
Maximum:	0.1538	Standard dev:	0.0154
1st quartile:	0.0680	5% quantile:	0.0562
3rd quartile:	0.0888	95% quantile:	0.1065

Table 36: *Distribution of* χ *for 2000 random text pairs of 26 letters*

Minimum:	0.0178		
Median:	0.0385	Mean value:	0.0386
Maximum:	0.0680	Standard dev:	0.0075
1st quartile:	0.0340	5% quantile:	0.0266
3rd quartile:	0.0429	95% quantile:	0.0518

15 Adjusting the Columns of a Disk Cipher

As a last application in this chapter we look at the problem: How to adjust the alphabets in the columns of a disk cipher? From Chapter 2 we know that this works only when the primary alphabet is known.

Imagine a ciphertext from a disk cipher whose period l we know already. Write the ciphertext in rows of length l. Then the columns are monoalphabetically encrypted, each with (in most cases) another alphabet. By Proposition 5 (iv) we expect a χ-value of about $\frac{1}{n}$ for each pair of columns. Since the alphabets for the columns are secondary alphabets of a disk cipher they differ only by a relative shift in the alphabet. There are 26 different possible shifts. These can be checked by exhaustion: We try all 26 possibilities (including the trivial one, bearing in mind that two columns can have the same alphabet). The perfect outcome would be 25 values near $\frac{1}{n}$, and one outcome around the coincidence index of the plaintext language, clearly indicating the true alphabet shift. The experimental results of Section 14 give hope that real outcome should approximate the ideal one in a great number of cases.

Example 1

Let us try out this idea for the ciphertext from Section 9. We are pretty sure that the period is 7. (And we also adjusted the columns by visual inspection in Chapter 2.) The first two columns are

```
ARCYPMEAZKRWKHZLRXTRTMYYRLMTVYCMRBZZKOLKKTKOTCUKKOMVBLYUYYZALR
  OEKWZMWZZRYZOOTUYURMTYYSOZEKLYVUYBYTZYKOVMYYMZMZVYROKYTYMUWZ
  PZTZLSPLYLZVYYYBYMQMWWRXZYOKKMYZTZAKQZZT
OMZYYDMYPQMHMFKAMMAACDNNZPIMYZHCJSCNCJQMMYLEMMPNNPZYSNYHPNMOAM
  CAJMPZIVNMPADAHNKFNNAHNVFJHFXNYPNSYFMKNFMDNPZFGJMVMCMXYZZMQC
  MSYIMVAMKZOANZVSZFKMYEMQHZQMNDPMHDMKIYJF
```

Using the Perl script `adjust.pl` we get the results

Shift:	0	1	2	3	4	5	6
χ:	0.0499	0.0365	0.0348	0.0285	0.0320	0.0341	0.0298
7	8	9	10	11	12	13	14
0.0416	0.0307	0.0421	0.0402	0.0448	**0.0799**	0.0495	0.0373
15	16	17	18	19	20	21	22
0.0375	0.0293	0.0330	0.0276	0.0307	0.0306	0.0316	0.0352
23	24	25					
0.0338	0.0461	0.0529					

The result is clear without ambiguity: The correct shift is 12. Going through all $7 \times 6/2 = 21$ pairs of columns we use the Perl script `coladj.pl` and get results in Table 37 that are consistent with each other and with the results of Chapter 2.

Table 37: *The optimal alphabet shifts for 7 columns*

Column:	0	1	2	3	4	5
1	12					
2	4	18				
3	15	3	11			
4	10	24	6	21		
5	24	12	20	9	14	
6	3	17	25	14	19	5

Example 2

The best guess for the period of the short ciphertext of Section 13 was $l = 20$. Therefore we consider 20 columns of lengths 8 or 7:

```
M D J J L D S K Q B G Y M Z C Y K B Y T
Z V R Y U P J T Z N W P Z X S K C H F G
E F Y F S E N V F W K O R M X Z Q G Y T
K E D I Q W R V P M O Y M Q V D Q W D N
U B Q Q M X E Q C A C X Y L P V U O S G
E J Y D S P Y Y N A X O R E C Y J A F A
M F C O F D Q K T A C B A H W F Y J U I
L X B Y A D T T
```

We have to assume the primary alphabet as known in order to know how to shift the columns, that is, how to identify the distance of the secondary alphabets of two columns relative to each other. The primary alphabet is `QWERTZUABCDFGHIJKLMNOPSVXY`, the complete alphabet table is in Table 38.

The method from Example 1 gives $20 \times 19/2 = 190$ proposals for optimal shifts between columns. However even for the first columns we already get inconsistent results. We face a complex optimization problem. Instead of continuing with the next columns we better would follow a proposal by Sinkov: Pick up the highest χ-values and try to build clusters of fitting columns. But also this approach fails. After several hours off the track we try to understand why.

Let us imagine a plaintext of the same length, written in rows of length 20, columns of length 7 or 8. Take two columns that each have one letter twice and five or six single letters. Shifting the alphabets in such a way that the "twins" become identical letters, they contribute a summand of

$$\frac{4}{49} \approx 0.0818 \text{ for lengths } 7/7, \ \frac{4}{56} \approx 0.0714 \text{ for } 8/7, \ \frac{4}{64} \approx 0.0625 \text{ for } 8/8,$$

Table 38: *The alphabet table used in the example*

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Q	W	E	R	T	Z	U	A	B	C	D	F	G	H	I	J	K	L	M	N	O	P	S	V	X	Y
W	E	R	T	Z	U	A	B	C	D	F	G	H	I	J	K	L	M	N	O	P	S	V	X	Y	Q
E	R	T	Z	U	A	B	C	D	F	G	H	I	J	K	L	M	N	O	P	S	V	X	Y	Q	W
R	T	Z	U	A	B	C	D	F	G	H	I	J	K	L	M	N	O	P	S	V	X	Y	Q	W	E
T	Z	U	A	B	C	D	F	G	H	I	J	K	L	M	N	O	P	S	V	X	Y	Q	W	E	R
Z	U	A	B	C	D	F	G	H	I	J	K	L	M	N	O	P	S	V	X	Y	Q	W	E	R	T
U	A	B	C	D	F	G	H	I	J	K	L	M	N	O	P	S	V	X	Y	Q	W	E	R	T	Z
A	B	C	D	F	G	H	I	J	K	L	M	N	O	P	S	V	X	Y	Q	W	E	R	T	Z	U
B	C	D	F	G	H	I	J	K	L	M	N	O	P	S	V	X	Y	Q	W	E	R	T	Z	U	A
C	D	F	G	H	I	J	K	L	M	N	O	P	S	V	X	Y	Q	W	E	R	T	Z	U	A	B
D	F	G	H	I	J	K	L	M	N	O	P	S	V	X	Y	Q	W	E	R	T	Z	U	A	B	C
F	G	H	I	J	K	L	M	N	O	P	S	V	X	Y	Q	W	E	R	T	Z	U	A	B	C	D
G	H	I	J	K	L	M	N	O	P	S	V	X	Y	Q	W	E	R	T	Z	U	A	B	C	D	F
H	I	J	K	L	M	N	O	P	S	V	X	Y	Q	W	E	R	T	Z	U	A	B	C	D	F	G
I	J	K	L	M	N	O	P	S	V	X	Y	Q	W	E	R	T	Z	U	A	B	C	D	F	G	H
J	K	L	M	N	O	P	S	V	X	Y	Q	W	E	R	T	Z	U	A	B	C	D	F	G	H	I
K	L	M	N	O	P	S	V	X	Y	Q	W	E	R	T	Z	U	A	B	C	D	F	G	H	I	J
L	M	N	O	P	S	V	X	Y	Q	W	E	R	T	Z	U	A	B	C	D	F	G	H	I	J	K
M	N	O	P	S	V	X	Y	Q	W	E	R	T	Z	U	A	B	C	D	F	G	H	I	J	K	L
N	O	P	S	V	X	Y	Q	W	E	R	T	Z	U	A	B	C	D	F	G	H	I	J	K	L	M
O	P	S	V	X	Y	Q	W	E	R	T	Z	U	A	B	C	D	F	G	H	I	J	K	L	M	N
P	S	V	X	Y	Q	W	E	R	T	Z	U	A	B	C	D	F	G	H	I	J	K	L	M	N	O
S	V	X	Y	Q	W	E	R	T	Z	U	A	B	C	D	F	G	H	I	J	K	L	M	N	O	P
V	X	Y	Q	W	E	R	T	Z	U	A	B	C	D	F	G	H	I	J	K	L	M	N	O	P	S
X	Y	Q	W	E	R	T	Z	U	A	B	C	D	F	G	H	I	J	K	L	M	N	O	P	S	V
Y	Q	W	E	R	T	Z	U	A	B	C	D	F	G	H	I	J	K	L	M	N	O	P	S	V	X

to the χ-value. If accidentally there is another common letter, these values rise to

$$\frac{5}{49} \approx 0.1020 \text{ for lengths } 7/7,\ \frac{5}{56} \approx 0.0893 \text{ for } 8/7,\ \frac{5}{64} \approx 0.0781 \text{ for } 8/8.$$

And therefore we'll get many false alarms that will make the task of finding the correct solution very time-consuming. An experiment with plaintext comfirms this. Here all shifts should be 0, however we found the maximal χ-value for a shift of 0 in less then 20% of all cases.

To get better chances for success we need some known plaintext or more ciphertext or luck. We had luck and got more ciphertext. The following two messages b and c,

```
AWYFN DHZPE PENES YGAVO YHGAD VTNLL TFKKH FHGYT DOGJI HJHHB
OOYFV EWDSJ MOIFY DRTLA BRRFE ZQGYQ AVYCH BQZPR RZTTH IONZE
SCEFH EFJBJ RNRWE TGVZR EYIIQ IZRWT OLGOC ICLFS EMYAH E

LIGJC KTNLF KBMZH XYWFB UWVPC RNYAJ WEVKV BRVPN PXYOT KVGLE
MBVHE WFZSM UOWFI EYXLB XRRKC XKGPT YONFY DKZLU CXRDC YJWZT
UWPDS VZWNU KORLK WUXUO WVHFL IEGXJ ZUKGC YJVDN EFYDK GJZON
BYXEV EWQSD MMHSS GJ
```

could be encrypted with the same key. Number 1 and 2 have a coincidence index $\kappa(a, b) \approx 0.0411$ only. But $\kappa(a, c) \approx 0.0811$, $\kappa(b, c) \approx 0.1027$. For both b and c the period 20 is confirmed by the SINKOV statistic and also by the autocoincidence spectrum. Therefore we align all three messages below each other with rows of length 20. From bad experience we know we should proceed very thoughtfully. Therefore we first look at the letter frequencies in the single columns (of lengths 22 to 25). The columns 2, 3, and 12 contain a letter in 7 exemplars. We try to adjust these columns in such a way that the most frequent letters match. For column 3 relative to column 2 we get a χ-value of 0.1072 for a shift of 14, the next χ-value being 0.0608. If we write the columns as rows, the result looks like this

```
Column 02: JRYDQYCBYGGIYEIYGVYWNPHYH
Column 03: JYFIQDOYFAJFCFIAJPOFFDFDS
shifted:   RHYEIXBHYPRYVYEPRCBYYXYXD
```

In both cases the most frequent letter with 7 occurrences is `Y`. For column 12 we get the optimal shift 22 relative to column 2 with a χ-value of 0.1273, the next χ-value being 0.0836. This also looks good and gives the result

```
Column 02: JRYDQYCBYGGIYEIYGVYWNPHYH
Column 12: MZRMYRANKYRTRGMVVRRRKX
shifted:   IWYIPYRJGPYQYBINNYYYGO
```

Also in the shifted column 12 the letter Y occurs 7 times. If we are right, comparing columns 3 and 12 should lead to the same result. Indeed the optimal shift is 8 with $\chi \approx 0.1109$, the next χ-value being 0.0727.

This makes us confident that we are on the right track, and encourages us to set Y it to plaintext e. We continue our task under the hypothesis that columns 2, 3, and 12 match with the given shifts as

```
...
JRYDQYCBYGGIYEIYGVYWNPHYH
RHYEIXBHYPRYVYEPRCBYYXYXD
...
IWYIPYRJGPYQYBINNYYYGO
...
```

We take this text fragment as cluster "A" and try to match further columns. First we take columns where the most frequent letters occur 6 or 5 times.

```
A vs  5: Optimal shift is 15 with chi = 0.0906 (next is 0.0683)
A vs  8: Optimal shift is  8 with chi = 0.1092 (next is 0.0758)
A vs 14: Optimal shift is 16 with chi = 0.1092 (next is 0.0859)

A vs  0: Optimal shift is 23 with chi = 0.0878 (next is 0.0817)
A vs  5: Optimal shift is  0 with chi = 0.0809 (next is 0.0619)
A vs  9: Optimal shift is 21 with chi = 0.0966 (next is 0.0663)
```

The most convincing match is with column 8, therefore we join it to our cluster, forming cluster "B":

```
...
JRYDQYCBYGGIYEIYGVYWNPHYH
RHYEIXBHYPRYVYEPRCBYYXYXD
...
BHNRLWGRYPYRKCYJYYYWUE
...
IWYIPYRJGPYQYBINNYYYGO
...
```

Looking at the distribution of letters the Y stands out by far—that is no surprise because we picked columns with the most frequent letters and matched these. As a more meaningful check we transform our cluster to (presumed) plaintext; this means decrypting the fragments with the secondary alphabet that transforms e to Y, that is PSVXYQWERTZUABCDFGHIJKLMNO. This gives the supposed plaintext fragment (to be read top down):

```
...
uiepfeonerrtehtercegyases
isehtdnseaiecehaioneededp
...
nsyiwgrieaeivoeueeeglh
...
tgetaeiuraefentyyeeerz
...
```

This looks promising. Trying to extend this cluster by a formal procedure is dangerous because there could be columns with a most frequent (plaintext) letter other then `e`. Instead we look at neighboring columns, say at column 4 that should give a readable continuation of columns 2 and 3, in particular extending the digraph `th` in a meaningful way. The proposed shift should have a `Y` (for `e`) as 15th letter, or maybe a `P` (for `a`), or an `R` (for `i`).

Cluster B versus column 4 yields the optimal shift 3 with $\chi \approx 0.0753$, the 15th letter being `R` (for `i`). The next best values are $\chi \approx 0.0664$ for a shift of 12, the 15th letter then being `G` (for `r`), and $\chi \approx 0.0604$ for a shift of 25, the 15th letter being `Y` (for `e`). To decide between these possible solutions we decrypt the shifted columns and get the proposed cleartext columns

```
zoeiaetpbswhvvivrrmwhezye
ixnrjncykbfqeereaavfqnihn
vkaewaplxosdrrernnisdavua
```

Joining them to columns 3 and 4 the first one looks somewhat inauspicuous but possible, the second one looks awkward, the third one looks best and is our first choice. This gives the three adjacent columns

```
uiepfeonerrtehtercegyases
isehtdnseaiecehaioneededp
vkaewaplxosdrrernnisdavua
```

and the new cluster "C" of (monoalphabetic) ciphertext, comprising columns 2, 3, 4, 8, 12:

```
...
JRYDQYCBYGGIYEIYGVYWNPHYH
RHYEIXBHYPRYVYEPRCBYYXYXD
KZPYLPDUMCHXGGYGBBRHXPKJP
...
BHNRLWGRYPYRKCYJYYYWUE
...
IWYIPYRJGPYQYBINNYYYGO
...
```

Note that for joining further columns we must not work with the (proposed) plaintext columns because the transformation between plaintext and ciphertext is not a simple shift.

Comparing the adjacent columns with cluster C we obtain

```
C vs  1: Optimal shift is  1 with chi = 0.0642 (next is 0.0632)
C vs  5: Optimal shift is 15 with chi = 0.0844 (next is 0.0686)
C vs  7: Optimal shift is 20 with chi = 0.0676 (next is 0.0621)
C vs  9: Optimal shift is  6 with chi = 0.0695 (next is 0.0653)
C vs 11: Optimal shift is  5 with chi = 0.0695 (next is 0.0638)
C vs 13: Optimal shift is 23 with chi = 0.0684 (next is 0.0588)
```

The best value seems that for column 13, so let's try this one first (skipping the dead end via column 5). The new cluster D is

```
...
JRYDQYCBYGGIYEIYGVYWNPHYH     uiepfeonerrtehtercegyases
RHYEIXBHYPRYVYEPRCBYYXYXD     isehtdnseaiecehaioneededp
KZPYLPDUMCHXGGYGBBRHXPKJP     vkaewaplxosdrrernnisdavua
...
BHNRLWGRYPYRKCYJYYYWUE        nsyiwgrieaeivoeueeeglh
...
IWYIPYRJGPYQYBINNYYYGO        tgetaeiuraefentyyeeerz
EPJVIYDYHBBWXLEHDHAICY        hauctepesnngdwhspsmtoe
...
```

This looks good, and detecting the two `th`'s between the cleartext columns 12 and 13 we try column 14 next.

```
D vs 14: Optimal shift is 16 with chi = 0.0945 (next is 0.0793)
```

If we rely on this result, we get the next cluster E:

```
...
JRYDQYCBYGGIYEIYGVYWNPHYH     uiepfeonerrtehtercegyases
RHYEIXBHYPRYVYEPRCBYYXYXD     isehtdnseaiecehaioneededp
KZPYLPDUMCHXGGYGBBRHXPKJP     vkaewaplxosdrrernnisdavua
...
BHNRLWGRYPYRKCYJYYYWUE        nsyiwgrieaeivoeueeeglh
...
IWYIPYRJGPYQYBINNYYYGO        tgetaeiuraefentyyeeerz
EPJVIYDYHBBWXLEHDHAICY        hauctepesnngdwhspsmtoe
PBDCAPHBYCIYIPYCIPPEPC        anpomasneotetaeotaahao
...
```

Good! Let's continue with column 15:

```
E vs 15: Optimal shift is  0 with chi = 0.0719 (next is 0.0574)
```

Joining the resulting "cleartext" to columns 12, 13, 14 gives the disturbing result

```
tgetaeiuraefentyyeeerz
hauctepesnngdwhspsmtoe
anpomasneotetaeotaahao
evkpceqeqhktjtdngdegeh
```

Therefore we dismiss this proposal. Unfortunately also the next best χ-value gives no sensible result. On the other hand the shifts giving a possible complement to the `th` have a quite small χ-value. Therefore we leave column 15 and retry column 1:

```
E vs  1: Optimal shift is  1 with chi = 0.0631 (next is 0.0577)
```

This would give us cluster F:

```
...
FXGRCKGYEIPPXDQNJEYPPEXGN     qdriovrehtaadpfyuheaahdry
JRYDQYCBYGGIYEIYGVYWNPHYH     uiepfeonerrtehtercegyases
RHYEIXBHYPRYVYEPRCBYYXYXD     isehtdnseaiecehaioneededp
KZPYLPDUMCHXGGYGBBRHXPKJP     vkaewaplxosdrrernnisdavua
...
BHNRLWGRYPYRKCYJYYYWUE        nsyiwgrieaeivoeueeeglh
...
IWYIPYRJGPYQYBINNYYYGO        tgetaeiuraefentyyeeerz
EPJVIYDYHBBWXLEHDHAICY        hauctepesnngdwhspsmtoe
PBDCAPHBYCIYIPYCIPPEPC        anpomasneotetaeotaahao
...
```

The plaintext now begins with `.quiv...`. A dictionary search finds hits such as "equivalent", "equivocal", and "a quiver". We compare cluster F with column 1 and look for shifts that make the first letter `a` (`P` in our secondary alphabet) or `e` (`Y`). We have luck! The optimal shift gives `e`, so we take this as our favourite solution:

```
F vs  0: Optimal shift is  7 with chi = 0.0717 (next is 0.0696)
```

and form the next cluster G:

```
YGCVHCYXIULYIRCCXHEHUHBCY     erocsoedtlwetioodshslsnoe
FXGRCKGYEIPPXDQNJEYPPEXGN     qdriovrehtaadpfyuheaahdry
JRYDQYCBYGGIYEIYGVYWNPHYH     uiepfeonerrtehtercegyases
RHYEIXBHYPRYVYEPRCBYYXYXD     isehtdnseaiecehaioneededp
KZPYLPDUMCHXGGYGBBRHXPKJP     vkaewaplxosdrrernnisdavua
...
BHNRLWGRYPYRKCYJYYYWUE        nsyiwgrieaeivoeueeeglh
...
IWYIPYRJGPYQYBINNYYYGO        tgetaeiuraefentyyeeerz
EPJVIYDYHBBWXLEHDHAICY        hauctepesnngdwhspsmtoe
PBDCAPHBYCIYIPYCIPPEPC        anpomasneotetaeotaahao
...
```

Noting the fragments `ciphe` in "line" 4 (fourth column in the schema above) and `ipher` in "line" 14, we cannot resist completing them as `cipher`.

```
G vs  5: Optimal shift is 11 with chi = 0.0708 (next is 0.0697)
G vs 19: Optimal shift is 21 with chi = 0.0775 (next is 0.0585)
```

Note that we now see how misleading our former results for column 5 were. This is caused by the six `a`'s in this column that the χ-method tried to associate with the `e`'s of other columns.

Adding both of these results in one step gives cluster H:

```
YGCVHCYXIULYIRCCXHEHUHBCY     erocsoedtlwetioodshslsnoe
FXGRCKGYEIPPXDQNJEYPPEXGN     qdriovrehtaadpfyuheaahdry
JRYDQYCBYGGIYEIYGVYWNPHYH     uiepfeonerrtehtercegyases
RHYEIXBHYPRYVYEPRCBYYXYXD     isehtdnseaiecehaioneededp
KZPYLPDUMCHXGGYGBBRHXPKJP     vkaewaplxosdrrernnisdavua
YDLKHDYYYGEYVLRLZMZLYGRWW     alsrolaaandaysesgtgsanecc
...
BHNRLWGRYPYRKCYJYYYWUE        nsyiwgrieaeivoeueeeglh
...
IWYIPYRJGPYQYBINNYYYGO        tgetaeiuraefentyyeeerz
EPJVIYDYHBBWXLEHDHAICY        hauctepesnngdwhspsmtoe
PBDCAPHBYCIYIPYCIPPEPC        anpomasneotetaeotaahao
...
YAYIAECJYDPVXLRIHYYJIZ        emetmhouepacdwitseeutk
```

We see that column 6 should start with `l` (`U`). And this is also the "χ-optimal" solution:

```
H vs  6: Optimal shift is 10 with chi = 0.0734 (next is 0.0554)
```

And column 7 should start with `e` (`Y`):

```
H vs  7: Optimal shift is 20 with chi = 0.0647 (next is 0.0639)
```

We are not amused, also the next best χ is unwanted. However the shift that gives `e` has a χ-value of 0.0639 that is acceptable. We fill in columns 6 and 7:

```
YGCVHCYXIULYIRCCXHEHUHBCY     erocsoedtlwetioodshslsnoe
FXGRCKGYEIPPXDQNJEYPPEXGN     qdriovrehtaadpfyuheaahdry
JRYDQYCBYGGIYEIYGVYWNPHYH     uiepfeonerrtehtercegyases
RHYEIXBHYPRYVYEPRCBYYXYXD     isehtdnseaiecehaioneededp
KZPYLPDUMCHXGGYGBBRHXPKJP     vkaewaplxosdrrernnisdavua
YDLKHDYYYGEYVLRLZMZLYGRWW     alsrolaaandaysesgtgsanecc
UYRHGCDIVIYHDPJIRACQJGYY      leisroptctespautimofuree
YHUUCBYHIESHIXGEBPAIDPI       esllonesthbstdrhnamtpat
BHNRLWGRYPYRKCYJYYYWUE        nsyiwgrieaeivoeueeeglh
...
IWYIPYRJGPYQYBINNYYYGO        tgetaeiuraefentyyeeerz
EPJVIYDYHBBWXLEHDHAICY        hauctepesnngdwhspsmtoe
PBDCAPHBYCIYIPYCIPPEPC        anpomasneotetaeotaahao
...
YAYIAECJYDPVXLRIHYYJIZ        emetmhouepacdwitseeutk
```

It's time, for easier reading, to arrange our findings in the right order where "columns" are columns:

```
equivalen...tha....e   rdiskless...gan....m
oreeasily...eup....e   ciphersli...tco....t
softworow...atm....m   ovedalong...eea....h
eronpaper...ips....o   denslats
theexacti...uen....u   ltraonthe...rse....e
warisdeba...ano....p   eatedasse...ent....a
tdecrypti...fge....c   iphersadv...edt....d
oftheeuro...nwa....w   oyears
duringthe...the....i   shcontinu...yso....t
heenigmae...ypt....s   sagessome...esa....e
layedafte...ema....e   shadanupg...eth....u
ndseveral...roa....t   oreduceth...zeo....k
eyspace
```

Now its easy to complete the text: In the first row read `equivalent` and complete column 9. In the fourth row read `cipher slide` and complete column 10. Then read `with` in the first row and complete column 11. Then in the last two rows we recognize `the size of ... keyspace`, this allows us to complete column 15. Now in the first two rows we read `cipher disk` and complete the remaining columns 16, 17, 18.

This is the final solution:

```
equivalentwithaciphe   rdisklesselegantbutm
oreeasilymadeupisthe   ciphers1ideitconsist
softworowsthatmaybem   ovedalongsideeachoth
eronpaperstripsorwoo   denslats
theexactinfluenceofu   ltraonthecourseofthe
warisdebatedanoftrep   eatedassessmentistha
tdecryptionofgermanc   iphersadvancedtheend
oftheeuropeanwarbytw   oyears
duringthewarthebriti   shcontinuallysolvedt
heenigmaencryptedmes   sagessometimesabitde
layedafterthemachine   shadanupgradetheyfou
ndseveralapproachest   oreducethesizeofthek
eyspace
```

16 Modeling a Language by a MARKOV Process

For deriving theoretical results a common model of language is the interpretation of texts as results of MARKOV processes. This model was introduced by SHANNON in his fundamental papers published after World War II.

If we look at letter frequencies only, we define a MARKOV process of order 0. If we also incorporate bigram frequencies into our model, it becomes a MARKOV process of order 1, if we include trigram frequencies, of order 2, and so on.

In this section we want to derive theoretical expectation values for κ, φ, and χ. For this the order of the MARKOV model is irrelevant.

Message Sources

Let the alphabet Σ be equipped with a probability distribution, assigning the probability p_s to the letter $s \in \Sigma$. In particular $\sum_{s\in\Sigma} p_s = 1$. We call (Σ, p) a **message source** and consider random variables X in Σ, that is mappings $X\colon \Omega \longrightarrow \Sigma$ where Ω is a finite probability space with probability measure P, such that $P(X^{-1}s) = p_s$ for all $s \in \Sigma$.

Picking a letter of Σ at random from the message source is modeled as evaluating $X(\omega)$ for some $\omega \in \Omega$. We calculate the expectation values of the KRONECKER symbols for random variables $X, Y\colon \Omega \longrightarrow \Sigma$ and letters $s \in \Sigma$ where Y may belong to a message source (Σ, q) with a possibly different probability distribution $q = (q_s)_{s\in\Sigma}$:

$$\delta_{sX}(\omega) = \begin{cases} 1 & \text{if } X(\omega) = s \\ 0 & \text{otherwise} \end{cases} \qquad \delta_{XY}(\omega) = \begin{cases} 1 & \text{if } X(\omega) = Y(\omega) \\ 0 & \text{otherwise} \end{cases}$$

Lemma 4 (i) $\mathrm{E}(\delta_{sX}) = p_s$ *for all* $s \in \Sigma$.

(ii) *If X and Y are independent, then* $\mathrm{E}(\delta_{XY}) = \sum_{s\in\Sigma} p_s q_s$.

(ii) *If X and Y are independent, then δ_{sX} and δ_{sY} are independent.*

Proof. (i) Since δ takes only the values 0 and 1, we have

$$\mathrm{E}(\delta_{sX}) = 1 \cdot P(X^{-1}s) + 0 \cdot P(\Omega - X^{-1}s) = P(X^{-1}s) = p_s.$$

(ii) In the same way, using the independence of X and Y,

$$\begin{aligned} \mathrm{E}(\delta_{X,Y}) &= 1 \cdot P(\omega \mid X(\omega) = Y(\omega)) + 0 \cdot P(\omega \mid X(\omega) \neq Y(\omega)) \\ &= P(X = Y) = \sum_{s\in\Sigma} P(X^{-1}s \cap Y^{-1}s) \\ &= \sum_{s\in\Sigma} P(X^{-1}s) \cdot P(Y^{-1}s) = \sum_{s\in\Sigma} p_s q_s \end{aligned}$$

(iii) $\delta_{sX}^{-1}(1) = \{\omega \,|\, X(\omega) = s\} = X^{-1}s$, and $\delta_{sX}^{-1}(0) = \Omega - X^{-1}s$. The same for Y. The assertion follows because $P(X^{-1}s \cap Y^{-1}s) = P(X^{-1}s) \cdot P(Y^{-1}s)$. $\diamond$

Picking a random text of length r is modeled by evaluating an r-tuple of random variables at some ω. This leads to the following definition:

Definition. A **message** of length r from the message source (Σ, p) is a sequence $X = (X_1, \ldots, X_r)$ of random variables $X_1, \ldots, X_r \colon \Omega \longrightarrow \Sigma$ such that $P(X_i^{-1}s) = p_s$ for all $i = 1, \ldots, r$ and all $s \in \Sigma$.

Note. In particular the X_i are identically distributed. They are not necessarily independent.

The Coincidence Index of Message Sources

Definition. Let $Y = (Y_1, \ldots, Y_r)$ be another message of length r from a possibly different message source (Σ, q). Then the **coincidence index** of X and Y is the random variable

$$\mathsf{K}_{XY} \colon \Omega \longrightarrow \mathbb{R}$$

defined by

$$\mathsf{K}_{XY}(\omega) := \frac{1}{r} \cdot \#\{i = 1, \ldots, r \mid X_i(\omega) = Y_i(\omega)\} = \frac{1}{r} \cdot \sum_{i=1}^{r} \delta_{X_i(\omega), Y_i(\omega)}$$

We calculate its expectation under the assumption that each pair of X_i and Y_i is independent. From Lemma 4, using the additivity of E, we get

$$\mathrm{E}(\mathsf{K}_{XY}) = \frac{1}{r} \cdot \sum_{i=1}^{r} \mathrm{E}(\delta_{X_i,Y_i}) = \frac{1}{r} \cdot r \cdot \sum_{s \in \Sigma} p_s q_s = \sum_{s \in \Sigma} p_s q_s$$

independently of the length r. Therefore it is adequate to call this expectation the **coincidence index κ_{LM} of the two message sources** L, M. We have proven:

Theorem 2 *The coincidence index of two message sources $L = (\Sigma, p)$ and $M = (\Sigma, q)$ is*

$$\kappa_{LM} = \sum_{s \in \Sigma} p_s q_s$$

Now we are ready to calculate theoretical values for the "typical" coincidence indices of languages under the assumption that the model "message source" fits their real behaviour:

Example 1, random texts versus any language M: Here all $p_s = 1/n$, therefore $\kappa_{\Sigma*} = n \cdot \sum_{s\in\Sigma} 1/n \cdot q_s = 1/n$.

Example 2, English texts versus English: From Table 39 we get the value 0.0653.

Example 3, German texts versus German: The table gives 0.0758.

Example 4, English versus German: The table gives 0.0664.

Note that these theoretical values for the real languages differ slightly from the former empirical values. This is due to two facts:

- The model—as every mathematical model—is an approximation to the truth.
- The empirical values underly statistical variations and depend on the kind of texts that were evaluated.

The Cross-Product Sum of Message Sources

For a message $X = (X_1, \ldots, X_r)$ from a message source (Σ, p) we define the (relative) letter frequencies as random variables

$$\mathrm{M}_{sX}\colon \Omega \longrightarrow \mathbb{R}, \quad \mathrm{M}_{sX} = \frac{1}{r} \cdot \sum_{i=1}^{r} \delta_{sX_i},$$

or more explicitly,

$$\mathrm{M}_{sX}(\omega) = \frac{1}{r} \cdot \#\{i \mid X_i(\omega) = s\} \quad \text{for all } \omega \in \Omega.$$

We immediately get the expectation

$$\mathrm{E}(\mathrm{M}_{sX}) = \frac{1}{r} \cdot \sum_{i=1}^{r} \mathrm{E}(\delta_{sX_i}) = p_s.$$

Definition. Let $X = (X_1, \ldots, X_r)$ be a message from the source (Σ, p), and $Y = (Y_1, \ldots, Y_t)$, a message from the source (Σ, q). Then the **cross-product sum** of X and Y is the random variable

$$\mathsf{X}_{XY}\colon \Omega \longrightarrow \mathbb{R}, \quad \mathsf{X}_{XY} := \frac{1}{rt} \cdot \sum_{s\in\Sigma} \mathrm{M}_{sX}\mathrm{M}_{sY}.$$

Table 39: Calculating theoretical values for coincidence indices

Letter s	English p_s	German q_s	Square p_s^2	Square q_s^2	Product $p_s q_s$
A	0.082	0.064	0.006724	0.004096	0.005248
B	0.015	0.019	0.000225	0.000361	0.000285
C	0.028	0.027	0.000784	0.000729	0.000756
D	0.043	0.048	0.001849	0.002304	0.002064
E	0.126	0.175	0.015876	0.030625	0.022050
F	0.022	0.017	0.000484	0.000289	0.000374
G	0.020	0.031	0.000400	0.000961	0.000620
H	0.061	0.042	0.003721	0.001764	0.002562
I	0.070	0.077	0.004900	0.005929	0.005390
J	0.002	0.003	0.000004	0.000009	0.000006
K	0.008	0.015	0.000064	0.000225	0.000120
L	0.040	0.035	0.001600	0.001225	0.001400
M	0.024	0.026	0.000576	0.000676	0.000624
N	0.067	0.098	0.004489	0.009604	0.006566
O	0.075	0.030	0.005625	0.000900	0.002250
P	0.019	0.010	0.000361	0.000100	0.000190
Q	0.001	0.001	0.000001	0.000001	0.000001
R	0.060	0.075	0.003600	0.005625	0.004500
S	0.063	0.068	0.003969	0.004624	0.004284
T	0.091	0.060	0.008281	0.003600	0.005460
U	0.028	0.042	0.000784	0.001764	0.001176
V	0.010	0.009	0.000100	0.000081	0.000090
W	0.023	0.015	0.000529	0.000225	0.000345
X	0.001	0.001	0.000001	0.000001	0.000001
Y	0.020	0.001	0.000400	0.000001	0.000020
Z	0.001	0.011	0.000001	0.000121	0.000011
Sum	1.000	1.000	0.0653	0.0758	0.0664

To calculate its expectation we assume that each X_i is independent of all Y_j, and each Y_j is independent of all X_i. Under this assumption let us call the messages X and Y **independent**. Then from Lemma 4 and the formula

$$\mathsf{X}_{XY} := \frac{1}{rt} \cdot \sum_{s\in\Sigma} \sum_{i=1}^{r} \sum_{j=1}^{t} \delta_{sX_i}\delta_{sY_j}$$

we get

$$\mathrm{E}(\mathsf{X}_{XY}) = \frac{1}{rt} \cdot \sum_{s\in\Sigma} \sum_{i=1}^{r} \sum_{j=1}^{t} \mathrm{E}(\delta_{sX_i})\mathrm{E}(\delta_{sY_j}) = \sum_{s\in\Sigma} p_s q_s$$

again independently of the length r. Therefore we call this expectation the **cross-product sum χ_{LM} of the two message sources** L, M. We have proven:

Theorem 3 *The cross-product sum of two message sources $L = (\Sigma, p)$ and $M = (\Sigma, q)$ is*

$$\chi_{LM} = \sum_{s\in\Sigma} p_s q_s.$$

The Inner Coincidence Index of a Message Source

Let $X = (X_1, \ldots, X_r)$ be a message from a source (Σ, p). In analogy with Sections 10 and 14 we define the random variables

$$\Psi_X, \Phi_X \colon \Omega \longrightarrow \mathbb{R}$$

by the formulas

$$\Psi_X := \sum_{s\in\Sigma} M_{sX}^2, \qquad \Phi_X := \frac{r}{r-1} \cdot \Psi_x - \frac{1}{r-1}.$$

We try to calculate the expectation of Ψ_X first:

$$\begin{aligned}
\Psi_X &= \frac{1}{r^2} \cdot \sum_{s\in\Sigma} \left(\sum_{i=1}^{r} \delta_{sX_i} \right)^2 \\
&= \frac{1}{r^2} \cdot \sum_{s\in\Sigma} \left(\sum_{i=1}^{r} \delta_{sX_i} + \sum_{i=1}^{r} \sum_{j\neq i} \delta_{sX_i}\delta_{sX_j} \right)
\end{aligned}$$

since $\delta_{sX_i}^2 = \delta_{sX_i}$. Taking the expectation value we observe that for a sensible result we need the assumption that X_i and X_j are *independent* for $i \neq j$.

> In the language of MARKOV chains this means that we assume a MARKOV chain of order 0: The single letters of the messages from the source are independent from each other.

Under this assumption we get

$$\begin{aligned}
\mathrm{E}(\Psi_X) &= \frac{1}{r^2}\cdot\sum_{s\in\Sigma}\left(\sum_{i=1}^{r} p_s + \sum_{i=1}^{r}\sum_{j\neq i}\mathrm{E}(\delta_{sX_i})\mathrm{E}(\delta_{sX_j})\right)\\
&= \frac{1}{r^2}\cdot\left(\sum_{i=1}^{r}\underbrace{\sum_{s\in\Sigma} p_s}_{1} + \sum_{s\in\Sigma} p_s^2\cdot\underbrace{\sum_{i=1}^{r}\sum_{j\neq i} 1}_{r\cdot(r-1)}\right)\\
&= \frac{1}{r} + \frac{r-1}{r}\cdot\sum_{s\in\Sigma} p_s^2.
\end{aligned}$$

For Φ_X the formula becomes a bit more elegant:

$$\mathrm{E}(\Phi_X) = \frac{r}{r-1}\cdot\left(\frac{r-1}{r}\cdot\sum_{s\in\Sigma} p_s^2 + \frac{1}{r}\right) - \frac{1}{r-1} = \sum_{s\in\Sigma} p_s^2.$$

Let us call this expectation $\mathrm{E}(\Phi_X)$ the **(inner) coincidence index** of the message source (Σ, p), and let us call (by abuse of language) the message source **of order** 0 if its output messages are MARKOV chains of order 0 only. (Note that for a mathematically correct definition we should have included the "transition probabilities" into our definition of message source.) Then we have proved

Theorem 4 *The coincidence index of a message source $L = (\Sigma, p)$ of order* 0 *is*

$$\varphi_L = \sum_{s\in\Sigma} p_s^2.$$

The assumption of order 0 is relevant for small text lengths and negligeable for large texts, because for "natural" languages dependencies between letters affect small distances only. Reconsidering the tables in Section 11 we note in fact that the values for texts of lengths 100 correspond to the theoretical values, whereas for texts of lengths 26 the values are suspiciously smaller. An explanation could be that repeated letters, such as `ee`, `oo`, `rr`, are relatively rare and contribute poorly to the number of coincidences. This affects the power of the φ-test in an unfriendly way.

On the other hand considering SINKOV's test for the period in Section 13 we note that the columns of a polyalphabetic ciphertext are decimated excerpts from natural texts where the dependencies between letters are irrelevant: The assumption of order 0 is justified for SINKOV's test.

17 Stochastic Languages

The stochastic model of language as a stationary Markov process easily led to useful theoretic results that fit well with empirical observations. On the other hand it is far from the computer scientific model that regards a language as a fixed set of strings with certain properties and that is intuitively much closer to reality. In fact the Markov model may produce *every* string in Σ^* with a non-zero probability! (We assume that each letter $s \in \Sigma$ has a non-zero probability—otherwise we would throw it away.) Experience tells us that only a very small portion of all character strings represent meaningful texts in any natural language. Here we consider an alternative model that respects this facet of reality, but otherwise is somewhat cumbersome.

Recall from Chapter 1 that a language is a subset $M \subseteq \Sigma^*$.

A Computer Theoretic Model

The statistical cryptanalysis of the monoalphabetic substitution relied on the hypothesis—supported by empirical evidence—that the average relative frequencies of the letters $s \in \Sigma$ in texts of sufficient length from this language approximate typical values p_s. This is even true when we consider only fixed positions j in the texts, at least for almost all j—the first letters of texts for example usually have different frequencies.

Now we try to build a mathematical model of language that reflects this behaviour. Let $M \subseteq \Sigma^*$ a language, and $M_r := M \cap \Sigma^r$ for $r \in \mathbb{N}$ the set of texts of length r. The average frequency of the letter $s \in \Sigma$ at the position $j \in [0 \ldots r-1]$ of texts in M_r is

$$\mu_{sj}^{(r)} := \frac{1}{\# M_r} \cdot \sum_{a \in M_r} \delta_{sa_j}$$

(This sum counts the texts $a \in M_r$ with the letter s at position j.)

Example Let $M = \Sigma^*$ Then

$$\mu_{sj}^{(r)} := \frac{1}{n^r} \cdot \sum_{a \in \Sigma^r} \delta_{sa_j} = \frac{1}{n} \quad \text{for all } s \in \Sigma,\ j = 1, \ldots, r-1,$$

because there are exactly n^{r-1} possible texts with fixed $a_j = s$.

Definition

The language $M \subseteq \Sigma^*$ is called **stochastic** if there is at most a finite exceptional set $J \subseteq \mathbb{N}$ of positions such that

$$p_s := \lim_{r \to \infty} \mu_{sj}^{(r)}$$

exists uniformly in j and is independent from j for all $j \in \mathbb{N} - J$ and all $s \in \Sigma$.

The p_s are called the **letter frequencies** of M and obviously coincide with the limit values for the frequencies of the letters over the complete texts.

Examples and Remarks

1. The exceptional set J for natural languages usually consists only of the start position 0 and the end position. That is, the first and last letters of texts may have different frequencies. For example in English the letter "t" is the most frequent first letter instead of "e", followed by "a" and "o". In German this is "d", followed by "w", whereas "t" almost never occurs as first letter.
2. The language $M = \Sigma^*$ is stochastic.
3. Because always $\sum_{s\in\Sigma} \mu_{sj}^{(r)} = 1$, also $\sum_{s\in\Sigma} p_s = 1$.

Note that this notation is not standard in the literature.

Also note that we consider a *theoretical model*. For a natural language it may not be well-defined whether a given text is meaningful or not, not even if it is taken from a newspaper.

The Mean Coincidence Between Two Languages

Let $L, M \subseteq \Sigma^*$ two stochastic languages with letter frequencies q_s and p_s for $s \in \Sigma$. We consider the mean value of the coincidences of texts of length r:

$$\kappa_{LM}^{(r)} := \frac{1}{\#L_r} \cdot \frac{1}{\#M_r} \cdot \sum_{a\in L_r} \sum_{b\in M_r} \kappa(a,b)$$

Theorem 5 *The mean coincidence of the stochastic languages L and M is asymptotically*

$$\lim_{r\to\infty} \kappa_{LM}^{(r)} = \sum_{s\in\Sigma} p_s q_s$$

The proof follows.

Interpretation: The coincidence of sufficiently long texts of the same length is approximately

$$\kappa(a,b) \approx \sum_{s\in\Sigma} p_s q_s$$

An Auxiliary Result

Lemma 5 *Let M be a stochastic language. Then the average deviation for all letters $s \in \Sigma$*

$$\frac{1}{r} \cdot \sum_{j=0}^{r-1} \left(\mu_{sj}^{(r)} - p_s \right) \to 0 \quad \text{for } r \to \infty$$

Proof. Fix $\varepsilon > 0$, and let r large enough that

1. $r \geq 4 \cdot \frac{\#J}{\varepsilon}$,
2. $|\mu_{sj}^{(r)} - p_s| < \frac{\varepsilon}{2}$ for all $j \in [0 \ldots r] - J$.

For $j \in J$ we have $|\mu_{sj}^{(r)} - p_s| \leq |\mu_{sj}^{(r)}| + |p_s| \leq 2$. Therefore

$$\frac{1}{r} \cdot \sum_{j=0}^{r-1} |\mu_{sj}^{(r)} - p_s| < \frac{1}{r} \cdot 2 \cdot \#J + \frac{r - \#J}{r} \cdot \frac{\varepsilon}{2} \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

$\diamond$

Remark The mean frequency of s in texts of length r is

$$\mu_s^{(r)} = \frac{1}{r} \cdot \sum_{j=0}^{r-1} \mu_{sj}^{(r)} = \frac{1}{r} \cdot \frac{1}{\#M_r} \cdot \sum_{a \in M_r} \delta_{sa_j}$$

For this we get the limit

Corollary 5 $\lim_{r \to \infty} \mu_s^{(r)} = p_s$

Proof of the Theorem

$$\begin{aligned}
\kappa_{LM}^{(r)} &= \frac{1}{\#L_r \cdot \#M_r} \cdot \sum_{a \in L_r} \sum_{b \in M_r} \left(\frac{1}{r} \cdot \sum_{j=0}^{r-1} \sum_{s \in \Sigma} \delta_{sa_j} \delta_{sb_j} \right) \\
&= \sum_{s \in \Sigma} \frac{1}{r} \cdot \sum_{j=0}^{r-1} \left[\frac{1}{\#L_r} \sum_{a \in L_r} \delta_{sa_j} \right] \cdot \left[\frac{1}{\#M_r} \sum_{b \in M_r} \delta_{sb_j} \right] \\
&= \sum_{s \in \Sigma} \frac{1}{r} \cdot \sum_{j=0}^{r-1} [q_s + \varepsilon_{sj}] \cdot [p_s + \eta_{sj}] \\
&= \sum_{s \in \Sigma} \left[p_s q_s + \frac{p_s}{r} \cdot \sum_{j=0}^{r-1} \varepsilon_{sj} + \frac{q_s}{r} \cdot \sum_{j=0}^{r-1} \eta_{sj} + \frac{1}{r} \cdot \sum_{j=0}^{r-1} \varepsilon_{sj} \eta_{sj} \right]
\end{aligned}$$

The second and third summands converge to 0 by the lemma. The fourth converges to 0 because $|\varepsilon_{sj} \eta_{sj}| \leq 1$. Therefore the sum converges to $\sum_{s \in \Sigma} p_s q_s$. $\diamond$

References

[1] Bauer F. L. *Decrypted Secrets; Methods and Maxims of Cryptology.* Berlin: Springer 1997.

[2] Deavours C. A., Kruh L. *Machine Cryptography and Modern Cryptanalysis.* Norwood: Artech House 1985.

[3] Friedman W. F. *The Riverbank Publications Volume 1.* (contains Publications No. 15, 16, 17, and 18) Laguna Hills: Aegean Park Press 1979.

[4] Ganesan R., Sherman A. T., *Statistical Techniques for Language Recognition: An Introduction and Guide for Cryptanalysts.* Cryptologia 17 (1993), 321–366.

[5] Ganesan R., Sherman A. T., *Statistical Techniques for Language Recognition: An Empirical Study Using Real and Simulated English.* Cryptologia 18 (1994), 289–331.

[6] Gleason A. M. *Elementary Course in Probability for the Cryptanalyst.* Laguna Hills: Aegean Park Press 1985.

[7] Kullback S. *Statistical Methods in Cryptanalysis.* Laguna Hills: Aegean Park Press 1976.

[8] Sinkov A. *Elementary Cryptanalysis.* Washington: The Mathematical Association of America 1966.