AUTOMATED ATTACKS ON COMPRESSION-BASED CLASSIFIERS

by

IGOR BURAGO

A THESIS

Presented to the Department of Computer and Information Science
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Master of Science

June 2014

THESIS APPROVAL PAGE

Student: Igor Burago

Title: Automated Attacks on Compression-Based Classifiers

This thesis has been accepted and approved in partial fulfillment of the requirements for the Master of Science degree in the Department of Computer and Information Science by:

Daniel Lowd	Chair
Dejing Dou	Core Member
Christopher Wilson	Core Member

and

Kimberly Andrews Espy	Vice President for Research & Innovation/Dean of the Graduate School

Original approval signatures are on file with the University of Oregon Graduate School.

Degree awarded June 2014

THESIS ABSTRACT

Igor Burago

Master of Science

Department of Computer and Information Science

June 2014

Title: Automated Attacks on Compression-Based Classifiers

Methods of compression-based text classification have proven their usefulness for various applications. However, in some classification problems, such as spam filtering, a classifier confronts one or many adversaries willing to induce errors in the classifier's judgment on certain kinds of input. In this thesis, we consider the problem of finding thrifty strategies for character-based text modification that allow an adversary to revert classifier's verdict on a given family of input texts. We propose three statistical statements of the problem that can be used by an attacker to obtain transformation models which are optimal in some sense. Evaluating these three techniques on a realistic spam corpus, we find that an adversary can transform a spam message (detectable as such by an entropy-based text classifier) into a legitimate one by generating and appending, in some cases, as few additional characters as 20% of the original length of the message.

CURRICULUM VITAE

NAME OF AUTHOR: Igor Burago

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene, Oregon
Far Eastern Federal University, Vladivostok, Russia

DEGREES AWARDED:

Master of Science, Computer & Information Science, 2014, University of Oregon
Bachelor of Science, Applied Mathematics and Informatics, 2011, Far Eastern Federal University

AREAS OF SPECIAL INTEREST:

Artificial Intelligence
Machine Learning
Adversarial Machine Learning

PROFESSIONAL EXPERIENCE:

Part-time software developer and system administrator, Pacific Fisheries Research Center (TINRO-Center), Vladivostok, Russia, 2007–2009

Volunteer jury member of the Far-Eastern regional stages of individual and team All-Russian high school programming competitions, Far Eastern Federal University, Vladivostok, Russia, 2006–2011

Volunteer software developer, Department of Computer Science, Institute of Mathematics and Computer Science, Far Eastern Federal University, Vladivostok, Russia, 2008

GRANTS, AWARDS AND HONORS:

Graduate Research Fellowship, Information Services, 2012–2014

Summa cum Laude, Far Eastern Federal University, 2011

PICES Ocean Monitoring Service Award, North Pacific Marine Science Organization (along with Igor I. Shevchenko and Olga N. Vasik, TINRO-Center), 2009

Scholarship of the President of the Far Eastern Federal University, 2009

Scholarship of the Administration of Vladivostok, 2009

Scholarship of the Governor of Primorsky Krai, 2009

Scholarship of the President of the Russian Federation, 2009

Scholarship of the President of the Far Eastern Federal University, 2008

Scholarship of the Governor of Primorsky Krai, 2008

Award of the President of the Russian Federation for Support of Talented Youth, 2006

PUBLICATIONS:

Burago, I. V., Vasik, O. N., Moiseenko, G. S., & Shevchenko, I. I. (2013). Metadata Exchange as a Preliminary Step in Creating a Common Information Space. (In Russian). In *Proceedings of the industry seminar "Mathematical Modeling and Information Technology in the Research of Biological Resources of the World's Oceans,"* September 25–27, 2013 (pp. 53–55). Vladivostok, Russia: TINRO-Center.

Burago, I. V. & Shevchenko, I. I. (2010). Automatic Generation of Enumeration Problems. In *The First Russia and Pacific Conference on Computer Technology and Applications (RPC 2010)*, September 6–9, 2010. Vladivostok, Russia: IACP FEB RAS. ISBN: 978-0-9803267-3-4 (CD).

Burago, I. V. & Shevchenko, I. I. (2009). Automatic generation of problems using the method of constraint propagation. (In Russian). In *The XXXIV Academician E. V. Zolotov Far Eastern Mathematical School-Seminar, "Fundamental Problems of Mathematics and Information Sciences,"* June 25–30, 2009 (pp. 162–163). Khabarovsk, Russia: Pacific National University Press.

Burago, I. V., Vasik, O. N., Moiseenko, G. S., & Shevchenko I. I. (2007). An infrastructure for the Metadata Exchange of Ecosystem Observations. (In Russian). In *Proceedings of the industry seminar "Mathematical Modeling and Information Technology in the Research of Biological Resources of the World's Oceans,"* October 1–3, 2007 (pp. 22–24). Vladivostok, Russia: TINRO-Center.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Daniel Lowd, for the amount of time he dedicated to helpful disscussions on the subject of this research.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

Automatic text classification is a widespread problem occurring in different applications. An important example of such application originates from the necessity of filtering unsolicited or undesired messages in email or other messaging systems. One of the effective methods for filtering messages is based on the observation that different kinds of texts often have distinctive compression characteristics.

In this approach, messages are seen as finite chunks of characters originating from streams that have probabilistic nature. More strictly, each kind of text, or class, is assumed to be independently defined by a probability distribution that describes the chances of a character to appear at each of the states of the stream producing texts of this class. Then, if that class distributions are known or can be reconstructed with sufficient accuracy from an available sample, any text is classified by comparing its compression characteristics for each of the class distributions. One metric directly connected with compressibility is the entropy which is usually calculated per character to make strings of unequal lengths comparable. If, for the text in question, the entropy per character, under the assumption that this text originated from one source, estimates to be lower than the same entropy for another source, then it is assumed that this text is more likely to belong to the former class rather than the latter.

This way, the objective of a compression-based classifier resolves into statistical problem of learning unknown class models by observing messages coming on the input of classifier. In the thesis, we make an attempt to look at the classifier problem from the adversarial point of view. In abstract, the goal of an adversary consists in tricking classifier into making decisions advantageous for a certain adversarial objective, e.g.

finding a way to modify a spam message so that it would not be classified as such, and would be label as legible. Unsurprisingly, the adversarial problem is, in some sense, inverse to the classifier problem: While the goal of the classifier is to model class sources by analyzing the input stream of texts, the goal of an adversary is to affect the input stream of the classifier to skew the resulting class models. That is, what was the output for the classifier becomes input for the adversary, and vice versa.

We approach the adversary problem as a statistical one. That is, for the example of message filtering, we assume that the adversary does not have a goal of getting a beneficial verdict of the classifier once for a single message, but rather wants to find a whole family of messages that would be classified in a certain way (while, optionally, satisfying some additional conditions aligned with adversary's goals). In other words, the adversary is interested in methodically changing statistical properties of the classifier's input stream.

To validate the methods discussed in this work, we concentrate on a typical application of entropy-based text classification—the problem of spam filtering. It allows us to evaluate our algorithms on a sufficient amount of real data, while remaining in the smallest case of binary decisions. Considering potential interests of an adversary in this problem, we introduce three different adversary problem settings that meaningfully formalize objectives of a spammer. In doing so, we tried to keep the balance between generality of a mathematically stated objective and feasibility of its analysis. We evaluate effectiveness of each strategy on the SpamAssassin public corpus of legitimate and spam email messages (Apache SpamAssassin Project, 2005) that is used widely for this purpose.

The remainder of the thesis is structured as follows. Chapter II provides background on the problem of entropy-based classification and related work on some of the

corresponding adversarial problems for this type of classifiers. Chapter III introduces the formal definition of the adversary problem and describes our approach to solving it. Chapter IV describes the experimental results obtained from evaluation of our method on a public email corpus. Finally, Chapter V concludes.

CHAPTER II

BACKGROUND AND RELATED WORK

The starting point of our research is the problem of compression-based text classification. Fundamentally, it rests on the assumption that when a pair of texts compress well together and, consequently, share some structural homogeneity, it is more likely that they belong to the same category. This assumption forms the basis for the whole class of methods using measures of compressibility from various compression models as measures of text similarity. This approach has been studied in a number of works targeting particular models, specific algorithms for building them, and their efficiency for estimating similarity of different types of texts (Cormack & Horspool, 1987; Frank, Chui, & Witten, 2000; Goodman, Heckerman, & Rounthwaite, 2005; Bratko, Cormack, Filipič, Lynam, & Zupan, 2006; Bratko, Filipič, & Zupan, 2006).

The focus of our research is specifically on the entropy-based classifiers that define similarity measure to be the cross entropy. At the same time, our approach is mostly agnostic to particular choice of algorithm to be used for learning probability models of classes. For the purposes of evaluation, to estimate class probability distributions, we use the algorithm of prediction by partial matching (PPM) (Cleary & Witten, 1984; Moffat, 1990; Cleary & Teahan, 1997; Teahan, 1995), which has been shown to suit well the special case of text classification—spam filtering (Bratko, Cormack, et al., 2006; Bratko, Filipič, & Zupan, 2006).

While a significant amount of effort has been applied to study the effectiveness of compression models in selected applications of text classification, there is still no complete understanding of how robust such algorithms are to different kinds of adversarial noise.

In (Lowd & Meek, 2005b), it has been shown that a word-based attack is effective against a maximum entropy spam filter (and a naive Bayes one, as well). The authors propose the attack of inserting and appending freestanding words to spam messages that are detectable as such by the classifier. To guide the selection of words, they consider two sets of heuristics depending on whether the adversary is able to determine that a modified message has been filtered out (active attack) or not (passive attack). In the former case, the knowledge of classifier's decisions is used as a supervisory signal.

In principle, random words that, being included in a barely-spam message, make it legitimate, are promoted for future additions; the ones that make a barely-legitimate message spam, are impeded. In the case of passive attack, the authors use frequency-based heuristics requiring the availability of training samples. According to this strategy, the words with higher frequency of occurrence in legitimate messages, or, alternatively, with higher ratio of frequencies in legitimate messages over spam messages, are assigned with higher probability of being used for an attack. Although these strategies prove the feasibility of attacking maximum entropy classifiers, they are too simplistic. In both active and passive cases, the information about the message being modified is unnecessary for approaching the optimal transformation procedure; it is global due to the nature of the considered classifiers.

The work (Lowd & Meek, 2005a) expands the subject to devising an attack on a binary linear classifiers in the case of incomplete or lacking information about the parameters of probabilistic models of those classifiers. This time, however, the focus is not on the ways of constructing an attack, but rather on methods of retrieving sufficient knowledge about the classifier to make a construction possible. The authors consider the cases of Boolean and continuous features for a few adversarial cost functions estimating the cost, or penalty, for each instance sent by an adversary to the input of a classifier.

For both types of features, they propose algorithms that for a linear cost function is able to discover classifier's weights in polynomial number of queries to the classifier.

In (Biggio, Nelson, & Laskov, 2011, 2012), the question of robustness of another linear classifier, support vector machines, is researched. The authors propose a poisoning algorithm iteratively improving the attack point in attempt to optimize the classification error which, as they show in evaluation, can be made as large as about one third. The problem considered in these works, however, is less relevant to the one we are working with in this thesis. The key distinction consists in that the authors concentrate on a different kind of adversarial position, where an attacker is supposed to be able to inject prepared inputs into the classifier's training data, thus poisoning them. In contrast, we focus on the adversary's task of disguising their activity in the input of the classifier given a fixed training data which is assumed to be unchanged during the time of the interaction. Additionally, the obtained results significantly depend on properties of support vector machines, and cannot be directly applied to maximum entropy classifiers.

CHAPTER III

METHOD

3.1. Classifier Problem

3.1.1. Preliminaries

Let $X \subseteq A^*$ be a space of arbitrary text strings over some finite alphabet A. On this space, we consider *sources* or *classes* of strings that are defined by probability distributions over the set X. In particular, from now on, whenever we discuss a classification problem, we assume that there exists a single *input source* of strings from X that come on the input of the classifier.

The input source is described by the probability $g(x) \equiv P(\xi = x)$ assigned to values $x \in X$ of the discrete random variable ξ standing for the input strings. The *classifier* reconstructs the probability distributions $f^{(\kappa)}(x)$ corresponding to one or more classes κ. Formally, we define probability $f^{(\kappa)}(x) \equiv P(\xi = x \mid C^{(\kappa)})$ for $C^{(\kappa)}$ being the event $\{\xi$ belongs to class $\kappa\}$. In this work we concentrate on the case of two classes of strings: legitimate *Ham* messages and unsolicited *Spam* messages that are designated with $\kappa = \mathrm{H}$ and $\kappa = \mathrm{S}$, respectively.

3.1.2. Finite Memory Markov Model

The probability of a string $x \in X$ originating from the class κ is equal to

$$f^{(\kappa)}(x) = \prod_{l=1}^{|x|} P(x_l \mid x_1^{l-1}, \kappa), \tag{3.1}$$

where x_l denotes the l-th character of the string x, and x_k^l the substring of x starting from the k-th and ending with the l-th character (if $k > l$, x_k^l is empty). For the sake of brevity, $P(x_l \mid x_1^{l-1}, \kappa)$ stands for the probability $P(\xi_l = x_l \mid \xi_1^{l-1} = x_1^{l-1}, C^{(\kappa)})$ of character x_l following the *context* x_1^{l-1}.

Naturally, we can parametrize distributions $f^{(\kappa)}(x)$ using these probabilities:

$$f^{(\kappa)}(x) = f(x, \theta^{(\kappa)}) = \prod_{l=1}^{|x|} \theta^{(\kappa)}_{i(x_1^{l-1}),\, j(x_l)}, \tag{3.2}$$

where $i(x_1^{l-1})$ and $j(x_l)$ denote the ordinal numbers of the context $x_1^{l-1} = c_i \in A^*$ and the character $x_l = a_j \in A$ for some orderings on the sets A and A^*, and parameters $\theta_{ij}^{(\kappa)}$ are the probabilities $P(\xi_l = a_j \mid \xi_1^{l-1} = c_i, C^{(\kappa)})$.

From this point on, we will also assume that each class κ can be modelled as a stationary and ergodic Markov chain which memory is bounded by certain *order* $K \geq 1$. Under the assumption that limited memory K is sufficient for evaluating probability (3.2), we can rewrite it for our convenience as

$$f(x, \theta) = \prod_{l=1}^{|x|} \theta_{i(x_{l-K}^{l-1}),\, j(x_l)} = \prod_{\substack{c_i \in A^K \\ a_j \in A}} \theta_{ij}^{n_{ij}(x)} \tag{3.3}$$

for $n_{ij}(x)$ being the number of times character a_j follows context c_i in string x (or, alternatively, substring $c_i a_j$ occurs in x), where

$$\sum_{a_j \in A} \theta_{ij} = 1, \qquad \text{for all } c_i \in A^K, \tag{3.4}$$

$$\sum_{\substack{c_i \in A^K \\ a_j \in A}} n_{ij}(x) = |x|. \tag{3.5}$$

This way, any string x is viewed as a set of overlapping $(K+1)$-grams with frequencies $n_{ij}(x)$, and parameters θ_{ij} characterize a class of strings as a whole.

Reasoning completely analogously for the probability $g(x)$, we obtain the same parametrized form:

$$g(x,\tau) = \prod_{\substack{c_i \in A^K \\ a_j \in A}} \tau_{ij}^{n_{ij}(x)}. \tag{3.6}$$

To avoid confusion, we use the letter τ to denote the vector of parameters of the input source as distinguished from vectors of class parameters $\theta^{(\kappa)}$.

3.1.3. Problem Statement

The above parametrization following from finite memory Markov models allows us to view the mathematical problem of inferring a class model as an optimization problem in the space of parameters:

$$R(\theta) = \mathbf{E}_{\xi}\big[\, r(\xi,\theta)\,\big] \to \min_{\theta} \tag{3.7}$$

for some measure function $r(\xi,\theta)$ evaluating the "loss" or "penalty" of classifying message ξ as belonging to the class described by the probability distribution with parameters θ. In other words, the objective of the problem (3.7) for each class κ is to find parameters $\theta^{(\kappa)}$ giving the least losses on average according to $r(\xi,\theta^{(\kappa)})$. The expectation is taken over the probability distribution $g(x)$ of strings ξ from input source. Generally, probability $g(x)$ is supposed to be unknown for all classes. For this reason, a version of the problem (3.7) for empirical averaging is considered:

$$\widehat{R}(\theta) = \sum_{x_k \in T} r(x_k,\theta) \to \min_{\theta}, \tag{3.8}$$

where T stands for a training sample of messages corresponding to the class in question. Hereinafter, for consistency, training samples of Ham and Spam classes are labeled as $T^{(\mathrm{H})}$ and $T^{(\mathrm{S})}$ accordingly.

When the inference problem is solved and the vectors of parameters $\theta^{(\mathrm{H})}$ and $\theta^{(\mathrm{S})}$ are estimated for each class, they can be used to make classifying decision based on the same principle of least loss:

$$q(x,\theta) = r(x,\theta^{(\mathrm{H})}) - r(x,\theta^{(\mathrm{S})}), \tag{3.9}$$

$$\kappa(x) = \begin{cases} \mathrm{H}, & \text{if } q(x,\theta) < \alpha^{(\mathrm{H})}; \\ \mathrm{S}, & \text{if } q(x,\theta) \geq \alpha^{(\mathrm{S})}. \end{cases} \tag{3.10}$$

In case of $\alpha^{(\mathrm{H})} \leq q(x,\theta) < \alpha^{(\mathrm{S})}$, additional measures are needed to decide the class (for example, increasing the length of the message in question). Most commonly, both parameters are set the same value, $\alpha^{(\mathrm{H})} = \alpha^{(\mathrm{S})} = \alpha$. The choice of parameter α is guided by its influence on the number of type I and type II errors.

3.1.4. Entropy Classification

Let us consider the measure function $r(\xi,\theta) = -\frac{1}{|\xi|}\log f(\xi,\theta)$. As it is obvious from the above definitions, the general criterial function (3.7) specializes to the cross entropy

$$R(\theta) = H(\theta) \equiv -\mathbf{E}_{\xi}\left[\frac{1}{|\xi|}\log f(\xi,\theta)\right] = -\sum_{x\in X}\frac{1}{|x|}g(x)\log f(x,\theta) \to \min_{\theta}. \tag{3.11}$$

We will refer to this specialization of the problem (3.7) as the *classifier problem*.

Similarly, the empiric version (3.8) becomes

$$\widehat{R}(\theta) = \widehat{H}(\theta) \equiv -\sum_{x_k \in T} \frac{1}{|x_k|} \log f(x_k, \theta) \to \min_{\theta}, \tag{3.12}$$

where T, of course, is assumed to be a sample of strings distributed according to $g(x)$.

Decision rule (3.10) can be rewritten as follows.

$$q(x, \theta) = \frac{1}{|x|} \log f^{(\mathrm{S})}(x) - \frac{1}{|x|} \log f^{(\mathrm{H})}(x) = \frac{1}{|x|} \log \frac{f^{(\mathrm{S})}(x)}{f^{(\mathrm{H})}(x)}, \tag{3.13}$$

$$\kappa(x) = \begin{cases} \mathrm{H}, & \text{if } q(x, \theta) < \alpha; \\ \mathrm{S}, & \text{if } q(x, \theta) \geq \alpha. \end{cases} \tag{3.14}$$

In practice, parameter α is often set to zero.

It is well known that if the function $g(x)$ is given and $f(x, \theta) > 0$ for all x such that $g(x) > 0$, then

$$f(x, \theta) \propto \frac{g(x)}{|x|} \tag{3.15}$$

is an exact solution of the problem (3.11). Because, as we have seen above, both $f(x)$ and $g(x)$ can be parametrized identically, at least in the case when all texts x have the same length (or the variation in lengths can be neglected), $f(x, \theta)$ can be constructed from $g(x, \tau)$ by letting $\theta = \tau$. The parameters τ, in turn, can be directly found by estimating conditional probabilities $P(x_l \mid x_{l-K}^{l-1})$ on some training sample T.

This observation forms the basis of the technique called Prediction by Partial Matching (PPM). Aside from differences in strategies of approximating probabilities for character-context pairs that do not occur in a given sample, PPM algorithms work as

simple frequency estimators setting

$$\theta_{ij} \approx \frac{N_{ij}}{N_i}, \tag{3.16}$$

for

$$N_{ij} = \sum_{x \in T} n_{ij}(x), \tag{3.17}$$

$$N_i = \sum_{a_j \in A} N_{ij}. \tag{3.18}$$

For versions of PPM estimators and the details of their implementation, see (Cleary & Witten, 1984; Teahan, 1995; Cleary & Teahan, 1997; Moffat, 1990).

3.2. Adversary Problem

As we just seen above, in the classifier problem (3.11) the goal was to find an optimal statistical model $f(x, \theta)$ for messages of some class, given a fixed input source defined by probabilities $g(x)$ which manifest itself in a sample T. To put it more strictly, the function $g(x)$ was fixed (although unknown), while the probability distribution $f(x, \theta)$ was known up to the vector θ which were the parameters in question.

It is also of interest to consider the inverse problem statement where given fixed statistical model $f(x, \theta)$ of some class, it is required to find the source distribution $g(x)$ which is the most favourable for certain classification outcome. In this setting, $g(x)$ becomes the function in question, while $f(x, \theta)$ is fixed through a known vector of parameters θ.

One example of such inverse objective is the problem of determining $g(x)$ generating messages that are as close to Ham messages as possible in terms of probability of passing the spam filter. Another version of the problem that also falls into this category is the following *adversary problem* (or, in case of spam filtering, the *spammer problem*).

For a given string z from some set of *base messages* Z, find probability distribution for generating strings x_t such that the result of some transformation $\psi(z, x_t)$ combining them complies with some statistical requirement, e.g. being classified as Ham on average. This setting is especially practical for a spammer when z by itself has low chances of passing the filter.

To state the spammer problem more formally, we assume the following. There is a *generator* algorithm which plays the role of a source of strings $x_t(\tau)$ for a specified vector of parameters τ. Strings $x_t(\tau)$ are considered to be generated randomly and independently, and have the same distribution in the space of strings X. The generated strings $x_t(\tau)$ are then used to obtain new messages $u_t = \psi(z, x_t(\tau))$ from a given message z according to the predetermined transformation ψ. In general, the function $\psi(z, x)$ can associate a pair of strings with any string. One such transformation that is simple but still keeps the problem non-trivial is string concatenation, $\psi(z, x) = zx$. Even though our method does not sufficiently depend on a particular transformation, for illustration purposes, when necessary, we will not be using other definitions of ψ except for the concatenation.

The objective of the inverse problem itself remains the same:

$$G(\tau) \equiv -\sum_{x \in X} \frac{1}{|x|} g(x, \tau) \log f(x) \to \min_{\tau}, \tag{3.19}$$

with the exception of that optimization is done for the parameters τ of the source distribution $g(x, \tau)$, not the class distribution $f(x, \theta) = f(x)$. The decision to search in the parametrized space of distributions $g(x, \tau)$ is justified by the necessity to obtain a generative (rather than discriminative) model of the desired message source.

As in the case of the classifier problem (3.11), it is well known that, in non-parametrical form, the inverse problem (3.19) also has an analytical solution. Any

function $g(x)$ such that

$$\sum_{x \in X_{f_{\max}}} g(x) = 1, \tag{3.20}$$

$$g(x) = 0, \qquad \text{for all } x \in X \setminus X_{f_{\max}}, \tag{3.21}$$

where

$$X_{f_{\max}} = \arg\max_{x} \frac{f(x)}{|x|}, \tag{3.22}$$

minimizes the cross entropy for a given $f(x)$. These solutions for the non-parametric problem, however, does not solve the spammer problem. None of the functions $g(x)$ satisfying the above properties is guaranteed to be represented in the space of parametrized functions $g(x, \tau)$ which makes them useless for generating $x_t(\tau)$. Moreover, even if this difficulty did not exist, the diversity of the generated messages would be extremely low, because any of such $g(x)$ leads to generating the same few messages from $X_{f_{\max}}$ over and over again which makes spammer easily detectable.

Empirical analog of the criterion (3.19) is

$$\widehat{G}(\tau) \equiv \sum_{x_k \in T} \frac{1}{|x|} g(x_k, \tau) \to \min_{\tau}, \tag{3.23}$$

where the sample T is obtained from a distribution $P(\xi = x) \propto \log \frac{1}{f(x)}$. Therefore, in order to approach the inference problem in the form (3.23), it is necessary to have an auxiliary instrumental sample which, unlike training samples for the classes or the combined sample for the input source, cannot be observed in practice.

3.3. Instrumental Sampling Approach

Let us introduce new parameters w_{ij} such that

$$\tau_{ij} = \frac{\exp(w_{ij})}{\sum_{a_j \in A} \exp(w_{ij})}, \tag{3.24}$$

where, as before, subscripts i and j correspond to some context $c_i \in A^K$ and character $a_j \in A$, respectively. For any values of w_{ij}, the required conditions on τ_{ij} hold automatically:

$$0 < \tau_{ij} < 1 \text{ and } \sum_{a_j \in A} \tau_{ij} = 1 \tag{3.25}$$

($0 \le \tau_{ij} \le 1$, if $w_{ij} = \pm\infty$ are allowed).

For the new parameters, the probability

$$g(x, \tau) = \prod_{\substack{c_i \in A^K \\ a_j \in A}} \tau_{ij}^{n_{ij}(x)} \tag{3.26}$$

changes to

$$\begin{aligned} g(x, \tau(w)) &= \prod_{\substack{c_i \in A^K \\ a_j \in A}} \left(\frac{\exp(w_{ij})}{\sum_{a_{j'} \in A} \exp(w_{ij'})} \right)^{n_{ij}(x)} \\ &= \prod_{c_i \in A^K} \frac{1}{Z_i^{n_i(x)}} \exp\left(\sum_{a_j \in A} w_{ij} n_{ij}(x) \right) \\ &= \prod_{c_i \in A^K} \left(\frac{1}{Z_i} \exp\left(\sum_{a_j \in A} w_{ij} \frac{n_{ij}(x)}{n_i(x)} \right) \right)^{n_i(x)}, \end{aligned} \tag{3.27}$$

where $n_{ij}(x)$ is, as usual, the number of occurrences of a substring $c_i a_j$ in x, and

$$n_i(x) = \sum_{a_j \in A} n_{ij}(x), \tag{3.28}$$

$$Z_i(w) = \sum_{a_j \in A} \exp(w_{ij}). \tag{3.29}$$

Now, let us calculate the gradient of the function $g(x, \tau(w))$ using the equality

$$\frac{\partial g(x,\tau(w))}{\partial w_{lk}} = g(x,\tau(w)) \frac{\partial}{\partial w_{lk}} \log g(x,\tau(w)), \tag{3.30}$$

where

$$\begin{aligned} \log g(x,\tau(w)) &= \sum_{\substack{c_i \in A^K \\ a_j \in A}} n_{ij}(x) \log\left(\frac{\exp(w_{ij})}{Z_i(w)}\right) \\ &= \sum_{\substack{c_i \in A^K \\ a_j \in A}} w_{ij} n_{ij}(x) - \sum_{\substack{c_i \in A^K \\ a_j \in A}} n_{ij}(x) \log Z_i(w). \end{aligned} \tag{3.31}$$

Then,

$$\begin{aligned} \frac{\partial g(x,\tau(w))}{\partial w_{lk}} &= g(x,\tau(w)) \frac{\partial}{\partial w_{lk}} \Bigg[\sum_{\substack{c_i \in A^K \\ a_j \in A}} w_{ij} n_{ij}(x) - \sum_{\substack{c_i \in A^K \\ a_j \in A}} n_{ij}(x) \log Z_i(w) \Bigg] \\ &= g(x,\tau(w)) \left(n_{lk}(x) - \frac{n_l(x)}{Z_l} \exp(w_{lk}) \right) \\ &= g(x,\tau(w))\, n_l(x) \left(\widehat{\tau}_{lk}(x) - \tau_{lk}(w) \right), \end{aligned} \tag{3.32}$$

where

$$\widehat{\tau}_{lk}(x) = \frac{n_{lk}(x)}{n_l(x)}. \tag{3.33}$$

Now, consider a problem of the form

$$\mathbf{E}_{\xi}\Big[F(\xi,w)\Big] \approx \sum_{x_k \in T} F(x_k,w) \to \min_{w}, \tag{3.34}$$

where the random variable $\xi(w)$ is distributed and strings x_k from an instrumental sample T are generated according to the probabilities $g(x,\tau(w))$. The problem (3.23) that has motivated us to consider this approach is a special case for

$$F(\xi,w) = F(x) = \frac{1}{|x|}\log\frac{1}{f(x)}. \tag{3.35}$$

Given that

$$\begin{aligned}
\frac{\partial}{\partial w_{lk}}\mathbf{E}_{\xi}\Big[F(\xi,w)\Big] &= \frac{\partial}{\partial w_{lk}}\Big[\sum_{x\in X} F(x,w)\,g(x,\tau(w))\Big] \\
&= \sum_{x\in X}\frac{\partial}{\partial w_{lk}}\Big[F(x,w)\,g(x,\tau(w))\Big] \\
&= \sum_{x\in X}\Big[\frac{\partial}{\partial w_{lk}}F(x,w)\,g(x,\tau(w)) + F(x,w)\frac{\partial}{\partial w_{lk}}g(x,\tau(w))\Big] \\
&= \sum_{x\in X}\Big[\frac{\partial}{\partial w_{lk}}F(x,w) + F(x,w)\,n_l(x)\big(\widehat{\tau}_{lk}(x)-\tau_{lk}(w)\big)\Big]g(x,\tau(w)) \\
&= \mathbf{E}_{\xi}\Big[F(\xi,w)\Big(\frac{\partial}{\partial w_{lk}}\Big[\log F(\xi,w)\Big] + n_l(\xi)\big(\widehat{\tau}_{lk}(\xi)-\tau_{lk}(w)\big)\Big)\Big] \\
&= \mathbf{E}_{\xi}\Big[\frac{\partial}{\partial w_{lk}}F(\xi,w) + F(\xi,w)\,n_l(\xi)\big(\widehat{\tau}_{lk}(\xi)-\tau_{lk}(w)\big)\Big],
\end{aligned} \tag{3.36}$$

from the necessary condition of extremum, we see that optimal w satisfies the equation

$$\mathbf{E}_{\xi}\Big[\frac{\partial}{\partial w_{lk}}F(\xi,w) + F(\xi,w)\,n_l(\xi)\big(\widehat{\tau}_{lk}(\xi)-\tau_{lk}(w)\big)\Big] = 0, \tag{3.37}$$

which, in the case when $F(x,w) = F(x)$ is independent of parameters, simplifies to

$$\mathbf{E}_{\xi}\Big[F(\xi)\, n_l(\xi)\big(\widehat{\tau}_{lk}(\xi) - \tau_{lk}(w)\big)\Big] = 0, \tag{3.38}$$

for all $c_l \in A^K$, $a_k \in A$. (Both $n_l(x)$ and $n_{lk}(x)$ are random variables and, consequently, cannot be factored out of the expectation.)

Since $\xi \sim g(x, \tau(w))$, as the size of instrumental sample T grows, frequencies $\widehat{\tau}_{lk}(x)$ converge to the current estimations $\tau_{lk}(w)$ that were used to generate the sample in the first place. For this reason, any attempt of iterative optimization of (3.34) turns into a random walk around initial values of w_{ij}.

Moreover, for many practical generation procedures it is true that

$$\mathbf{E}_{\xi}\big[\widehat{\tau}_{lk}(\xi)\big] = \tau_{lk}(w). \tag{3.39}$$

In a simplified case of both $F(x)$ and $n_l(x)$ being independent of parameters w, which takes place when, for example, generation procedure stops after reaching the same length of x chosen beforehand, the equation (3.38) simply degenerates, and the problem becomes meaningless.

If the function F preserves some dependence on parameters—either in the general form $F(x,w)$, or in a weaker variant $F(x(w))$—the problem (3.38) is not strictly meaningless. However, for sufficiently long samples, as the difference $|\widehat{\tau}_{lk}(\xi) - \tau_{lk}(w)|$ approaches zero, the influence of the $F(x,w)$-multiplier becomes effectively eliminated making the expectation (3.38) almost independent of F. For this reason, we consider approaching problem (3.19) as (3.34) unpromising.

3.4. Importance Sampling Approach

Formally, we consider a vector of parameters τ to be a solution to the inverse problem, if

$$\mathscr{F}_D[q(u,\theta)] \equiv \mathscr{F}[q(u,\theta) \,|\, u = \psi(z,x), x \in D] \to \max_{\tau}, \tag{3.40}$$

where D is a set of text strings, the domain, and $\mathscr{F}(\cdot)$ is an ensemble operation defined on D. For example, the domain D might be the set of all strings of some bounded length, or some subset of that set. An empirical sample of strings produced by the generator used by the adversary can also be taken as the domain $D_\tau = \{x_t(\tau)\}_t$.

The choice of ensemble operation depends on what criterion of success aligns best with the goals of the spammer in a particular problem setting. Let us consider some of them.

(a) For all $x \in D_\tau$, messages $u = \psi(z,x)$ are successfully pass the spam filter:

$$\mathscr{F}_{D_\tau}\big[q(u,\theta)\big] = \min_{x \in D_\tau} \mathbf{1}^{(\mathrm{H})}(u) \to \max_{\tau}, \tag{3.41}$$

where

$$\mathbf{1}^{(\mathrm{H})}(u) \equiv \mathbf{1}\big[q(u,\theta) < \alpha\big] = \mathbf{1}\big[\log f^{(\mathrm{S})}(u) - \log f^{(\mathrm{H})}(u) < \alpha|u|\big]. \tag{3.42}$$

(b) As many messages $x \in D_{\tau,l} = \{x \in D_\tau \mid |x| \le l\}$ of a bounded length l are successfully pass the spam filter:

$$\mathscr{F}_{D_{\tau,l}}\big[q(u,\theta)\big] = \sum_{x \in D_{\tau,l}} \mathbf{1}^{(\mathrm{H})}(u) \to \max_{\tau}. \tag{3.43}$$

(c) Empirical frequency of passing the spam filter successfully estimated over a sample D_τ is maximal:

$$\mathscr{F}_{D_\tau}\big[q(u,\theta)\big] = \frac{1}{|D_\tau|}\sum_{x\in D_\tau} \mathbf{1}^{(\mathrm{H})}(u) \to \max_\tau. \tag{3.44}$$

(d) The average logarithmic ratio of probabilities $q(u,\theta)$ estimated over a sample D_τ is as minimal as possible:

$$\mathscr{F}_{D_\tau}\big[q(u,\theta)\big] = -\sum_{x\in D_\tau} q(u,\theta) \to \max_\tau, \tag{3.45}$$

or

$$\sum_{x\in D_\tau} q(u,\theta) \to \min_\tau. \tag{3.46}$$

Criterion (a) is too optimistic and requires the acceptance of the implicit assumption that there exists a vector τ which guarantees that all messages pass the filter. Solution of the problem in the sense of this criterion, generally, is unlikely to exist (unless the generator algorithm is complemented with constraints significantly restricting diversity of generated strings).

Criterion (b) does not take probabilities of strings x into account, while it would be worthwhile to ignore strings with zero or close to zero probabilities.

We consider criteria (c) and (d) to be more appropriate. Let us discuss the latter objective before the former.

3.4.1. Entropy-Based Criterion

Empirical criterion (3.45) is equivalent to the optimization problem

$$\mathscr{F}\big[q(u,\theta)\big] = \sum_{x\in X} q(u,\theta)\, g(x \mid z,\tau) = \mathbf{E}_\xi\big[q(u,\theta)\big] \to \min_\tau, \tag{3.47}$$

where the expected value is taken over the probability distribution $g(x \mid z, \tau)$ of text x being generated for the base string z and parameters τ.

Let us now rearrange the sum in (3.47) using the well known technique of importance sampling:

$$\begin{aligned} R(\tau \mid z) &= \mathbf{E}_{\xi}\Big[q(u,\theta)\Big] \\ &= \sum_{x \in X} q(u,\theta)\, g(x \mid z, \tau)\left(\gamma^{(\mathrm{H})}\frac{p^{(\mathrm{H})} f^{(\mathrm{H})}(x)}{p^{(\mathrm{H})} f^{(\mathrm{H})}(x)} + \gamma^{(\mathrm{S})}\frac{p^{(\mathrm{S})} f^{(\mathrm{S})}(x)}{p^{(\mathrm{S})} f^{(\mathrm{S})}(x)}\right) \\ &= \sum_{\kappa \in \{\mathrm{H},\mathrm{S}\}} p^{(\kappa)}\, \mathbf{E}_{\xi}^{(\kappa)}\left[\gamma^{(\kappa)}\, q(\xi \mid z, \theta)\, \frac{g(\xi \mid z, \tau)}{p^{(\kappa)} f^{(\kappa)}(\xi)}\right] \\ &= \mathbf{E}_{\xi,\kappa}\left[q(\xi \mid z, \theta)\, \frac{g(\xi \mid z, \tau)}{p^{(\kappa)} f^{(\kappa)}(\xi)}\right] \\ &= \mathbf{E}_{\xi,\kappa}\Big[W^{(\kappa)}(\xi \mid z, \theta)\, g(\xi \mid z, \tau)\Big] \to \min_{\tau}, \end{aligned} \tag{3.48}$$

for

$$W^{(\kappa)}(x \mid z, \theta) = \frac{\gamma^{(\kappa)}\, q(x \mid z, \theta)}{p^{(\kappa)} f^{(\kappa)}(x)}, \tag{3.49}$$

where for each class $\kappa \in \{\mathrm{H}, \mathrm{S}\}$, expected value $E_{\xi}^{(\kappa)}[\,\cdot\,]$ denotes conditional expectation $E_{\xi}[\,\cdot \mid \xi \sim f^{(\kappa)}(x)]$, $p^{(\kappa)}$ stands for the a priori probability of the class κ, and $\gamma^{(\mathrm{H})}$, $\gamma^{(\mathrm{S})}$ are arbitrary splitting weights such that $\gamma^{(\mathrm{H})} + \gamma^{(\mathrm{S})} = 1$ (for example, $\gamma^{(\mathrm{H})} = \gamma^{(\mathrm{S})} = \frac{1}{2}$ or $\gamma^{(\mathrm{H})} = p^{(\mathrm{H})}$, $\gamma^{(\mathrm{S})} = p^{(\mathrm{S})}$).

In this problem setting, all statistical information that can be available to the adversary—that is, both samples $T^{(\mathrm{H})}$ and $T^{(\mathrm{S})}$ of Ham and Spam messages—are used:

$$R(\tau \mid z) \approx \widehat{R}(\tau \mid z) = \sum_{(x_k, \kappa_k) \in T} W^{(\kappa_k)}(x_k \mid z,\, \theta)\, g(x_k \mid z, \tau) \to \min_{\tau}. \tag{3.50}$$

Here κ_k are true labellings of messages x_k from the sample T which is the union of samples $T^{(\mathrm{H})}$ and $T^{(\mathrm{S})}$ that are assumed to be drawn from distributions $f^{(\mathrm{H})}(x)$ and $f^{(\mathrm{S})}(x)$, respectively.

Due to the necessary condition of extremum, we have the equation:

$$\begin{aligned}\frac{\partial}{\partial \tau_{ij}} R(\tau \mid z) &= \frac{\partial}{\partial \tau_{ij}} \mathbf{E}_{\xi,\kappa}\Big[W^{(\kappa)}(\xi \mid z, \theta)\, g(\xi \mid z, \tau)\Big] \\ &= \mathbf{E}_{\xi,\kappa}\Big[W^{(\kappa)}(\xi \mid z, \theta)\, \frac{\partial}{\partial \tau_{ij}} g(\xi \mid z, \tau)\Big] = 0. \end{aligned} \tag{3.51}$$

Since it has the form $\mathbf{E}[\,\cdot\,] = 0$, it is natural for us to apply the method of stochastic optimization (Robbins & Monro, 1951). Switching to the parameters w_{ij} that introduced in (3.24), we obtain the stochastic algorithm

$$\begin{aligned} w_{ij}^{(t+1)} = w_{ij}^{(t)} - \gamma_t W^{(\kappa_{k(t)})}(z, x_{k(t)})\, g(x_{k(t)} \mid z, \tau(w^{(t)})) \\ \cdot\, n_i(x_{k(t)} \mid z)\big(\widehat{\tau}_{ij}(x_{k(t)} \mid z) - \tau_{ij}(w^{(t)})\big), \end{aligned} \tag{3.52}$$

where γ_t is a series satisfying the properties

$$\gamma_t \geq 0, \qquad \text{for all } t \geq 0, \tag{3.53}$$

$$\sum_{t=0}^{\infty} \gamma_t = \infty, \tag{3.54}$$

$$\sum_{t=0}^{\infty} \gamma_t^2 < \infty, \tag{3.55}$$

and $x_{k(t)}$ and $\kappa_{k(t)}$ run through the sample T (potentially repeatedly) in some order defined by $k(t)$.

3.4.2. Probability-Based Criterion

Obviously, objective function (3.44) is the empirical version of the criterion

$$\mathscr{F}\big[q(u,\theta)\big] = \sum_{x\in X} \mathbf{1}^{(\mathrm{H})}(u)\, g(x \mid z,\tau) = \mathbf{E}_{\xi}\big[\mathbf{1}^{(\mathrm{H})}(u)\big] \to \max_{\tau}, \tag{3.56}$$

where $\xi \sim g(x \mid z,\tau)$ and, as in the previous section, $g(x \mid z,\tau)$ is generational probability distribution for base string z and parameters τ. This criterion, in turn, makes the problem be equivalent to maximizing the probability of the transformed message $\psi(z,\xi)$ passing the spam filter:

$$R(\tau) = \Pr\big[\mathbf{1}^{(\mathrm{H})}(\psi(z,\xi)) \mid z,\tau\big] \to \max_{\tau}. \tag{3.57}$$

Since we have only two classes, maximization of the criterion (3.56),

$$R^{(\mathrm{H})}(\tau) = \sum_{x\in X} \mathbf{1}^{(\mathrm{H})}(\psi(z,x))\, g(x \mid z,\tau), \tag{3.58}$$

is equivalent to minimization of the dual criterion

$$R^{(\mathrm{S})}(\tau) = \sum_{x\in X} \mathbf{1}^{(\mathrm{S})}(\psi(z,x))\, g(x \mid z,\tau), \tag{3.59}$$

where $\mathbf{1}^{(\mathrm{S})}(u) = 1 - \mathbf{1}^{(\mathrm{H})}(u)$. Let us combine both of them into a single problem

$$R(\tau) \equiv \gamma^{(\mathrm{H})} R^{(\mathrm{H})}(\tau) - \gamma^{(\mathrm{S})} R^{(\mathrm{S})}(\tau) \tag{3.60}$$

$$= \sum_{x\in X} \Big(\gamma^{(\mathrm{H})} \mathbf{1}^{(\mathrm{H})}(\psi(z,x)) - \gamma^{(\mathrm{S})} \mathbf{1}^{(\mathrm{S})}(\psi(z,x))\Big) g(x \mid z,\tau) \to \max_{\tau}, \tag{3.61}$$

where $\gamma^{(\mathrm{H})}$ and $\gamma^{(\mathrm{S})}$ are some splitting weights such that $\gamma^{(\mathrm{H})} + \gamma^{(\mathrm{S})} = 1$.

3.4.2.1. Supervised Learning

Formally rearranging the criterion function (3.60) into two sums and applying the importance sampling for the distribution of the pair (ξ, κ), we see that

$$\begin{aligned} R(\tau) &= \sum_{x \in X} \Big(\gamma^{(\mathrm{H})} \mathbf{1}^{(\mathrm{H})}(\psi(z,x)) - \gamma^{(\mathrm{S})} \mathbf{1}^{(\mathrm{S})}(\psi(z,x)) \Big) g(x \mid z, \tau) \\ &= \sum_{\kappa \in \{\mathrm{H},\mathrm{S}\}} p^{(\kappa)} \sum_{x \in X} W^{(\kappa)}(z,x) f^{(\kappa)}(x)\, g(x \mid z, \tau) \\ &= \mathbf{E}_{\xi} \Big[W^{(\kappa)}(z,\xi)\, g(\xi \mid z, \tau) \Big] \to \max_{\tau}, \end{aligned} \tag{3.62}$$

for the random variable ξ distributed according to $f^{(\kappa)}(x)$ and

$$\begin{aligned} W^{(\kappa)}(z,x) &= \frac{\gamma^{(\mathrm{H})} \mathbf{1}^{(\mathrm{H})}(\psi(z,x)) - \gamma^{(\mathrm{S})} \mathbf{1}^{(\mathrm{S})}(\psi(z,x))}{f^{(\kappa)}(x)} \\ &= \frac{\mathbf{1}^{(\mathrm{H})}(\psi(z,x)) - \gamma^{(\mathrm{S})}}{f^{(\kappa)}(x)}. \end{aligned} \tag{3.63}$$

Assuming that a sample of messages $x_k \in T$ is available together with true labeling of classes $\kappa_k = \kappa(x_k)$, the criterion $R(\tau)$ can be estimated as

$$R(\tau) \approx \widehat{R}(\tau) \equiv \sum_{(x_k, \kappa_k) \in T} W^{(\kappa_k)}(z, x_k)\, g(x_k \mid z, \tau). \tag{3.64}$$

For the parameters w_{ij} that have been introduced earlier this chapter through the equality

$$\tau_{ij} = \frac{\exp(w_{ij})}{\sum_{a_j \in A} \exp(w_{ij})}, \tag{3.65}$$

we obtain that

$$\frac{\partial R(w)}{\partial w_{ij}} = \mathbf{E}_{\xi}\left[W^{(\kappa)}(z,\xi)\, g(\xi \mid z, \tau(w))\left(n_{ij}(\xi \mid z) - \frac{n_i(\xi \mid z)\exp(w_{ij})}{\sum_{a_l \in A}\exp(w_{il})} \right)\right]$$
$$= \mathbf{E}_{\xi}\Big[W^{(\kappa)}(z,\xi)\, g(\xi \mid z, \tau(w))\, n_i(\xi \mid z)\big(\widehat{\tau}_{ij}(\xi \mid z) - \tau_{ij}(w)\big)\Big], \tag{3.66}$$

where, as in previous sections,

$$\widehat{\tau}_{lk}(x) = \frac{n_{lk}(x)}{n_l(x)}, \tag{3.67}$$

and $n_i(x \mid z)$ and $n_{ij}(x \mid z)$ stand for the number of occurrences of the context $c_i \in A^K$ followed by any character and followed by the character $a_j \in A$, respectively, in the text x appended to the message z.

Thus, the stochastic approximation algorithm for the necessary condition of the extremum,

$$\frac{\partial R(w)}{\partial w_{ij}} = 0, \tag{3.68}$$

takes the form

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \gamma_t W^{(\kappa_{k(t)})}(z, x_{k(t)})\, g(x_{k(t)} \mid z, \tau(w^{(t)}))$$
$$\cdot\, n_i(x_{k(t)} \mid z)\big(\widehat{\tau}_{ij}(x_{k(t)} \mid z) - \tau_{ij}(w^{(t)})\big). \tag{3.69}$$

3.4.2.2. Unsupervised Learning

In case when true labeling κ_k of messages x_k from sample T is unknown, we can alternatively do the importance sampling for the distribution

$$f(x) = p^{(\mathrm{H})} f^{(\mathrm{H})}(x) + p^{(\mathrm{S})} f^{(\mathrm{S})}(x). \tag{3.70}$$

Then

$$R(\tau) = \sum_{x \in X} \frac{\gamma^{(\mathrm{H})} \mathbf{1}^{(\mathrm{H})}(\psi(z,x)) - \gamma^{(\mathrm{S})} \mathbf{1}^{(\mathrm{S})}(\psi(z,x))}{f(x)} f(x)\, g(x \mid z, \tau)$$
$$= \mathbf{E}_{\xi}\Big[W(z,\xi)\, g(\xi \mid z, \tau) \Big] \to \max_{\tau}, \tag{3.71}$$

where the random variable ξ is distributed in accordance with $f(x)$, and

$$W(z,x) = \frac{\gamma^{(\mathrm{H})} \mathbf{1}^{(\mathrm{H})}(\psi(z,x)) - \gamma^{(\mathrm{S})} \mathbf{1}^{(\mathrm{S})}(\psi(z,x))}{f(x)}$$
$$= \frac{\mathbf{1}^{(\mathrm{H})}(\psi(z,x)) - \gamma^{(\mathrm{S})}}{p^{(\mathrm{H})} f^{(\mathrm{H})}(x) + p^{(\mathrm{S})} f^{(\mathrm{S})}(x)}. \tag{3.72}$$

Since the criteria (3.71) and (3.62) differ only in definition of the weights $W(z,x)$ which are independent of w_{ij}, the resulting stochastic optimization algorithm is exactly the same as in (3.69) (again, up to differences between $W(z,x)$ and $W^{(\kappa)}(z,x)$).

3.5. Likelihood-Based Criterion

Let us again consider the transformation $u = \psi(z,x)$ of a message z with an arbitrary string x. Entropy per character of the resulting string u can be estimated empirically as

$$H(u \mid \tau) = -\frac{1}{|u|} \log\Bigg(\prod_{l=1}^{|u|} g(u_l \mid u_{l-K}^{l-1}, \tau) \Bigg) = -\frac{1}{|u|} \sum_{\substack{c_i \in A^K \\ a_j \in A}} n_{ij}(u) \log \tau_{ij}. \tag{3.73}$$

Averaged over random transformed messages u, it is equal to

$$H(\tau) = \mathbf{E}_u\Big[H(u \mid \tau) \Big] = -\sum_{c_i \in A^K} p_i \sum_{a_j \in A} p_{j|i} \log \tau_{ij} = -\sum_{c_i \in A^K} p_i\, H_i(\tau), \tag{3.74}$$

where

$$H_i(\tau) = \sum_{a_j \in A} p_{j|i} \log \tau_{ij}, \tag{3.75}$$

p_i is the probability of the context c_i occurring in a random transformed message u, and $p_{j|i}$ is the conditional probability of character a_j occurring in u after the context c_i. The function $H_i(\tau)$, in turn, can be estimated on a single message u as

$$H_i(\tau) \approx \widehat{H}_i(u \mid \tau) = \sum_{a_j \in A} \frac{n_{ij}(u)}{n_i(u)} \log \tau_{ij}. \tag{3.76}$$

Assuming that we have available a sample T of messages x out of the universal space X, we can split T into auxiliary samples depending on to what class $\psi(z, x)$ is assigned by the classifier:

$$T^{(\kappa)} = \{x_k \in T \mid \mathbf{1}^{(\kappa)}(\psi(z, x_k) \mid \theta) = 1\}, \tag{3.77}$$

where, as in previous sections,

$$\mathbf{1}^{(\mathrm{H})}(x \mid \theta) = \mathbf{1}[q(x, \theta) < \alpha], \tag{3.78}$$

$$\mathbf{1}^{(\mathrm{S})}(x \mid \theta) = \mathbf{1}[q(x, \theta) \geq \alpha]. \tag{3.79}$$

That is, $T^{(\mathrm{H})}$ and $T^{(\mathrm{S})}$ consist of messages $x_k \in T$ that make the base message z being recognized as Ham and Spam, respectively.

Considering these samples, we can generalize the estimate $\widehat{H}(u \mid \tau)$ to the estimates over samples $T^{(\mathrm{H})}$ and $T^{(\mathrm{S})}$:

$$R^{(\mathrm{H})}(\tau \mid z) = -\frac{1}{|T^{(\mathrm{H})}|} \sum_{x_k \in T^{(\mathrm{H})}} \sum_{c_i \in A^K} p_i \widehat{H}_i(\psi(z, x_k) \mid \tau), \tag{3.80}$$

$$R^{(\mathrm{S})}(\tau \mid z) = -\frac{1}{|T^{(\mathrm{S})}|} \sum_{x_k \in T^{(\mathrm{S})}} \sum_{c_i \in A^K} p_i \widehat{H}_i(\psi(z, x_k) \mid \tau). \tag{3.81}$$

Then, we can state our goal in a new way: Find parameters τ such that the entropy estimate $R^{(\mathrm{H})}(\tau \mid z)$ becomes low, while the estimate $R^{(\mathrm{S})}(\tau \mid z)$ remains high. One way to achieve these goals simultaneously is to formalize them as a problem of minimization the difference of the above objective functions:

$$R(\tau) = R^{(\mathrm{H})}(\tau \mid z) - R^{(\mathrm{S})}(\tau \mid z) \to \min_{\tau}, \tag{3.82}$$

subject to usual normalization requirements

$$\tau_{ij} \geq 0 \text{ and } \sum_{a_j \in A} \tau_{ij} = 1, \tag{3.83}$$

for all contexts $c_i \in A^K$ and all characters $a_j \in A$.

Substituting the entropy estimation (3.76) definition into the criterion (3.82), we have:

$$\begin{aligned} R(\tau \mid z) &= R^{(\mathrm{H})}(\tau \mid z) - R^{(\mathrm{S})}(\tau \mid z) \\ &= -\frac{1}{|T^{(\mathrm{H})}|} \sum_{u_k \in U^{(\mathrm{H})}} \sum_{c_i \in A^K} p_i \widehat{H}_i(u \mid \tau) + \frac{1}{|T^{(\mathrm{S})}|} \sum_{u_k \in U^{(\mathrm{S})}} \sum_{c_i \in A^K} p_i \widehat{H}_i(u \mid \tau) \\ &= -\sum_{c_i \in A^K} p_i \left(\frac{1}{|T^{(\mathrm{H})}|} \sum_{u_k \in U^{(\mathrm{H})}} \widehat{H}_i(u \mid \tau) - \frac{1}{|T^{(\mathrm{S})}|} \sum_{u_k \in U^{(\mathrm{S})}} \widehat{H}_i(u \mid \tau) \right) \\ &= -\sum_{c_i \in A^K} p_i \sum_{a_j \in A} (\nu_{ij}^{(\mathrm{H})} - \nu_{ij}^{(\mathrm{S})}) \log \tau_{ij} \to \min_{\tau}, \end{aligned} \tag{3.84}$$

for

$$U^{(\kappa)} = \{\psi(z, x_k) \mid x_k \in T^{(\kappa)}\}, \tag{3.85}$$

$$\nu_{ij}^{(\kappa)} = \frac{1}{|T^{(\kappa)}|} \sum_{u_k \in U^{(\kappa)}} \frac{n_{ij}(u_k)}{n_i(u_k)}. \tag{3.86}$$

Since parameters τ_{ij} occur only in summands for the context c_i, optimization (3.84) naturally falls into $|A^K|$ smaller problems:

$$R_i(\tau \mid z) = R_i^{(\mathrm{H})}(\tau \mid z) - R_i^{(\mathrm{S})}(\tau \mid z) \to \min_{\{\tau_{i1}, \tau_{i2}, \ldots, \tau_{i|A|}\}}, \tag{3.87}$$

where

$$R_i^{(\kappa)}(\tau \mid z) = -\sum_{a_j \in A} \nu_{ij}^{(\kappa)} \log \tau_{ij}. \tag{3.88}$$

Lemma 1. *The objective function*

$$R_i(\tau) = -\sum_{j \in J} \nu_{ij} \log \tau_{ij}, \tag{3.89}$$

where weights $\nu_{ij} \geq 0$ *for any* $j \in J$*, and parameters* τ_{ij} *are subject to constraints*

$$\tau_{ij} \geq 0 \ \textit{and} \ \sum_{j \in J} \tau_{ij} = s > 0, \tag{3.90}$$

reaches its minimum value at

$$\tau^*_{ij} = s \frac{\nu_{ij}}{\nu_i}, \tag{3.91}$$

where

$$\nu_i = \sum_{j \in J} \nu_{ij}. \tag{3.92}$$

Proof. Considering that $\log \epsilon \leq (\epsilon - 1)$ for any $\epsilon > 0$, we see that for an arbitrary vector of parameters τ,

$$\begin{aligned} R_i(\tau^*) - R_i(\tau) &= -\sum_{j \in J} \nu_{ij} \log \frac{s \nu_{ij}}{\nu_i} + \sum_{j \in J} \nu_{ij} \log \tau_{ij} \\ &= \sum_{j \in J} \nu_{ij} \log \frac{\tau_{ij} \nu_i}{s \nu_{ij}} \\ &\leq \sum_{j \in J} \nu_{ij} \left(\frac{\tau_{ij} \nu_i}{s \nu_{ij}} - 1 \right) \\ &= \frac{\nu_i}{s} \sum_{j \in J} \tau_{ij} - \sum_{j \in J} \nu_{ij} = 0, \end{aligned} \tag{3.93}$$

by definition of ν_i and normalization requirements on τ. From the obtained relation it immediately follows that $R_i(\tau^*) \leq R_i(\tau)$ for any τ. □

Lemma 2. *The objective function*

$$R_i(\tau) = -\sum_{j \in J} \nu_{ij} \log \frac{1}{\tau_{ij}}, \tag{3.94}$$

where weights $\nu_{ij} \geq 0$ *for any* $j \in J$*, and parameters* τ_{ij} *are subject to constraints*

$$\tau_{ij} \geq 0 \ \textit{and} \ \sum_{j \in J} \tau_{ij} = s > 0, \tag{3.95}$$

reaches its minimum value at

$$\tau_{ij}^{*} = \begin{cases} \dfrac{s}{|J_i^{\min}|}, & \textit{if } j \in J_i^{\min}; \\ 0, & \textit{if } j \in J \setminus J_i^{\min}; \end{cases} \tag{3.96}$$

where

$$J_i^{\min} = \{j \in J \mid \nu_{ij} = \min_{j \in J} \nu_{ij}\}. \tag{3.97}$$

Proof. Let us consider the following values of parameters under the temporary assumption that $\tau_{ij} \geq \varepsilon$ for some arbitrarily small $\varepsilon > 0$ and all $j \in J$.

$$\tau_{ij}^{*}(\varepsilon) = \begin{cases} s\dfrac{1 - (|J| - |J_i^{\min}|)\varepsilon}{|J_i^{\min}|}, & \text{if } j \in J_i^{\min}; \\ s\varepsilon, & \text{if } j \in J \setminus J_i^{\min}. \end{cases} \tag{3.98}$$

We can assume that $\sum_{j \in J \setminus J_i^{\min}} \nu_{ij} > 0$, which is always true unless $J_i^{\min} = J$.

It is clear that for smaller values of ε criterion function $R_i(\tau^{*}(\varepsilon))$ also gets smaller values.

$$R_i(\tau^{*}(\varepsilon)) = -\sum_{j \in J_i^{\min}} \nu_{ij} \log \frac{|J_i^{\min}|}{s\big(1 - (|J| - |J_i^{\min}|)\varepsilon\big)} - \sum_{j \notin J_i^{\min}} \nu_{ij} \log \frac{1}{s\varepsilon}. \tag{3.99}$$

Therefore, passing to the limit, we can make the criterion arbitrarily small while approaching the desired solution τ^*:

$$\lim_{\varepsilon \to 0} R_i(\tau^*(\varepsilon)) = -\infty, \tag{3.100}$$

$$\lim_{\varepsilon \to 0} \tau^*(\varepsilon) = \tau^*. \tag{3.101}$$

Generally speaking, this solution is not unique: any distribution of the probability mass across τ^*_{ij} for $j \in J_i^{\min}$ minimizes the criterion. However, one solution is sufficient for our purposes. □

Theorem 3. *The criterion function*

$$R_i(\tau \mid z) = -\sum_{a_j \in A} (\nu_{ij}^{(\mathrm{H})} - \nu_{ij}^{(\mathrm{S})}) \log \tau_{ij} \tag{3.102}$$

subject to constraints

$$\tau_{ij} \geq 0 \ \textit{and} \ \sum_{a_j \in A} \tau_{ij} = 1, \tag{3.103}$$

reaches its minimum value at

$$\tau^*_{ij} = \frac{\mu_{ij}}{\mu_i}, \tag{3.104}$$

where

$$\mu_{ij} = \max\{0, \nu_{ij}^{(\mathrm{H})} - \nu_{ij}^{(\mathrm{S})}\}, \tag{3.105}$$

$$\mu_i = \sum_{a_j \in A} \mu_{ij}. \tag{3.106}$$

Proof. Let us divide the sum in the objective function $R_i(\tau \mid z)$ into the following three sums over disjoint subsets of indices according to the sign of the difference $\delta_{ij} \equiv$

$\nu_{ij}^{(\mathrm{H})} - \nu_{ij}^{(\mathrm{S})}$:

$$\begin{aligned} R_i(\tau \mid z) &= -\sum_{a_j \in A} \delta_{ij} \log \tau_{ij} \\ &= -\sum_{j \in J_i^{+1}} \delta_{ij} \log \tau_{ij} - \sum_{j \in J_i^{-1}} \delta_{ij} \log \tau_{ij} - \sum_{j \in J_i^{0}} \delta_{ij} \log \tau_{ij} \\ &= -\sum_{j \in J_i^{+1}} \delta_{ij} \log \tau_{ij} - \sum_{j \in J_i^{-1}} (-\delta_{ij}) \log \frac{1}{\tau_{ij}}, \end{aligned} \tag{3.107}$$

where

$$J_i^{\sigma} = \{j \mid a_j \in A \wedge \operatorname{sgn}(\nu_{ij}^{(\mathrm{H})} - \nu_{ij}^{(\mathrm{S})}) = \sigma\}. \tag{3.108}$$

Similarly to J_i^{σ}, let

$$s_i^{\sigma} = \sum_{j \in J_i^{\sigma}} \tau_{ij}, \tag{3.109}$$

$$s_i^{+} + s_i^{-} + s_i^{0} = 1. \tag{3.110}$$

Clearly, the problem of finding optimal τ_{ij} can be solved separately for each of the sums in (3.107).

– For the first sum

$$R_i^{+}(\tau) = -\sum_{j \in J_i^{+1}} \delta_{ij} \log \tau_{ij}, \tag{3.111}$$

conditions of the Lemma 1 hold for $J = J_i^{+1}$, $\nu_{ij} = \delta_{ij}$, and $s = s_i^{+}$. Consequently, the function $R_i^{+}(\tau)$ is minimized for

$$\tau_{ij}^{*} = \frac{s_i^{+}\, \delta_{ij}}{\sum_{l \in J_i^{+1}} \delta_{il}}, \qquad j \in J_i^{+1}. \tag{3.112}$$

Notice that the greater the sum s_i^{+} becomes, the lesser is the minimal value $R_i^{+}(\tau^{*})$.

– For the second sum

$$R_i^-(\tau) = -\sum_{j \in J_i^{+1}} (-\delta_{ij}) \log \frac{1}{\tau_{ij}}, \tag{3.113}$$

conditions of the Lemma 2 hold for $J = J_i^{-1}$, $\nu_{ij} = -\delta_{ij}$, and $s = s_i^-$. As we have shown in Lemma 2, when parameters are bounded below by some arbitrarily small $\varepsilon > 0$, the function $R_i^-(\tau)$ is minimized for

$$\tau_{ij}^*(\varepsilon) = \begin{cases} s_i^- \dfrac{1 - (|A| - |J_i^{\min}|)\varepsilon}{|J_i^{\min}|}, & \text{if } j \in J_i^{\min}; \\ s_i^- \varepsilon, & \text{if } j \in J_i^{-1} \setminus J_i^{\min}; \end{cases} \tag{3.114}$$

$$J_i^{\min} = \{ j \in J_i^{-1} \mid -\delta_{ij} = \min_{l \in J_i^{-1}} (-\delta_{il}) = -\max_{l \in J_i^{-1}} (\delta_{il}) \}. \tag{3.115}$$

Notice that, since τ_{ij} occurs in $R_i^-(\tau)$ inversed, unlike in $R_i^+(\tau)$, the lesser the sum s_i^- becomes, the lesser is the minimal value $R_i^-(\tau^*)$.

– For the indices $j \in J_i^0$, the choice of τ_{ij} is irrelevant and does not change the value of $R_i(\tau \mid z)$ regardless of the magnitude of s_i^0.

In order to combine the independent solutions (3.112) and (3.114) optimizing the separate sums, it is necessary to determine in which proportion should the probability mass be distributed between parameters belonging to J_i^{+1}, J_i^{-1}, and J_i^0. As we have seen above, for the minimal value (as a function of the bound ε) to be the smallest, s_i^+ has to be as large as possible, while both s_i^- and s_i^0, to the contrary, have to be as small as

possible. Therefore, the optimal proportion for the parameters bounded below by ε is

$$s_i^0 = |J_i^0|\,\varepsilon, \tag{3.116}$$

$$s_i^- = |J_i^{-1}|\,\varepsilon, \tag{3.117}$$

$$s_i^+ = 1 - s_i^- - s_i^0. \tag{3.118}$$

The corresponding parameters are then

$$\tau_{ij}^*(\varepsilon) = \begin{cases} \dfrac{\delta_{ij}}{\sum_{l \in J_i^{+1}} \delta_{il}}\big(1 - (|A| - |J_i^{+1}|)\varepsilon\big), & \text{if } j \in J_i^{+1}; \\ \varepsilon, & \text{if } j \in J_i^{-1} \cup J_i^0. \end{cases} \tag{3.119}$$

Passing to the limit for $\varepsilon \to 0$, we finally obtain the parameters that deliver minimum to the function $R_i(\tau \mid z)$:

$$\begin{aligned} \tau_{ij}^* = \lim_{\varepsilon \to 0} \tau_{ij}^*(\varepsilon) &= \begin{cases} \dfrac{\delta_{ij}}{\sum_{l \in J_i^{+1}} \delta_{il}}, & \text{if } j \in J_i^{+1}; \\ 0, & \text{if } j \in J_i^{-1} \cup J_i^0; \end{cases} \\ &= \frac{\max\{0, \delta_{ij}\}}{\sum_{a_j \in A} \max\{0, \delta_{ij}\}} = \frac{\mu_{ij}}{\mu_i}. \end{aligned} \tag{3.120}$$

□

3.6. Generalization for Multiple Base Messages

Throughout this chapter we have considered the method for a single arbitrary base message z that is chosen beforehand. However, with minor modifications, the presented reasoning holds for the same criteria but averaged over multiple base messages $z_l \in Z$. Indeed, for both approaches described in section 3.4 resulting in stochastic optimization,

the only change averaging over Z makes is that the variable z, as well as x, runs over a sample on iterations:

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \gamma_t W^{(\kappa_{k(t)})}(z_{l(t)}, x_{k(t)})\, g(x_{k(t)} \mid z_{l(t)}, \tau(w^{(t)})) \\ \cdot n_i(x_{k(t)} \mid z_{l(t)}) \big(\widehat{\tau}_{ij}(x_{k(t)} \mid z_{l(t)}) - \tau_{ij}(w^{(t)})\big). \tag{3.121}$$

The same is true for the likelihood-based approach discussed in section 3.5. If the criterion (3.82) is averaged over a set of base messages Z, the order of summation can be seamlessly changed so that the new outer sum over $z_l \in Z$, together with the sum over $u_k \in U$, is taken before the sum over $c_i \in A^K$ and $a_j \in A$. Consequently, the resulting objective function takes the same form (3.84) but for

$$\nu_{ij}^{(\kappa)} = \frac{1}{|Z|\,|T^{(\kappa)}|} \sum_{z_l \in Z} \sum_{u_k \in T^{(\kappa)}} \frac{n_{ij}(\psi(z_l, x_k))}{n_i(\psi(z_l, x_k))}. \tag{3.122}$$

CHAPTER IV

EVALUATION

4.1. Methodology

In order to validate the method proposed in this research, first we implemented the entropy classifier for the problem of spam filtering. Following the definition of the problem given in section 3.1.4, our implementation uses the algorithm of prediction by partial matching (PPM) to learn the finite memory Markov models for each of the two classes, and then makes classifying decisions depending on for which class entropy per character is the minimal. We also implemented all three of the algorithms proposed in sections 3.4.1, 3.4.2, and 3.5.

Our numerical experiments were organized as follows. For each run of evaluation, first, a combined sample T of both legitimate ($T^{(\mathrm{H})}$) and spam ($T^{(\mathrm{S})}$) messages was drawn out of the SpamAssassin public corpus (Apache SpamAssassin Project, 2005). Each message in $x_k \in T$ was accompanied with the true class labelling $\kappa_k \in \{\mathrm{H}, \mathrm{S}\}$.

The sample T was additionally temporarily split at random in proportion seven to three into the training and testing samples, respectively. The former was used to train the classifier, the latter was used to ensure that performance of the classifier is within the expected boundaries (as compared, for example, to (Bratko, Cormack, et al., 2006)). All of the spam messages in T that were recognized as such according to the obtained class parameters $\theta^{(\mathrm{H})}$ and $\theta^{(\mathrm{S})}$, were remembered and declared to be the set of base messages Z.

Then, our algorithms (3.52), (3.69), and (3.104) were run on the combined sample T in order to obtain transformation parameters $\tau^{(\mathrm{E})}$, $\tau^{(\mathrm{P})}$, and $\tau^{(\mathrm{L})}$, correspondingly. The

first two algorithms based on the stochastic optimization were repeatedly run over all pairs $(z_l, x_k) \in Z \times T$, where the index k was incremented first. To control the convergence, after each pass over T (i.e. every $|T|$ iterations), the value of the criterion function corresponding to the current algorithm was estimated using a ten percent subsample of $Z \times T$. This estimation together with the total number of iterations performed by the moment were used to make a stopping decision.

Once in a several passes over T (between $|T|$ and $10|T|$ iterations, depending on the size of the problem), the current parameters $\tau^{(t)} = \tau(w^{(t)})$ were supplied to the Markov chain generator. For each base message $z \in Z$, the generator produced a continuation stream of characters distributed according to the distributions $g(x, \tau^{(t)})$ that were stopped when the string $\widetilde{x}$ of characters produced so far was enough to get the transformed message $u = \psi(z, \widetilde{x}) = z\widetilde{x}$ past the classifier's spam filter. If the length of $\widetilde{x}$ exceeded $20 \cdot |z|$, the generator was forcefully stopped. This way, for each z, a thousand of continuations $\widetilde{x}$ were generated to estimate a secondary evaluation measure, the average length of $\widetilde{x}$ required to make z legitimate to the classifier.

The third algorithm (3.104) required less work since it provided the analytical solution as long as the values μ_{ij} were calculated. To do so, a single pass of averaging $n_{ij}(\psi(z_l, x_k))$ over the samples $Z \times T^{(\mathrm{H})}$ and $Z \times T^{(\mathrm{S})}$ was done. After that, the same generation procedure described above was done, so there was an auxiliary measure for comparing this algorithm with the other two and the baseline strategy.

The role of a baseline strategy in our experiments was played by the same generation procedure called for the vector of parameters $\tau^{(\mathrm{H})} = \theta^{(\mathrm{H})}$ that were estimated during the training of the classifier on the sample of legitimate messages $T^{(\mathrm{H})}$. That same vector $\theta^{(\mathrm{H})}$ also served as an initial estimate for the stochastic optimization.

TABLE 4.1. Performance of the classifier on the full dataset.

True class	Classified as Ham	Classified as Spam
Ham	68.0% (1645)	0.5% (13)
Spam	1.5% (36)	30.0% (725)

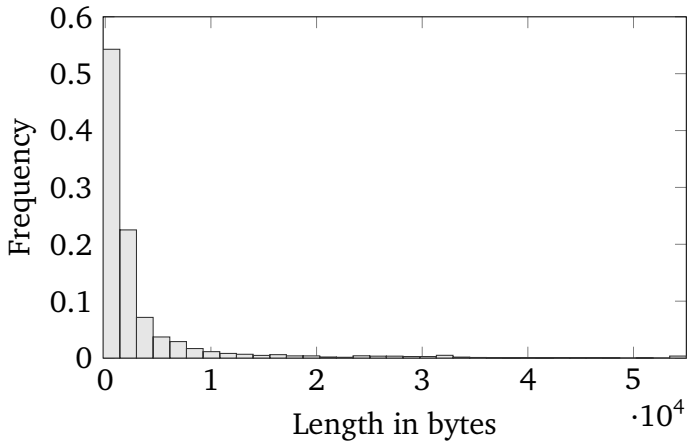

FIGURE 4.1. Histogram of lengths of all messages in the full dataset.

4.2. Results

Due to limited computational resources, during all evaluation runs, Markov models' memory was fixed to be three characters for the classifier as well as the adversary. In practice, entropy-based spam filters demonstrate the best performance for orders of Markov models between six and eight characters (Bratko, Cormack, et al., 2006)). However, even for $K = 3$ our implementation of the classifier based on the algorithm of prediction by partial matching has error rate of approximately 2% on the SpamAssassin dataset. Table 4.1 shows statistics of one such run when all 6046 bodies of email messages were split into 3627 training and 2419 testing messages. The distribution of lengths of the messages is shown in Figure 4.1.

For the order $K = 3$, the space of parameters τ, representing conditional probabilities of a one-byte character given a context of at most K another one-byte

characters, is bounded by

$$\sum_{k=0}^{K} |A|^{k+1} = 256^1 + 256^2 + 256^3 + 256^4 \approx 2^{32}. \tag{4.1}$$

The total number of character-context combinations for $K = 3$ that actually occur in all messages from the SpamAssassin dataset is approximately 524 000.

To avoid memory pressure and achieve faster convergence, the algorithms (3.52) and (3.69) requiring stochastic optimization, were run on a series of small subsets of the original dataset. Each time, approximately one percent of messages were sampled at random from the full dataset. Let us present evaluations for a typical such run on a 1% dataset done for the three algorithms, as it was described in the previous section.

The failure rate of the chosen concatenation-based transformation was zero for all spam messages and parameters τ obtained from all three algorithms as well as the Ham baseline $\tau^{(\mathrm{H})}$. That is, it was possible to generate an appendix x_k for each base spam message z_l such that their concatenation $u_k = \psi(z_l, x_k) = z_l x_k$ was classified as legitimate. For this reason, to compare performance for different parameters, we used a supplementary index of the ratio $|u_k|/|z_l|$ between the lengths of each transformed message. Note that none of the methods proposed in this work was constructed to directly optimize this length ratio.

Figures 4.2, 4.3, and 4.4 depict distributions of length ratios averaged over transformation appendices x_k generated according to the parameters $\tau^{(\mathrm{E})}$, $\tau^{(\mathrm{P})}$, $\tau^{(\mathrm{L})}$ that were optimized for the entropy-based, probability-based, and likelihood-based criteria, respectively. As it is easy to see from the cumulants provided to the right of each histogram, the algorithms using probability-based and likelihood-based criteria are more preferable to the one using entropy-based criterion in terms of the length ratio. However, comparing these graphs with Figure 4.5, showing the histogram and cumulant

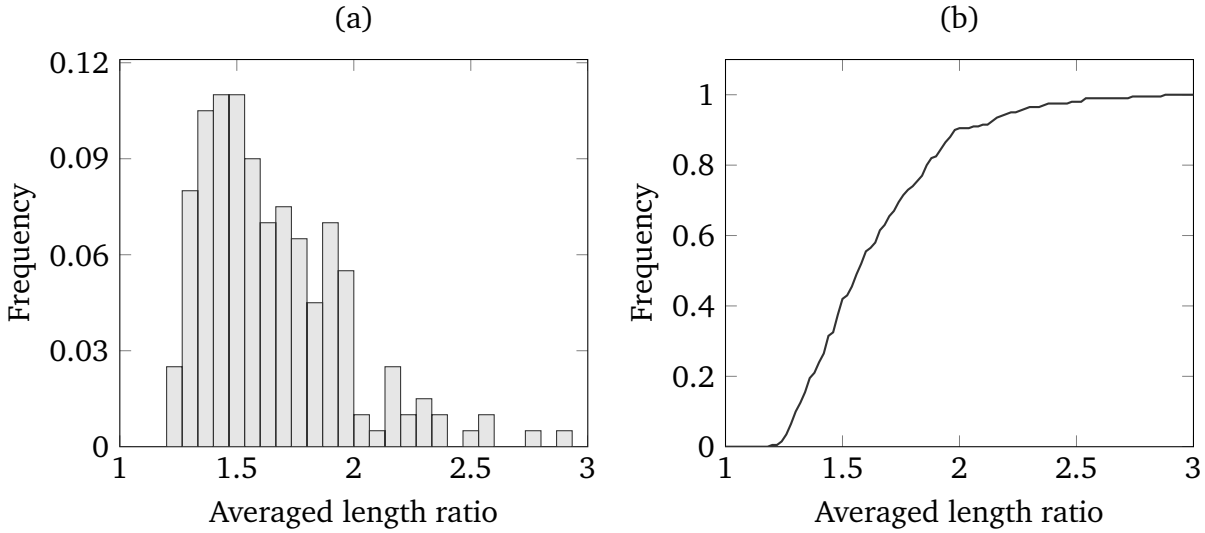

FIGURE 4.2. (a) Histogram of the length ratio $|\psi(z_l, x_k)|/|z_l|$ averaged over appendices x_k generated using the parameters $\tau^{(\mathrm{E})}$ optimized for the *entropy-based criterion* (3.48) on a *1% dataset*; (b) its cumulant.

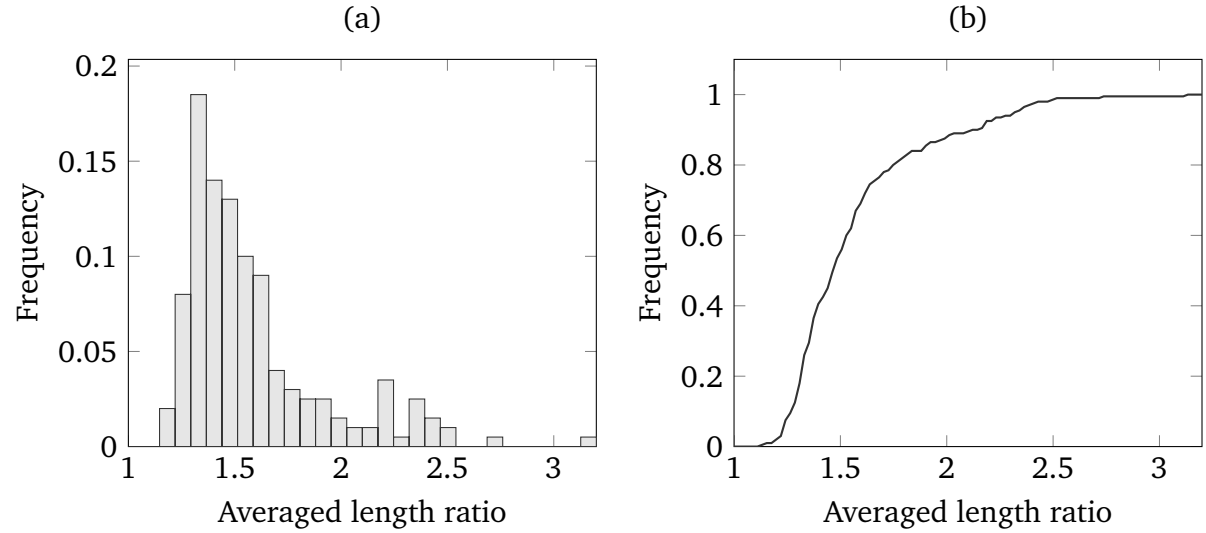

FIGURE 4.3. (a) Histogram of the length ratio $|\psi(z_l, x_k)|/|z_l|$ averaged over appendices x_k generated using the parameters $\tau^{(\mathrm{P})}$ optimized for the *probability-based criterion* (3.60) on a *1% dataset*; (b) its cumulant.

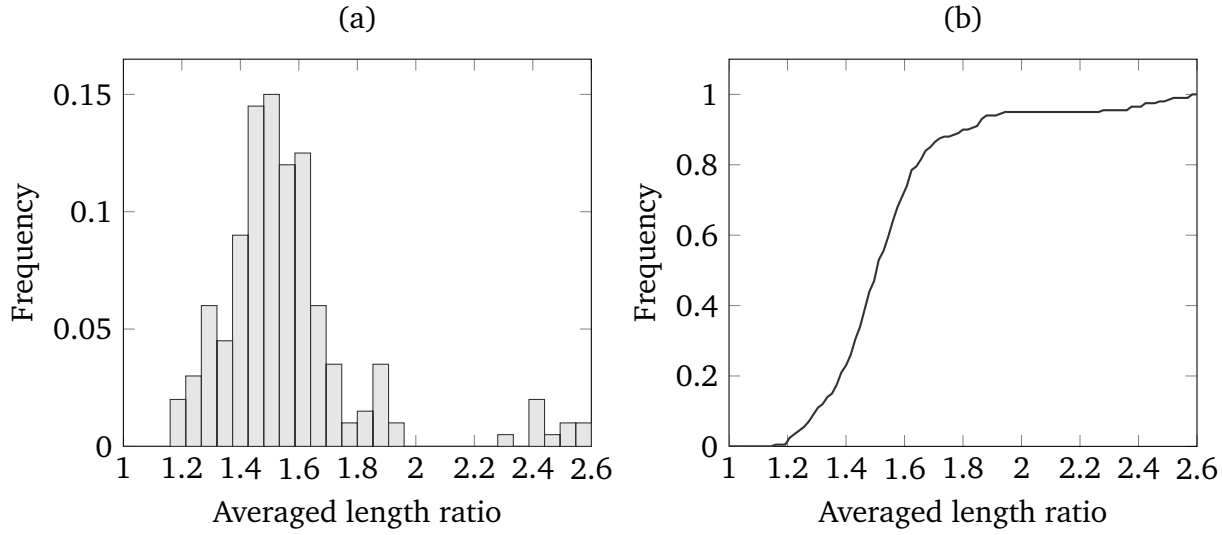

FIGURE 4.4. (a) Histogram of the length ratio $|\psi(z_l, x_k)|/|z_l|$ averaged over appendices x_k generated using the optimal parameters $\tau^{(\mathrm{L})}$ (3.104) for the *likelihood-based criterion* (3.84) on a *1% dataset*; (b) its cumulant.

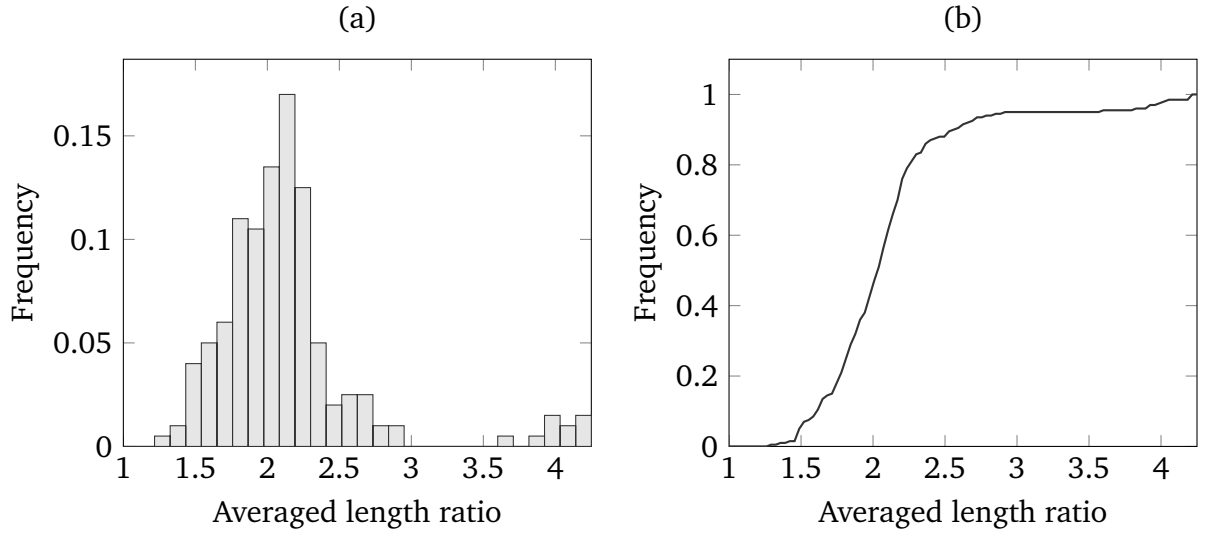

FIGURE 4.5. (a) Histogram of the length ratio $|\psi(z_l, x_k)|/|z_l|$ averaged over appendices x_k generated using the *baseline parameters* $\tau^{(\mathrm{H})}$ estimated from $\theta^{(\mathrm{H})}$ on a *1% dataset*; (b) its cumulant.

TABLE 4.2. Summary statistics for the performance of different generation parameters on 1% datasets.

Index	Optimization based on			Ham
	Entropy	Probability	Likelihood	baseline
Failure rate	0%	0%	0%	0%
Averaged length ratio				
Minimum	1.21	1.15	1.16	1.32
5% quantile	1.29	1.26	1.26	1.52
Median	1.58	1.49	1.52	2.06
Mean	1.65	1.59	1.57	2.13
95% quantile	2.26	2.35	2.13	3.27
Maximum	2.89	3.14	2.61	4.27

for the parameters $\tau^{(\mathrm{H})}$, all three techniques provide a noticeably better performance compared to the baseline of generating Ham-like appendix. A short summary of the statistics from these figures is given in Table 4.2.

Figure 4.6 shows several examples of generated transformation texts for a few short spam messages from the dataset. Each of the original spam messages (typeset on white background) is followed by strings produced by the generation procedure according to optimal parameters (highlighted with gray background). Any of the presented appendices is sufficient to make the corresponding spam message look as a legitimate text, and cannot be shortened without changing the class of the transformed message back to spam.

For the purpose of comparing the systems of probability distributions resulting from the aforementioned attack techniques, we measured the the difference between two probability distributions defined by parameters $\tau_i^{(1)}$ and $\tau_i^{(2)}$ for each context $c_i \in A^K$. To do so, we used the distance function

$$d(\tau_i^{(1)}, \tau_i^{(2)}) = \sum_{a_j \in A} |\tau_{ij}^{(1)} - \tau_{ij}^{(2)}|. \tag{4.2}$$

```
  Hi we are luke's secret following we
↪love luke fictitious!

  We are also your long lost friend! Hi

  This email has nothing to do with
↪lukefictitious.com

  We wil be putting up our very own fan
↪site soon
  and wanted to let you know in advance!

  Have a beautifull day!
```

```
Joseph
```

```
Regard E
```

```
__________
Exm
```

```
  Hey,  I just wanted to tell you about a
↪GREAT website.
↪http://www.metrojokes.com    Features
↪lots of jokes!  Extremly unique
↪features and classified in categories.
↪I appriciate your time.

  Thank you
```

```
your loved one out of your typical diam
```

```
 - No, the out their until your typi
```

```
from you're decent? I
```

```
ass, but the
sun doesn't
```

```
fat able to be and wonderful
>>
```

```
  (suddenlysusan@Stoolmail.zzn.com) on
↪Tuesday, July 30, 2002 at 17:07:56
  : Why Spend upwards of $4000 on a DVD
↪Burner when we will show you an
↪alternative that will do the exact same
↪thing for just a fraction of the cost?
↪Copy your DVD's NOW.
```

```
This?

I
```

```
It spamassin-dev
```

```
--
"If you."
```

```
This a multi-part, surround you're du
```

```
This a must IM. Build
searcharself with think?
```

```
This a deady to be looking of some
↪merge.net
```

```
  DON'T MISS OUT ON AN AMAZING BUSINESS
↪OPPORTUNITY AND WEIGHT LOSS PRODUCT!
  PLEASE VISIT
↪www.good4u.autodreamteam.com
  THERE IS NO OBLIGATION
  AND IT'S WORTH A LOOK!
```

```
Remore
> OK guys -- I r
```

```
md: rules.
>
> with smart_0088
```

```
 [evel

Yet emailename="smime
```

```
> OK guys -- I reck_f
```

```
>
> BSMTP-support people
> OK guy
```

FIGURE 4.6. Examples of original spam messages z_l (white background) and several appendices x_k corresponding to each z_l that are generated using parameters $\tau^{(\mathrm{E})}$ optimized on a 1% dataset (gray background).

TABLE 4.3. Summary statistics for the performance of the generation parameters optimal for the likelihood-based criterion on the full datasets.

Index	Likelihood optimization	Ham baseline
Failure rate	0%	0.16%
Averaged length ratio		
Minimum	1.001	1.001
5% quantile	1.068	1.248
Median	1.184	1.704
Mean	1.232	1.976
95% quantile	1.594	4.101
Maximum	2.929	8.498

Figure 4.9 features distributions of distances $d(\tau_i^{(1)}, \tau_i^{(2)})$ for all pairs of parameters $\tau_i^{(1)}, \tau_i^{(2)} \in \{\tau_i^{(\mathrm{E})}, \tau_i^{(\mathrm{P})}, \tau_i^{(\mathrm{L})}, \tau_i^{(\mathrm{H})}\}$ that were fitted on a 1% dataset using all three algorithms, as well as the baseline Ham parameters. The minimal distance of $d(\tau_i^{(1)}, \tau_i^{(2)}) = 0$ indicates that distributions $\tau_i^{(1)}$ and $\tau_i^{(2)}$ are identical. The maximal distance of $d(\tau_i^{(1)}, \tau_i^{(2)}) = 1$ corresponds to greatest difference between $\tau_i^{(1)}$ and $\tau_i^{(2)}$.

Since the likelihood-based algorithm (3.104) does not have as high computational requirements as the other two algorithms resorting to stochastic optimization, it was possible for us to run it on the full dataset. Resulting distributions of the length ratio for the parameters optimal for the likelihood-based criterion and the baseline Ham parameters are presented in Figures 4.7 and 4.8 in a similar fashion to the case of a 1% dataset shown above. Table 4.3 lists the same five quantiles of the length ratio as well as its mean values. Comparing these statistics with the ones in Table 4.2, we can conclude that the breach between the Ham-like generation and the likelihood-based algorithm is even greater ($\approx 70\%$ vs. $\approx 50\%$).

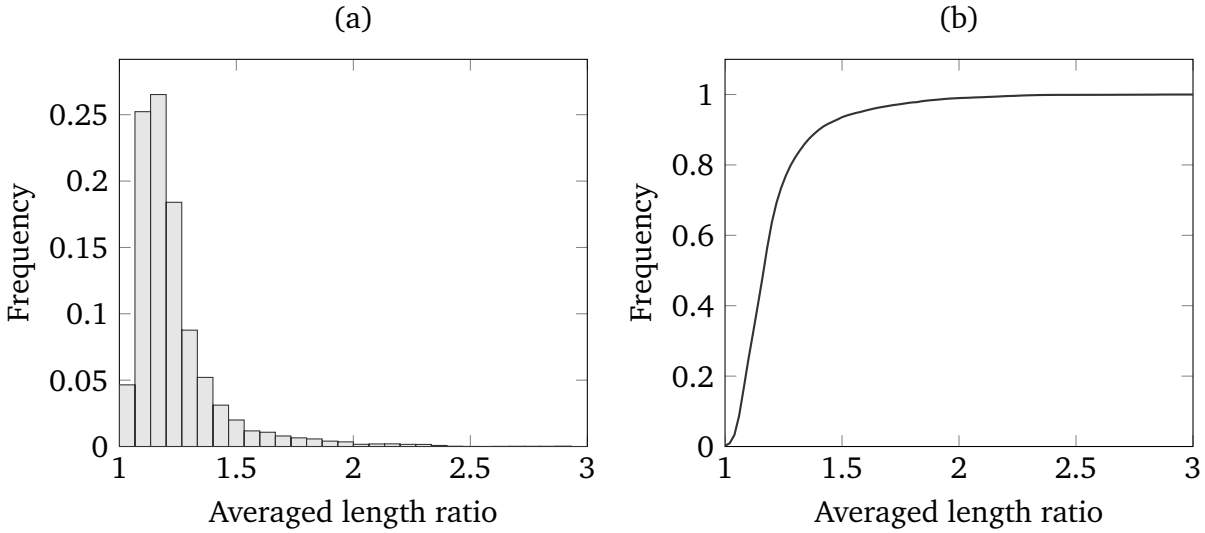

FIGURE 4.7. (a) Histogram of the length ratio $|\psi(z_l, x_k)|/|z_l|$ averaged over appendices x_k generated using the optimal parameters $\tau^{(\mathrm{L})}$ (3.104) for the *likelihood-based criterion* (3.84) on the *full dataset*; (b) its cumulant.

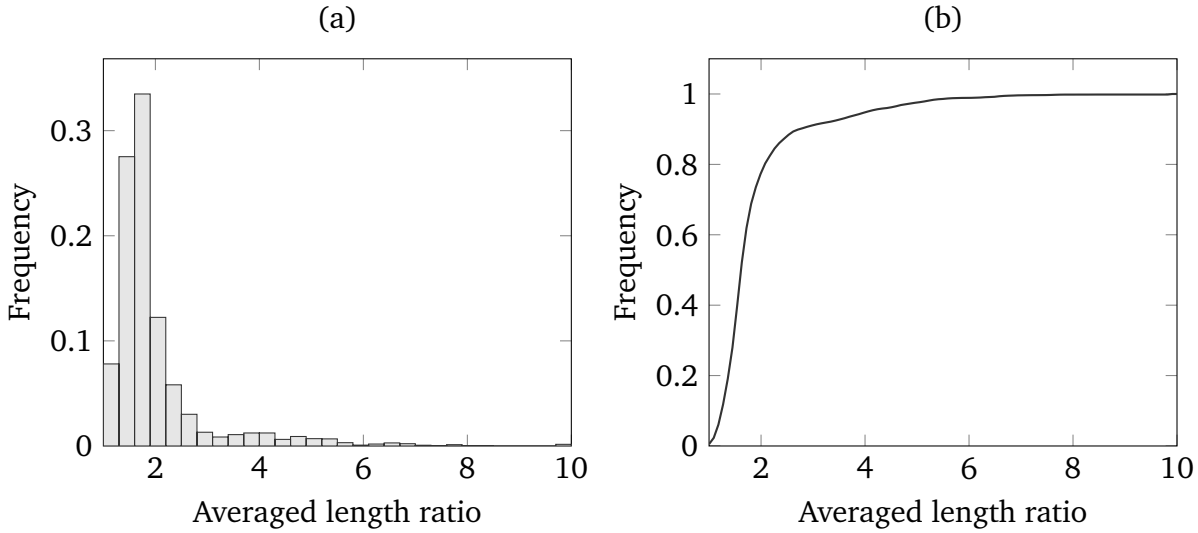

FIGURE 4.8. (a) Histogram of the length ratio $|\psi(z_l, x_k)|/|z_l|$ averaged over appendices x_k generated using the *baseline parameters* $\tau^{(\mathrm{H})}$ estimated from $\theta^{(\mathrm{H})}$ on the *full dataset*; (b) its cumulant.

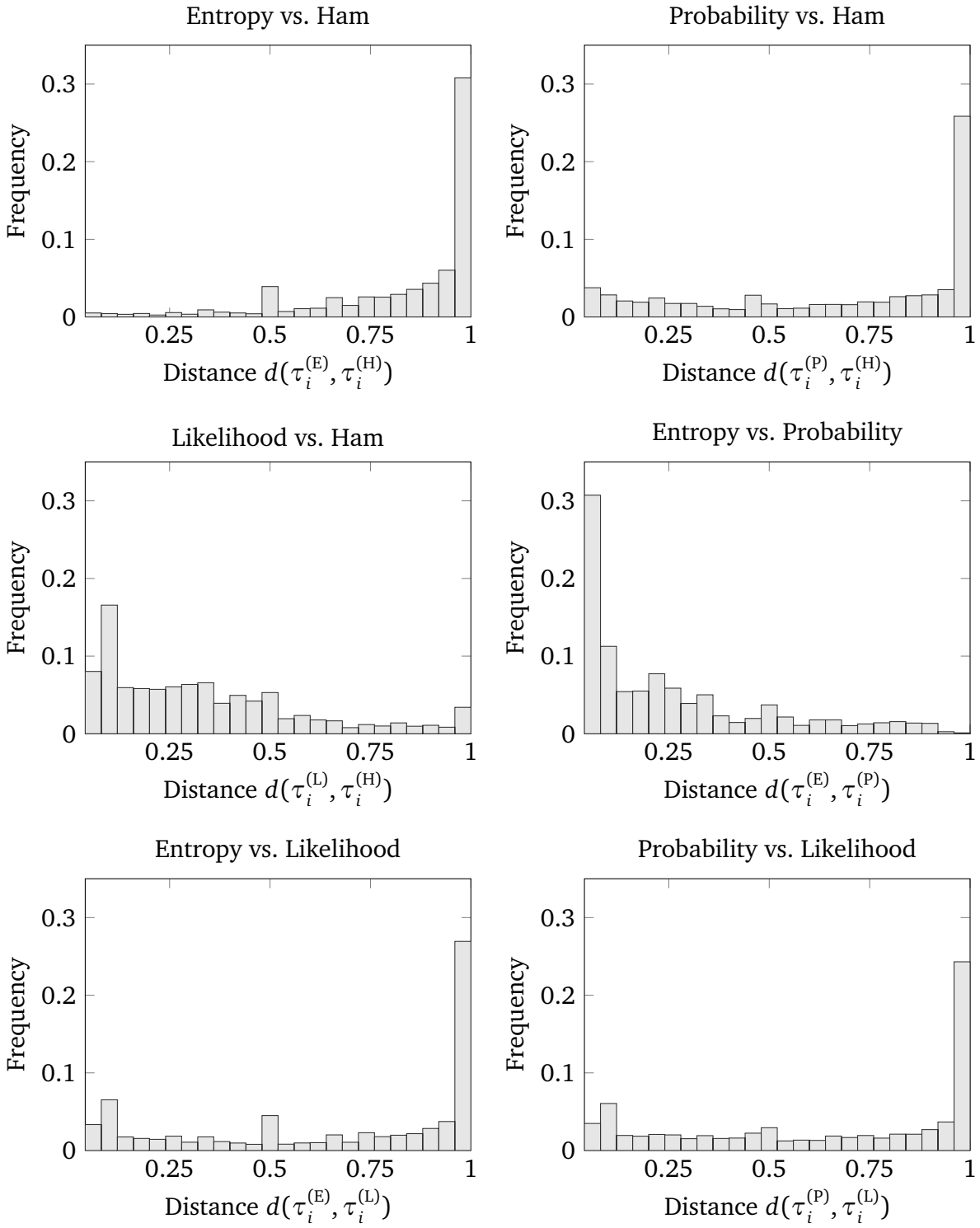

FIGURE 4.9. Histograms of the distances between parameters obtained through different adversary algorithms as well as the baseline parameters.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

We introduced three formalizations of possible adversarial objectives for classifiers using cross-entropy as the deciding criterion. Each of the three approaches has proved its efficiency as compared to the baseline approach of using the probability distributions estimated only on legitimate messages to define a transformation source. The third technique showed itself as the most efficient of three, both in terms of transformation and computational requirements. Although the first two techniques have same implementation difficulties, after appropriate calibration, they showed comparable performance. Together, all three methods have shown the feasibility of statistically attacking compression-based classifiers using relatively limited extent of transformation (for the SpamAssassin corpus, on average, approximately 20% of original message's length is to be generated and appended).

Future work includes three directions of possible extension of this research. To begin with, it is of interest to explore how methods of parametrized optimization, examples of which are considered in this work, compare with other algorithms allowing to optimize the contents of transformation texts directly on a per character basis. This problem setting can potentially increase the number of different criterion functions that can be considered to formalized the adversary problem. The methods that might be applied in that setting include Markov chain Monte Carlo methods, genetic algorithms, simulated annealing, and others.

Another promising direction consists in analyzing the dynamics of classifier-adversary system for the particular case of entropy-based classification considered in this thesis. Although it is hard to attack this problem in general, it might be feasible to

derive some useful properties for the specific algorithms that we have discussed in our work.

Last but not least, it is important to research ways of improving performance of compression-based classifiers that might be possible given the knowledge of potential adversary attacks.

REFERENCES CITED

Apache SpamAssassin Project. (2005). The SpamAssassin public mail corpus. Available at https://spamassassin.apache.org/publiccorpus/.

Biggio, B., Nelson, B., & Laskov, P. (2011). Support vector machines under adversarial label noise. In *Journal of Machine Learning Research: Workshop and Conference Proceedings* (Vol. 20, pp. 97–112). MIT.

Biggio, B., Nelson, B., & Laskov, P. (2012). Poisoning attacks against support vector machines. In *Proceedings of The 29th International Conference on Machine Learning*. Omnipress.

Bratko, A., Cormack, G. V., Filipič, B., Lynam, T. R., & Zupan, B. (2006). Spam filtering using statistical data compression models. *The Journal of Machine Learning Research*, *7*, 2673–2698.

Bratko, A., Filipič, B., & Zupan, B. (2006). Towards practical PPM spam filtering: Experiments for the TREC 2006 Spam Track. In *Proceedings of the 15th Text REtrieval Conference (TREC 2006)*. Gaithersburg, MD.

Cleary, J. G. & Teahan, W. J. (1997). Unbounded length contexts for PPM. *The Computer Journal*, *40*(2 and 3), 67–75.

Cleary, J. G. & Witten, I. H. (1984). Data compression using adaptive coding and partial string matching. *IEEE Transactions on Communications*, *32*(4), 396–402.

Cormack, G. V. & Horspool, R. N. S. (1987). Data compression using dynamic Markov modelling. *The Computer Journal*, *30*(6), 541–550.

Frank, E., Chui, C., & Witten, I. H. (2000). Text categorization using compression models. In *Proceedings of DCC-00, IEEE Data Compression Conference* (pp. 200–209). Snowbird, US: IEEE Computer Society Press, Los Alamitos, US.

Goodman, J., Heckerman, D., & Rounthwaite, R. (2005). Stopping spam. *Scientific American*, *292*(4), 42–49.

Lowd, D. & Meek, C. (2005a). Adversarial learning. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 641–647). ACM.

Lowd, D. & Meek, C. (2005b). Good word attacks on statistical spam filters. In *Proceedings of the Second Conference on Email and Anti-Spam*. Palo Alto, CA.

Moffat, A. (1990). Implementing the PPM data compression scheme. *IEEE Transactions on Communications*, *38*(11), 1917–1921.

Robbins, H. & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, *22*, 400–407.

Teahan, W. J. (1995). Probability estimation for PPM. In *New Zealand Computer Science Research Students' Conference, 1995*. University of Waikato, Hamilton, New Zealand.