Artificial Intelligence/ Language Processing

C. Montgomery Editor

Breaking Substitution Ciphers Using a Relaxation Algorithm

Shmuel Peleg and Azriel Rosenfeld
University of Maryland

Substitution ciphers are codes in which each letter of the alphabet has one fixed substitute, and the word divisions do not change. In this paper the problem of breaking substitution ciphers is represented as a probabilistic labeling problem. Every code letter is assigned probabilities of representing plaintext letters. These probabilities are updated in parallel for all code letters, using joint letter probabilities. Iterating the updating scheme results in improved estimates that finally lead to breaking the cipher. The method is applied successfully to two examples.

Key Words and Phrases: cryptography, substitution ciphers, probabilistic classification, relaxation

CR Categories: 3.42, 3.63

1. Introduction

Let $\sum$ be an alphabet consisting of letters and a space symbol. A substitution cipher is a permutation in which every letter of the alphabet in the message $M = m_1 \dots m_l$ except for the space symbol is replaced consistently by another letter to give the coded message $C = c_1 \dots c_l$. A key for a substitution cipher is a transformation $K{:}\sum \rightarrow \sum$ such that $M = K(c_1) \dots K(c_l)$. Manual methods for breaking such codes are well-known [1, 7], and are based primarily on the relative frequencies of single letters, and on certain combinations of letters in certain positions.

Authors' current address: S. Peleg and A. Rosenfeld, Computer Science Center, University of Maryland, College Park, MD 20742.

In this paper, a completely automatic method for breaking substitution ciphers is presented, based on relaxation methods. Relaxation algorithms have recently been introduced in image processing [4, 6]. They are iterative parallel classification algorithms, where every element in a graph structure tries to estimate its class membership probabilities based on those of its neighbors. The process is iterated until a satisfactory classification is achieved. A new formulation of relaxation [4, 5], based on probability theory, paves the way for more general applications of relaxation. The use of relaxation for domains other than image classification is demonstrated in this paper.

Section 2 describes the relaxation approach to probabilistic graph labeling; Section 4 discusses the application of this approach to substitution ciphers; and Section 5 summarizes the results obtained.

2. Probability Updating

The problem of probabilistic graph labeling is as follows:

Let $G = (V, E)$ be a graph with $V = \{v_1, \dots, v_n\}$ the set of nodes and E the set of arcs, and let $\Lambda = (\lambda_1, \dots, \lambda_L)$ be a set of labels. With every node $v_i \in V$ a random variable l_i is associated, specifying the probabilities of the possible labels for that node. Initially, based on some measurements, a probability distribution $P_i^{(0)}: \Lambda \rightarrow [0, 1]$ is estimated for every random variable l_i. In this section the updating of these distributions is discussed, based on the distributions at neighboring nodes, and on statistical relations among the random variables l_i.

Given a graph $G = (V, E)$, a set of labels Λ, and an estimate for a discrete probability distribution $P_i : \Lambda \rightarrow [0, 1]$ for each random variable l_i, new estimated probability distributions for l_i are to be computed.

We can regard the probability estimate vectors P_i as events, i.e., we can think of them as being chosen from a space of possible vectors. We shall now consider various prior and conditional probabilities involving these P_i events and the outcomes of the random variables l_i. In particular, we shall consider probabilities of the form

(1) $\text{Prob}(l_i = \alpha | P_i)$; this is just the probability that node v_i has label α, given that its estimated probability distribution of labels is P_i. We denote this probability by $P_i(l_i = \alpha)$.

(2) $\text{Prob}(P_i | l_i = \alpha)$; this is the probability that the distribution estimate for node v_i is P_i, given that the true label of v_i is α.

Evidently we have $P_i(l_i = \alpha) = \text{Prob}(P_i | l_i = \alpha) \cdot \text{Prob}(l_i = \alpha) / \text{Prob}(P_i)$. (The problem of actually calculating $\text{Prob}(P_i)$ will not be considered yet.)

Let v_i, v_j, and v_k be three nodes such that v_j and v_k are neighbors of v_i. We can consider the joint events $(l_i = \alpha, l_j = \beta, l_k = \gamma)$ and (P_i, P_j, P_k), and write

$$\begin{aligned}&\text{Prob}(l_i = \alpha, l_j = \beta, l_k = \gamma | P_i, P_j, P_k)\\ &\equiv P_{ijk}(l_i = \alpha, l_j = \beta, l_k = \gamma)\\ &= \frac{\text{Prob}(P_i, P_j, P_k | l_i = \alpha, l_j = \beta, l_k = \gamma)\cdot\text{Prob}(l_i = \alpha, l_j = \beta, l_k = \gamma)}{\text{Prob}(P_i, P_j, P_k)}\end{aligned} \tag{1}$$

We now assume that

$$\begin{aligned}&\text{Prob}(P_i, P_j, P_k | l_i = \alpha, l_j = \beta, l_k = \gamma)\\ &= \text{Prob}(P_i | l_i = \alpha)\,\text{Prob}(P_j | l_j = \beta)\,\text{Prob}(P_k | l_k = \gamma)\\ &= \frac{\text{Prob}(l_i = \alpha | P_i)\text{Prob}(P_i)}{\text{Prob}(l_i = \alpha)} \cdot \frac{\text{Prob}(l_j = \beta | P_j)\text{Prob}(P_j)}{\text{Prob}(l_j = \beta)} \cdot \frac{\text{Prob}(l_k = \gamma | P_k)\text{Prob}(P_k)}{\text{Prob}(l_k = \gamma)}\\ &= \frac{P_i(l_i = \alpha)P_j(l_j = \beta)P_k(l_k = \gamma)\,\text{Prob}(P_i)\,\text{Prob}(P_j)\,\text{Prob}(P_k)}{\text{Prob}(l_i = \alpha)\,\text{Prob}(l_j = \beta)\,\text{Prob}(l_k = \gamma)}\end{aligned}$$

The meaning of this assumption is that the probability estimate P_i is directly dependent only on l_i, and once l_i is given, the probability of the estimate being P_i is independent of P_j, $j \neq i$, and of l_j, $j \neq i$. Under this assumption, (1) becomes

$$\begin{aligned}&P_{ijk}(l_i = \alpha, l_j = \beta, l_k = \gamma)\\ &= P_i(l_i = \alpha)\,P_j(l_j = \beta)\,P_k(l_k = \gamma)\\ &\cdot \frac{\text{Prob}(l_i = \alpha, l_j = \beta, l_k = \gamma)}{\text{Prob}(l_i = \alpha)\,\text{Prob}(l_j = \beta)\,\text{Prob}(l_k = \gamma)}\\ &\cdot \frac{\text{Prob}(P_i)\,\text{Prob}(P_j)\,\text{Prob}(P_k)}{\text{Prob}(P_i, P_j, P_k)}\end{aligned} \tag{2}$$

From now on, we will denote

$$\frac{\text{Prob}(l_i = \alpha, l_j = \beta, l_k = \gamma)}{\text{Prob}(l_i = \alpha)\,\text{Prob}(l_j = \beta)\,\text{Prob}(l_k = \gamma)}$$

by $r_{ijk}(\alpha, \beta, \gamma)$. The quantities $r_{ijk}(\alpha, \beta, \gamma)$ are independent of the estimated distributions P_i. These quantities are computed in advance by using some model to find the required probabilities. See [3] for an approach to computing these coefficients in the absence of such models. Now

$$P_{ijk}(l_i = \alpha) = \sum_{\beta\in\Lambda}\sum_{\gamma\in\Lambda} P_{ijk}(l_i = \alpha, l_j = \beta, l_k = \gamma),$$

and

$$\sum_{\alpha\in\Lambda}\sum_{\beta\in\Lambda}\sum_{\gamma\in\Lambda} P_{ijk}(l_i = \alpha, l_j = \beta, l_k = \gamma) = 1.$$

We thus have

$$\begin{aligned}&P_{ijk}(l_i = \alpha)\\ &= \frac{\sum_{\beta\in\Lambda}\sum_{\gamma\in\Lambda} P_{ijk}(l_i = \alpha, l_j = \beta, l_k = \gamma)}{\sum_{\lambda\in\Lambda}\sum_{\beta\in\Lambda}\sum_{\gamma\in\Lambda} P_{ijk}(l_i = \lambda, l_j = \beta, l_k = \gamma)}\\ &= \frac{P_i(l_i = \alpha)\cdot\sum_{\beta\in\Lambda}\sum_{\gamma\in\Lambda} P_j(l_j = \beta)\,P_k(l_k = \gamma)r_{ijk}(\alpha, \beta, \gamma)}{\sum_{\lambda\in\Lambda} P_i(l_i = \lambda)\cdot\sum_{\beta\in\Lambda}\sum_{\gamma\in\Lambda} P_j(l_j = \beta)\,P_k(l_k = \gamma)r_{ijk}(\lambda, \beta, \gamma)},\end{aligned} \tag{3}$$

since the factor

$$\frac{\text{Prob}(P_i)\,\text{Prob}(P_j)\,\text{Prob}(P_k)}{\text{Prob}(P_i, P_j, P_k)}$$

cancels out.

Note that $P_{ijk}(l_i = \alpha)$ is dependent on the nodes v_i, v_j, and v_k. In a similar approach, rules for any number of nodes can be derived. A rule using two nodes, rather than three as is done here, can be found in [4].

Given all the possible pairs of neighbors, all the estimates suggested by all the pairs are to be combined into one estimate. Two methods can be used.

The first method takes the average of all the estimates given by all pairs of neighbors. In this case, the iterative updating expression is

$$P_i^{(n+1)}(l_i = \alpha) = \text{Average}\{P_{ijk}^{(n+1)}(l_i = \alpha)\} \tag{4}$$

where $P_{ijk}^{(n+1)}(l_i = \alpha)$ is computed from the $P_i^{(n)}$'s by expression (3), and the average is taken over all possible pairs of neighbors.

The second method, developed in [2], considers the updating as done pair after pair. Define $Q_{ijk}^{(n)}(\alpha)$ to be

$$Q_{ijk}^{(n)}(\alpha) = \sum_{\beta\in\Lambda}\sum_{\gamma\in\Lambda} P_j^n(l_j = \beta)P_k^n(l_k = \gamma)r_{ijk}(\alpha, \beta, \gamma).$$

Using $Q_{ijk}^{(n)}(\alpha)$ changes (3) into

$$P_{ijk}^{(n+1)}(l_i = \alpha) = \frac{P_i^{(n)}(l_i = \alpha)\cdot Q_{ijk}^{(n)}(\alpha)}{\sum_{\beta\in\Lambda} P_i^{(n)}(l_i = \beta)Q_{ijk}^{(n)}(\beta)}$$

As was shown in [2], the estimate for the total effect of all pairs of labels is

$$P_i^{(n+1)}(l_i = \alpha) = \frac{P_i^{(n)}(l_i = \alpha)\cdot\prod_{j,k} Q_{ijk}^{(n)}(\alpha)}{\sum_{\beta\in\Lambda} P_i^{(n)}(l_i = \beta)\prod_{j,k} Q_{ijk}^{(n)}(\beta)} \tag{5}$$

where j, k varies over all possible pairs of neighbors for node v_i.

3. The Coefficients

This paper handles problems from the domain of English text, and a node in the graph will represent a letter of the coded message. The relations that will be used are the probabilities of certain combinations of letters appearing in English plaintext. Only trigrams will be used; the event $(l_i = \alpha, l_j = \beta, l_k = \gamma)$ means that nodes v_i, v_j, and v_k represent the sequence $\alpha\beta\gamma$. The *a priori* probability $\text{Prob}(l_i = \alpha, l_j = \beta, l_k = \gamma)$ is equal to the probability that a randomly chosen trigram from an English text is $\alpha\beta\gamma$, and $\text{Prob}(l_i = \alpha)$ is equal to the probability that a randomly chosen letter from English text will be α. The above probabilities can be estimated from a long sample of English text. They were computed from the novel *Wuthering Heights* by Emily Bronte, which contains approximately one million letters. The coefficients $r_{ijk}(\alpha, \beta, \gamma)$ used in the updating rule are then

$$r_{ijk}(\alpha, \beta, \gamma) = \frac{\text{Prob}(\alpha\beta\gamma)}{\text{Prob}(\alpha)\ \text{Prob}(\beta)\ \text{Prob}(\gamma)}$$

where v_i represents a letter preceding v_j, v_k represents a letter following v_j, Prob($\alpha\beta\gamma$) is the *a priori* probability of a randomly chosen trigram being $\alpha\beta\gamma$, and Prob(α) is the probability of a randomly chosen letter being α.

4. Decoding

Let $\sum$ be the English alphabet together with a space symbol. A key K for a coded message $C = c_1 \dots c_l$ is the transformation $K{:}\sum \rightarrow \sum$ such that $M = K(c_1) \dots K(c_l)$ is the original message. In this section the key will be obtained from the coded message in two steps. Every node v_α in the graph will represent a letter α in the coded message. At the first step initial probabilities are assigned to every letter for every node. $P_\alpha^{(0)}(\beta)$ is the initial probability that $K(\alpha)$ will be β. The second step involves iterative application of relaxation to the probabilistically labeled graph to obtain a less ambiguous labeling. For every iteration, a key $K^{(n)}$ is constructed from $P^{(n)}$ such that $K^{(n)}(\alpha) = \beta$ if $P_\alpha^{(n)}(\beta)$ is the maximal element in $P_\alpha^{(n)}$. It will be seen that the number of elements in $K^{(n)}$ which agree with K increases with the number of iterations performed.

4.1 Initial Probabilities

First order statistics are used to obtain the initial probabilistic labeling. Let $f_e(\beta)$ be the relative frequency of the letter β in English. An initial probabilistic labeling can be found by using a multinomial model for the appearance of letters in a text [2]. In such a model, every letter in a text has the probability $f_e(\beta)$ of being the English letter β. For every such β, we can consider the binomial event of a letter being β or being something else, using $f_e(\beta)$ as the parameter of that binomial distribution. Given a coded message of length n, let the code letter α appear k times with relative frequency $f_c(\alpha) = k/n$. Using the binomial model, we can estimate the probability of the code letter α being the code for the English letter β (or the probability of $K(\alpha) = \beta$).

Using Bayes' rule, we have

$$\begin{aligned}&\text{Prob}(K(\alpha) = \beta \,|\, f_c(\alpha) = k/n)\\ &= \frac{\text{Prob}(f_c(\alpha) = k/n \,|\, K(\alpha) = \beta)\cdot\text{Prob}(K(\alpha) = \beta)}{\text{Prob}(f_c(\alpha) = k/n)}\end{aligned} \tag{6}$$

The following three expressions are used to compute (6):

$$\text{Prob}(K(\alpha) = \beta) = \frac{1}{|\sum|} \tag{7}$$

This means that all codes are equally likely.

$$\begin{aligned}&\text{Prob}(f_c(\alpha) = k/n)\\ &= \sum_{\beta\in\Sigma} \text{Prob}(f_c(\alpha) = k/n \,|\, K(\alpha) = \beta)\\ &\qquad\cdot\text{Prob}(K(\alpha) = \beta)\end{aligned} \tag{8}$$

This expression is an identity in probability theory.

$$\begin{aligned}&\text{Prob}(f_c(\alpha) = k/n \,|\, K(\alpha) = \beta)\\ &= \tbinom{n}{k}[f_e(\beta)]^k[1 - f_e(\beta)]^{n-k}\end{aligned} \tag{9}$$

This arises from the assumption of a binomial distribution with probability $f_e(\beta)$ for a letter being β.

By substituting (7), (8), and (9) in (6), we can derive a simplified expression

$$\begin{aligned}&\text{Prob}(K(\alpha) = \beta \,|\, f_c(\alpha) = k/n)\\ &= \frac{[f_e(\beta)]^k[1 - f_e(\beta)]^{n-k}}{\sum_{\lambda\in\Sigma} [f_e(\lambda)]^k[1 - f_e(\lambda)]^{n-k}}\end{aligned} \tag{10}$$

The values computed in (10) are used as initial probability estimates, and

$$P_\alpha^{(0)}(l_\alpha = \beta) = \text{Prob}(K(\alpha) = \beta \,|\, f_c(\alpha) = k/n). \tag{11}$$

It was found in the experiments that the particular way in which initial probabilities are assigned does not significantly change the results of the relaxation process. Even when an arbitrary expression like

$$P_\alpha^{(0)}(l_\alpha = \beta) \cong \left(1 - \frac{|f_c(\alpha) - f_e(\beta)|}{f_c(\alpha) + f_e(\beta)}\right)^4$$

(which assigns a probability of α being β in accordance with how close β's frequency in English is to α's frequency in the message) was used to compute the initial estimates, the results did not change significantly.

Note that no initial estimate is computed for the space symbol, since it is assumed that the space symbol is not changed by the code.

Table I. Relative frequencies of letters in the reference text, the technical report paragraph, and the Gettysburg Address.

Letters	English	Report	Gettysburg
A	0.078	0.072	0.089
B	0.014	0.008	0.012
C	0.024	0.036	0.027
D	0.048	0.046	0.050
E	0.129	0.117	0.144
F	0.022	0.025	0.023
G	0.021	0.035	0.024
H	0.066	0.031	0.070
I	0.072	0.086	0.059
J	0.001	0.001	0
K	0.008	0.007	0.003
L	0.040	0.032	0.037
M	0.025	0.027	0.011
N	0.072	0.089	0.067
O	0.075	0.071	0.081
P	0.016	0.029	0.013
Q	0.001	0.001	0.001
R	0.058	0.060	0.069
S	0.060	0.071	0.038
T	0.086	0.088	0.110
U	0.030	0.031	0.018
V	0.009	0.015	0.021
W	0.021	0.006	0.024
X	0.002	0.003	0
Y	0.022	0.008	0.009
Z	0.000	0.001	0

Table II. The Keys (the Substitutions with the Highest Probabilities) After Each Iteration, with the Corresponding Probabilities, for the Gettysburg Address, Using the Average Updating Scheme (12).

ITERATION →	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CODE LETTER																
A	T/ 52	T/ 41	A/ 54	A/ 75	A/ 86	A/ 92	A/ 95	A/ 97	A/ 98	A/ 98	A/ 99	A/ 99	'A/ 99	A/ 99	A/ 99	A/ 99
B	B/ 33	B/ 58	B/ 71	B/ 79	B/ 84	B/ 88	B/ 91	B/ 92	B/ 93	B/ 94	B/ 94	B/ 94	B/ 94	B/ 94	B/ 94	B/ 94
C	M/ 21	C/ 28	C/ 40	C/ 53	C/ 66	C/ 78	C/ 86	C/ 91	C/ 94	C/ 96	C/ 97	C/ 97	C/ 97	C/ 97	C/ 97	C/ 97
D	D/ 42	D/ 49	D/ 55	D/ 60	D/ 65	D/ 69	D/ 72	D/ 75	D/ 76	D/ 77	D/ 78	D/ 79	D/ 79	D/ 79	D/ 79	D/ 80
E	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100
F	C/ 16	W/ 22	W/ 30	W/ 39	W/ 44	W/ 45	W/ 42	F/ 45	F/ 52	F/ 57	F/ 62	F/ 66	F/ 69	F/ 72	F/ 74	F/ 76
G	C/ 17	G/ 17	G/ 29	G/ 50	G/ 70	G/ 82	G/ 88	G/ 91	G/ 94	G/ 95	G/ 96	G/ 97	G/ 98	G/ 98	G/ 98	G/ 99
H	I/ 19	H/ 35	H/ 62	H/ 86	H/ 97	H/100	H/100	H/100	H/100	H/100	H/100	H/100	H/100	H/100	H/100	H/100
I	S/ 29	R/ 31	R/ 24	I/ 38	I/ 56	I/ 70	I/ 82	I/ 89	I/ 94	I/ 96	I/ 98	I/ 99	I/ 99	I/100	I/100	I/100
J																
K	X/ 44	X/ 38	K/ 76	K/ 89	K/ 93	K/ 95	K/ 97	K/ 98	K/ 98	K/ 99	K/ 99	K/ 99	K/100	K/100	K/100	K/100
L	L/ 52	L/ 61	L/ 70	L/ 80	L/ 88	L/ 93	L/ 96	L/ 97	L/ 98	L/ 98	L/ 99	L/ 99	L/ 99	L/ 99	L/100	L/100
M	B/ 29	B/ 29	B/ 30	B/ 29	B/ 26	B/ 23	M/ 26	M/ 30	M/ 34	M/ 37	M/ 39	M/ 41	M/ 43	M/ 45	M/ 46	M/ 47
N	H/ 21	N/ 25	N/ 38	N/ 56	N/ 72	N/ 81	N/ 86	N/ 90	N/ 92	N/ 93	N/ 94	N/ 95	N/ 95	N/ 96	N/ 96	N/ 96
O	A/ 26	A/ 29	O/ 35	O/ 50	O/ 66	O/ 79	O/ 86	O/ 90	O/ 92	O/ 92	O/ 93	O/ 93	O/ 92	O/ 92	O/ 92	O/ 92
P	B/ 35	B/ 55	B/ 69	B/ 78	B/ 83	B/ 88	B/ 90	B/ 92	B/ 94	B/ 95	B/ 95	B/ 96	B/ 96	B/ 96	B/ 96	B/ 96
Q	Q/ 28	X/ 98	X/100	X/100	X/100	X/100	X/100	X/100	X/ 98	X/ 83	Q/ 83	Q/ 99	Q/100	Q/100	Q/100	Q/100
R	H/ 19	H/ 27	R/ 34	R/ 51	R/ 66	R/ 78	R/ 85	R/ 90	R/ 93	R/ 94	R/ 96	R/ 96	R/ 97	R/ 97	R/ 98	R/ 98
S	L/ 61	L/ 54	L/ 40	L/ 26	M/ 19	S/ 22	S/ 33	S/ 44	S/ 52	S/ 58	S/ 63	S/ 66	S/ 68	S/ 70	S/ 71	S/ 72
T	E/ 86	E/ 54	T/ 72	T/ 88	T/ 95	T/ 98	T/ 99	T/100	T/100	T/100	T/100	T/100	T/100	T/100	T/100	T/100
U	G/ 16	P/ 24	P/ 24	U/ 40	U/ 65	U/ 77	U/ 85	U/ 91	U/ 96	U/ 99	U/100	U/100	U/100	U/100	U/100	U/100
V	G/ 16	C/ 25	C/ 34	C/ 35	M/ 31	V/ 66	V/ 87	V/ 95	V/ 98	V/ 99	V/100	V/100	V/100	V/100	V/100	V/100
W	C/ 17	W/ 26	W/ 43	W/ 62	W/ 75	W/ 83	W/ 88	W/ 90	W/ 92	W/ 93	W/ 93	W/ 94	W/ 94	W/ 94	W/ 94	W/ 94
X																
Y	V/ 42	K/ 79	K/ 79	K/ 61	Y/ 61	Y/ 80	Y/ 90	Y/ 94	Y/ 97	Y/ 98	Y/ 98	Y/ 99	Y/ 99	Y/ 99	Y/ 98	Y/ 98
Z																
CORRECT:	5	9	14	16	17	19	20	21	21	21	22	22	22	22	22	22

Table III. Same as Table II, But for the Technical Report Paragraph.

ITERATION: → (CODE LETTER)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	N/ 19	I/ 26	I/ 31	A/ 36	A/ 43	A/ 49	A/ 55	A/ 61	A/ 67	A/ 73	A/ 77	A/ 81	A/ 84	A/ 87	A/ 89	A/ 90
B	K/ 43	K/ 40	B/ 50	B/ 69	B/ 79	B/ 83	B/ 83	B/ 81	B/ 77	B/ 71	B/ 66	B/ 62	B/ 59	B/ 58	B/ 57	B/ 57
C	L/ 45	L/ 47	L/ 45	L/ 37	C/ 41	C/ 55	C/ 65	C/ 70	C/ 76	C/ 81	C/ 83	C/ 84	C/ 85	C/ 85	C/ 85	C/ 85
D	D/ 45	D/ 45	D/ 46	D/ 51	D/ 61	D/ 74	D/ 83	D/ 89	D/ 92	D/ 94	D/ 95	D/ 95	D/ 96	D/ 96	D/ 97	D/ 97
E	E/ 99	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100
F	M/ 17	G/ 18	F/ 23	F/ 35	F/ 49	F/ 61	F/ 69	F/ 74	F/ 78	F/ 80	F/ 82	F/ 83	F/ 83	F/ 84	F/ 84	F/ 84
G	L/ 38	U/ 42	U/ 41	D/ 35	D/ 40	D/ 38	G/ 45	G/ 56	G/ 66	G/ 74	G/ 79	G/ 82	G/ 84	G/ 86	G/ 86	G/ 87
H	U/ 35	M/ 34	M/ 45	M/ 55	M/ 64	M/ 69	M/ 70	M/ 61	H/ 52	H/ 82	H/ 92	H/ 96	H/ 98	H/ 99	H/ 99	H/ 99
I	T/ 39	A/ 29	A/ 32	A/ 32	I/ 41	I/ 52	I/ 64	I/ 73	I/ 80	I/ 85	I/ 88	I/ 90	I/ 92	I/ 93	I/ 94	I/ 94
J	Q/ 27	J/ 75	J/ 91	J/ 98	J/100	J/100	J/100	J/100	J/100	J/100	J/100	J/100	J/100	J/100	J/100	J/100
K	K/ 49	K/ 68	K/ 77	K/ 82	K/ 86	K/ 90	K/ 94	K/ 96	K/ 98	K/ 99	K/ 99	K/100	K/100	K/100	K/100	K/100
L	U/ 38	L/ 26	L/ 36	L/ 46	L/ 55	L/ 63	L/ 70	L/ 76	L/ 79	L/ 82	L/ 84	L/ 85	L/ 85	L/ 85	L/ 85	L/ 85
M	M/ 20	M/ 22	M/ 25	M/ 30	M/ 37	M/ 46	M/ 53	M/ 59	M/ 63	M/ 68	M/ 72	M/ 76	M/ 79	M/ 83	M/ 85	M/ 87
N	T/ 49	T/ 43	N/ 42	N/ 74	N/ 91	N/ 97	N/ 99	N/100	N/100	N/100	N/100	N/100	N/100	N/100	N/100	N/100
O	I/ 19	O/ 30	O/ 44	O/ 58	O/ 68	O/ 75	O/ 79	O/ 81	O/ 81	O/ 81	O/ 80	O/ 80	O/ 80	O/ 80	O/ 81	O/ 81
P	U/ 27	C/ 24	C/ 30	C/ 35	C/ 38	C/ 42	C/ 44	C/ 46	C/ 47	C/ 46	C/ 45	C/ 42	C/ 39	C/ 35	P/ 32	P/ 40
Q	Q/ 27	Q/ 76	Q/ 94	Q/ 99	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100
R	S/ 25	R/ 40	R/ 57	R/ 72	R/ 84	R/ 91	R/ 95	R/ 97	R/ 98	R/ 99	R/ 99	R/ 99	R/100	R/100	R/100	R/100
S	I/ 19	N/ 28	N/ 31	N/ 30	R/ 28	S/ 32	S/ 38	S/ 45	S/ 51	S/ 57	S/ 62	S/ 65	S/ 67	S/ 68	S/ 69	S/ 70
T	T/ 46	T/ 41	T/ 39	T/ 40	T/ 46	T/ 55	T/ 62	T/ 73	T/ 87	T/ 95	T/ 97	T/ 99	T/ 99	T/ 99	T/100	T/100
U	U/ 35	U/ 44	U/ 56	U/ 71	U/ 87	U/ 95	U/ 98	U/ 99	U/100	U/100	U/100	U/100	U/100	U/100	U/100	U/100
V	P/ 26	P/ 26	V/ 44	V/ 69	V/ 86	V/ 95	V/ 98	V/ 99	V/100	V/100	V/100	V/100	V/100	V/100	V/100	V/100
W	K/ 53	K/ 58	K/ 54	K/ 44	B/ 50	B/ 58	B/ 59	B/ 57	B/ 54	B/ 47	W/ 40	W/ 55	W/ 65	W/ 72	W/ 77	W/ 81
X	X/ 41	X/ 85	X/ 97	X/100	X/100	X/100	X/100	X/100	X/100	X/100	X/100	X/100	X/100	X/100	X/100	X/100
Y	K/ 43	K/ 88	K/ 95	K/ 93	K/ 82	K/ 61	Y/ 61	Y/ 74	Y/ 81	Y/ 83	Y/ 84	Y/ 84	Y/ 84	Y/ 84	Y/ 84	Y/ 84
Z	Q/ 27	Z/ 85	Z/ 98	Z/ 99	Z/100	Z/100	Z/100	Z/100	Z/100	Z/100	Z/100	Z/100	Z/100	Z/100	Z/100	Z/100
CORRECT:	8	13	17	18	20	21	23	23	24	24	25	25	25	25	26	26

Table IV. Same as Table II, But Using the Product Updating Rule (13).

CODE LETTER / ITERATION: →	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	T/ 52	A/100	A/100	A/100	A/100	A/100	A/100	A/100	A/100	A/100	A/100	A/100	A/100	A/100	A/100	A/100
B	B/ 33	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100
C	M/ 21	C/ 98	C/100	C/100	C/100	C/100	C/100	C/100	C/100	C/100	C/100	C/100	C/100	C/100	C/100	C/100
D	D/ 42	S/ 80	D/100	D/100	D/100	D/100	D/100	D/100	D/100	D/100	D/100	D/100	D/100	D/100	D/100	D/100
E	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100
F	C/ 16	M/ 91	F/100	F/100	F/100	F/100	F/100	F/100	F/100	F/100	F/100	F/100	F/100	F/100	F/100	F/100
G	C/ 17	C/ 68	G/100	G/100	G/100	G/100	G/100	G/100	G/100	G/100	G/100	G/100	G/100	G/100	G/100	G/100
H	I/ 19	R/ 96	H/100	H/100	H/100	H/100	H/100	H/100	H/100	H/100	H/100	H/100	H/100	H/100	H/100	H/100
I	S/ 29	I/ 98	I/100	I/100	I/100	I/100	I/100	I/100	I/100	I/100	I/100	I/100	I/100	I/100	I/100	I/100
J																
K	X/ 44	K/100	K/100	K/100	K/100	K/100	K/100	K/100	K/100	K/100	K/100	K/100	K/100	K/100	K/100	K/100
L	L/ 52	L/100	L/100	L/100	L/100	L/100	L/100	L/100	L/100	L/100	L/100	L/100	L/100	L/100	L/100	L/100
M	B/ 29	P/ 72	M/100	M/100	M/100	M/100	M/100	M/100	M/100	M/100	M/100	M/100	M/100	M/100	M/100	M/100
N	H/ 21	N/100	N/100	N/100	N/100	N/100	N/100	N/100	N/100	N/100	N/100	N/100	N/100	N/100	N/100	N/100
O	A/ 26	O/100	O/100	O/100	O/100	O/100	O/100	O/100	O/100	O/100	O/100	O/100	O/100	O/100	O/100	O/100
P	B/ 35	B/ 98	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100
Q	Q/ 28	X/ 98	X/ 56	Q/ 97	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100
R	H/ 19	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100
S	L/ 61	S/ 81	S/100	S/100	S/100	S/100	S/100	S/100	S/100	S/100	S/100	S/100	S/100	S/100	S/100	S/100
T	E/ 86	T/100	T/100	T/100	T/100	T/100	T/100	T/100	T/100	T/100	T/100	T/100	T/100	T/100	T/100	T/100
U	G/ 16	U/100	U/100	U/100	U/100	U/100	U/100	U/100	U/100	U/100	U/100	U/100	U/100	U/100	U/100	U/100
V	G/ 16	C/ 42	V/100	V/100	V/100	V/100	V/100	V/100	V/100	V/100	V/100	V/100	V/100	V/100	V/100	V/100
W	C/ 17	W/ 95	W/100	W/100	W/100	W/100	W/100	W/100	W/100	W/100	W/100	W/100	W/100	W/100	W/100	W/100
X																
Y	V/ 42	Y/ 87	Y/100	Y/100	Y/100	Y/100	Y/100	Y/100	Y/100	Y/100	Y/100	Y/100	Y/100	Y/100	Y/100	Y/100
Z																
CORRECT:	5	15	21	22	22	22	22	22	22	22	22	22	22	22	22	22

Table V. Same as Table IV, But for the Technical Report Paragraph.

CODE LETTER \ ITERATION →	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	N/ 19	A/100	A/100	A/100	A/100	A/100	A/100	A/100	A/100	A/100	A/100	A/100	A/100	A/100	A/100	A/100
B	K/ 43	K/ 42	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100	B/100
C	L/ 45	C/ 55	C/100	C/100	C/100	C/100	C/100	C/100	C/100	C/100	C/100	C/100	C/100	C/100	C/100	C/100
D	D/ 45	S/ 69	S/ 98	S/100	S/ 62	D/100	D/100	D/100	D/100	D/100	D/100	D/100	D/100	D/100	D/100	D/100
E	E/ 99	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100	E/100
F	M/ 17	M/ 55	F/ 97	F/100	F/100	F/100	F/100	F/100	F/100	F/100	F/100	F/100	F/100	F/100	F/100	F/100
G	L/ 38	R/ 38	G/100	G/100	G/100	G/100	G/100	G/100	G/100	G/100	G/100	G/100	G/100	G/100	G/100	G/100
H	U/ 35	L/ 99	H/100	H/100	H/100	H/100	H/100	H/100	H/100	H/100	H/100	H/100	H/100	H/100	H/100	H/100
I	T/ 39	I/100	I/100	I/100	I/100	I/100	I/100	I/100	I/100	I/100	I/100	I/100	I/100	I/100	I/100	I/100
J	Q/ 27	J/ 75	J/100	J/100	J/100	J/100	J/100	J/100	J/100	J/100	J/100	J/100	J/100	J/100	J/100	J/100
K	K/ 49	K/ 99	K/100	K/100	K/100	K/100	K/100	K/100	K/100	K/100	K/100	K/100	K/100	K/100	K/100	K/100
L	U/ 38	L/100	L/100	L/100	L/100	L/100	L/100	L/100	L/100	L/100	L/100	L/100	L/100	L/100	L/100	L/100
M	M/ 20	M/ 99	R/ 98	R/ 99	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100
N	T/ 49	N/100	N/100	N/100	N/100	N/100	N/100	N/100	N/100	N/100	N/100	N/100	N/100	N/100	N/100	N/100
O	I/ 19	O/ 83	O/100	O/100	O/100	O/100	O/100	O/100	O/100	O/100	O/100	O/100	O/100	O/100	O/100	O/100
P	U/ 27	C/100	G/ 98	P/100	P/100	P/100	P/100	P/100	P/100	P/100	P/100	P/100	P/100	P/100	P/100	P/100
Q	Q/ 27	Q/ 76	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100	Q/100
R	S/ 25	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100	R/100
S	I/ 19	S/100	S/100	S/100	S/100	S/100	S/100	S/100	S/100	S/100	S/100	S/100	S/100	S/100	S/100	S/100
T	T/ 46	T/ 56	T/100	T/100	T/100	T/100	T/100	T/100	T/100	T/100	T/100	T/100	T/100	T/100	T/100	T/100
U	U/ 35	U/ 95	U/100	U/100	U/100	U/100	U/100	U/100	U/100	U/100	U/100	U/100	U/100	U/100	U/100	U/100
V	P/ 26	V/100	V/100	V/100	V/100	V/100	V/100	V/100	V/100	V/100	V/100	V/100	V/100	V/100	V/100	V/100
W	K/ 53	K/ 89	W/ 96	W/100	W/100	W/100	W/100	W/100	W/100	W/100	W/100	W/100	W/100	W/100	W/100	W/100
X	X/ 41	X/100	X/100	X/100	X/100	X/100	X/100	X/100	X/100	X/100	X/100	X/100	X/100	X/100	X/100	X/100
Y	K/ 43	K/ 99	K/100	K/100	K/100	K/100	K/100	K/ 99	Y/ 43	Y/100	Y/100	Y/100	Y/100	Y/100	Y/100	Y/100
Z	Q/ 27	Z/ 85	Z/ 98	Z/100	Z/100	Z/100	Z/100	Z/100	Z/100	Z/100	Z/100	Z/100	Z/100	Z/100	Z/100	Z/100
CORRECT:	8	18	22	23	23	24	24	24	25	25	25	25	25	25	25	25

4.2 Relaxation

The graph $G = (V, E)$ used consists of 27 nodes, $V = \{v_{\#}, v_a, \ldots, v_z\}$, where $v_{\#}$ represents the space symbol and v_α represents the code letter α for all letters that can appear in C. The "arcs" in the graph are triples of nodes, $E \subseteq V^3$. Every arc represents an occurrence of a trigram in C. The arcs are

$$E = \{(v_\alpha, v_\beta, v_\gamma) \mid \text{There is one occurrence of } \alpha\beta\gamma \text{ in } C\}.$$

An arc $(v_\alpha, v_\beta, v_\gamma)$ will appear in E as many times as the sequence $\alpha\beta\gamma$ appears in C.

Let E_λ be the set of all arcs passing through v_λ:

$$E_\lambda = \{(v_\alpha, v_\lambda, v_\beta) \mid (v_\alpha, v_\lambda, v_\beta) \in E.$$

Given the probability vectors $P^{(n)}$ at the nth iteration, the probabilities at the $(n + 1)$st iteration using (4) are

$$P_\lambda^{(n+1)}(l_\lambda = \omega) = \frac{1}{|E_\lambda|} \sum_{(v_\alpha, v_\lambda, v_\beta) \in E_\lambda} P_{\lambda\alpha\beta}^{(n+1)}(l_\lambda = \omega) \qquad (12)$$

where the $P_{\lambda\alpha\beta}^{(n+1)}$ are computed from the $P^{(n)}$'s using (3). Since $P_{\lambda\alpha\beta}^{(n+1)}$ is the probability vector for v_λ as suggested by the sequence $\alpha\lambda\beta$, $P_\lambda^{(n+1)}$ is the average (vector) of all probability vectors suggested by all the occurrences of λ in C, together with the preceding and the succeeding letters.

We can alternatively use (5) to compute the $P^{(n+1)}$ vectors from the $P^{(n)}$ vectors. Let $Q_{\lambda\alpha\beta}^{(n)}(\omega)$ be

$$Q_{\lambda\alpha\beta}^{(n)}(\omega) = \sum_{\delta \in \Sigma} \sum_{\epsilon \in \Sigma} P_\alpha(l_\alpha = \delta) P_\beta(l_\beta = \epsilon) r_{\alpha\lambda\beta}(\delta, \omega, \epsilon)$$

where $(v_\alpha, v_\lambda, v_\beta) \in E_\lambda$. $Q_{\lambda\alpha\beta}^{(n)}(\omega)$ can be interpreted as the support that interpretation ω in node v_λ gets from the probabilistic labeling at a predecessor v_α of v_λ and a successor v_β of v_λ. From the Q's a new estimate is computed analogous to (5):

$$P_\lambda^{(n+1)}(l_\lambda = \omega) = \frac{P_\lambda^{(n)}(l_\lambda = \omega) \cdot \prod_{(\alpha, \lambda, \beta) \in E_\lambda} Q_{\lambda\alpha\beta}^{(n)}(\omega)}{\sum_\delta P_\lambda^{(n)}(l_\lambda = \delta) \cdot \prod_{(\alpha, \lambda, \beta) \in E_\lambda} Q_{\lambda\alpha\beta}^{(n)}(\delta)} \qquad (13)$$

5. Examples

As examples we use two short passages. One is a paragraph taken from a recent technical report (996 characters), and the other is Lincoln's Gettysburg Address (1149 characters). The frequencies of letters in our reference text and the two messages are given in Table I. The messages were coded using the identity substitution, so that the coded message is identical to the original message. The key in this case is, of course, the identity transformation where $K(\alpha) = \alpha$ for every $\alpha \in \Sigma$. Using such an encoding makes no sense for a human cryptoanalyst, since every person familiar with the language can immediately recognize the message, and there is no need for decoding. But for the computer system, which does not know English, finding the identity key is just as difficult as finding any other key.

Tables II and III summarize the maximum-probability substitution $K^{(n)}$ for each iteration, together with the corresponding probabilities, for the two examples using the average rule (12). The usage of one product rule (13) for the two examples is shown in Tables IV and V.

It can be observed from the examples that the multiplicative updating rule (13) converges faster than the average updating rule (12). But for the technical passage this speed caused a wrong classification of one letter, that was correctly classified by the "slow" averaging scheme.

6. Concluding Remarks

This paper has demonstrated the application of relaxation methods to the solution of substitution ciphers. We used ciphers in which blanks were left intact, but the method should work well even if another character were substituted for blank, since blank has significantly higher frequency than any other letter. Further experiments with messages where blanks were eliminated also resulted in a mostly correct key, but took more iterations to achieve.

The results illustrate how relaxation methods can be useful in solving a variety of probabilistic graph labeling problems. The application of relaxation to the problem of ambiguous segmentation of handwritten words with uncertain interpretation can be found in [5].

Acknowledgments. The support of the National Science Foundation under Grant MCS-76-23763 is gratefully acknowledged, as is the help of Mrs. D. Shifflett in preparing this paper. The authors also wish to thank R. Kirby for many helpful suggestions.

Received January 1979; revised September 1979

References

1. Gaines, H.F. *Cryptanalysis.* Dover, New York, 1956.
2. Kirby, R. A product rule relaxation method. To appear in *Comptr. Graphics and Image Processing.*
3. Peleg, S., and Rosenfeld, A. Determining compatibility coefficients for curve enhancement relaxation processes. *IEEE Trans. Systems, Man and Cybernetics SMC-8* (July 1978), 548–554.
4. Peleg, S. A new probabilistic relaxation scheme. IEEE Conf. on Pattern Recognition and Image Processing, Chicago, Aug. 1979, pp. 337–343.
5. Peleg, S. Ambiguity reduction in handwriting with ambiguous segmentation and uncertain interpretation. *Comptr. Graphics and Image Processing 10* (July 1979), 235–245.
6. Rosenfeld, A., Hummel, R.A., and Zucker, S.W. Scene labelling by relaxation operations. *IEEE Trans. Systems, Man and Cybernetics SMC-6* (June 1976), 420–433.
7. Sinkov, A. *Elementary Cryptanalysis: A Mathematical Approach.* Random House, New York, 1968.