Automated Attacks on Compression-Based Classifiers

Igor Burago
Department of Computer Science
University of California, Irvine
iburago@uci.edu

Daniel Lowd
Department of Computer
and Information Science
University of Oregon
lowd@cs.uoregon.edu

ABSTRACT

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 paper, 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 11% of the original length of the message.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning

Keywords

Adversarial machine learning, Evasion attacks, Compression-based classifiers, Prediction by partial mapping (PPM), Text classification, Spam filtering

1. INTRODUCTION

Machine learning is now being used to detect many kinds of malicious activity, including online auction fraud [11], fake reviews [1, 21], social network spam [4, 33], email spam [7, 16, 31], comment spam [23], and malware [3, 24, 29]. In response, adversaries continually modify their behavior to avoid being detected. As a result, machine learning models that work well on historical data may work very poorly in practice as adversaries find and exploit their weaknesses [17, 38]. For example, spammers regularly modify their spam messages to make them appear less spammy to a classifier while remaining as persuasive as possible.

AISec'15, October 16 2015, Denver, Colorado, USA

ACM 978-1-4503-3826-4/15/10 ...$15.00
DOI: http://dx.doi.org/10.1145/2808769.2808778

In adversarial domains such as these, we would prefer to use classifiers that are harder to evade. Previous analysis has shown that linear classifiers with independent features are relatively easy to defeat [20, 25, 26]. When the parameters of the classifier are known, the adversary can identify the most influential features. For classifying spam messages, many of the features are individual words. By removing the "spammiest" words and adding the "hammiest" words, a spammer can disguise an email with a relatively small number of modifications. When the parameters of the classifier are unknown, good features to modify can be guessed using background knowledge [26], the classifier's training data or similar data from the same distribution [5, 36], or through direct interaction with the classifier [25]. These attacks are surprisingly effective against realistic spam filters. Nelson et al. [28] demonstrate that convex-inducing classifiers with continuous features are also vulnerable, at least in theory.

Given the vulnerability of linear classifiers to these attacks, researchers have proposed using other classifiers instead. Jorgensen et al. [22] introduce a multiple instance learning strategy designed to be more robust to "good word" attacks, in which only "hammy" words are added. However, it is not clear how well this idea generalizes to other problems, other sets of features, or other evasion techniques, including removing "spammy" words. A more general alternative is to use a classifier that exploits sequence information to produce predictions that are more accurate and harder to defeat. Bratko et al. [7] propose compression-based classifiers in general and prediction by partial mapping (PPM) in particular as a more effective and robust approach to spam classification.

Compression-based classifiers represent the data as a sequence of overlapping features rather than as a set of independent features. This means that the effect of adding a word to a message could depend on where in the message it is added — the same word could be more or less effective at the beginning, end, or somewhere in the middle of the message. Thus, evading a compression-based classifier is more complicated than evading a linear classifier. However, there has not yet been a systematic study of how compression-based classifiers could be evaded.

In this paper, we formally define the problem of evading a compression-based classifier, present three methods for solving the problem, and evaluate them on a standard email spam dataset. All three methods are based on a single key assumption that an adversary has an access to a sample of legitimate and spam messages that has the same statistical

properties as the sample that was used to train the classifier. Our methods find a *distribution* of sequences that can be appended to help disguise any instance. This is more efficient than trying to disguise each instance separately. It is also much less brittle than appending a single, fixed sequence, which could be quickly learned by the classifier.

Empirically, we find that the median spam can be disguised by increasing its length by just 11%. This suggests that, like linear classifiers, compression-based classifiers are indeed vulnerable to evasion techniques.

2. RELATED WORK

Research in adversarial machine learning has shown that linear spam filters are succeptible to "good word attacks" in which an attacker evades a spam filter by appending non-spammy words to a spam email [26]. This attack can be generalized into the problem of finding a minimal or near-minimal set of changes to transform a malicious instance into one that is labeled as innocent [25]. In follow-up work, other researchers have developed efficient evasion attacks against convex-inducing classifiers [28] and certain combinations of linear classifiers [32].

Robust learning algorithms can reduce the vulnerability of linear classifiers, so that attackers need to make larger modifications in order to evade detection [9,10,15,35]. Other researchers have proposed non-linear spam filters that are specifically robust to good word attacks, such as multiple instance learning [22] and compression-based classifiers [8]. The authors demonstrate that these methods are robust to good word attacks, but do not fully explore what other attacks might be possible against them.

Compression-based text classification rests on the assumption that two documents are likely to be of the same kind if they compress well together. This approach has been widely studied and applied for categorizing different types of text [6–8,14,18,19,23,39]. We focus on entropy-based classifiers that use cross entropy as a similarity measure. However, our work is mostly agnostic to how the classifier's parameters are estimated. For evaluation, we use the prediction by partial matching (PPM) algorithm [12,13,27,34], which has proven effective for a special case of text classification — spam filtering [7,8]. While a significant amount of effort has been applied to studying the effectiveness of compression models on selected applications of text classification, there is still no complete understanding of how robust such algorithms are to different kinds of adversarial noise.

3. COMPRESSION-BASED CLASSIFIERS

3.1 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 the 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.

3.2 Finite-Memory Markov Model

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

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

where x_l is the l-th character of the string x, and x_k^l is the substring of x from the k-th up to 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:

$$f^{(\kappa)}(x) = f(x, \theta^{(\kappa)}) = \prod_{l=1}^{|x|} \theta^{(\kappa)}_{i(x_1^{l-1}),\, j(x_l)}, \tag{2}$$

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 modeled 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 (2), we can rewrite it for our convenience as

$$f(x, \theta) = \prod_{l=1}^{|x|} \theta_{i(x_{l-K}^{l-1}),\, j(x_l)} = \prod_{\substack{c_i \in A^K \\ a_j \in A}} \theta_{ij}^{n_{ij}(x)}, \tag{3}$$

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

$$\sum_{\substack{c_i \in A^K \\ a_j \in A}} n_{ij}(x) = |x|, \quad \text{and} \quad \sum_{a_j \in A} \theta_{ij} = 1 \text{ for all } c_i \in A^K. \tag{4}$$

This way, any string x and any class κ are viewed as sets of overlapping $(K+1)$-grams with frequencies $n_{ij}(x)$ and conditional probabilities θ_{ij}, respectively.

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

$$g(x, \tau) = \prod_{\substack{c_i \in A^K \\ a_j \in A}} \tau_{ij}^{n_{ij}(x)}. \tag{5}$$

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 $\theta^{(\kappa)}$.

3.3 Classification Problem

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

$$R(\theta) = \mathbf{E}_\xi\big[r(\xi, \theta)\big] \to \min_\theta, \tag{6}$$

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 (6) 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 the input source. Generally, probability $g(x)$ is supposed to be unknown for all classes. For this reason, a version of the problem (6) for empirical averaging is considered:

$$\widehat{R}(\theta) = \sum_{x_k \in T} r(x_k, \theta) \to \min_{\theta}, \tag{7}$$

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:

$$q(x, \theta) = r(x, \theta^{(\mathrm{H})}) - r(x, \theta^{(\mathrm{S})}), \tag{8}$$

$$\kappa(x) = \begin{cases} \mathrm{H}, & \text{if } q(x, \theta) < \alpha^{(\mathrm{H})}; \\ \mathrm{S}, & \text{if } q(x, \theta) \geq \alpha^{(\mathrm{S})}. \end{cases} \tag{9}$$

In case of $\alpha^{(\mathrm{H})} \leq q(x, \theta) < \alpha^{(\mathrm{S})}$, additional measures are needed to decide the class (e.g., 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 the number of type I and type II errors.

3.4 Entropy Classification

Let us consider the measure

$$r(\xi, \theta) = -\frac{1}{|\xi|} \log f(\xi, \theta). \tag{10}$$

Then, as it is obvious from the above definitions, the general criterial function (6) specializes to the cross entropy

$$\begin{aligned} R(\theta) = H(\theta) &\equiv -\mathbf{E}_{\xi} \left[\frac{1}{|\xi|} \log f(\xi, \theta) \right] \\ &= -\sum_{x \in X} \frac{1}{|x|} g(x) \log f(x, \theta) \to \min_{\theta}. \end{aligned} \tag{11}$$

We will refer to this specialization of the problem (6) as the *classifier problem*. Similarly, empiric version (7) becomes

$$\widehat{R}(\theta) = \widehat{H}(\theta) \equiv -\sum_{x_k \in T} \frac{1}{|x_k|} \log f(x_k, \theta) \to \min_{\theta}, \tag{12}$$

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

Decision rule (9) can be rewritten as

$$q(x, \theta) = \frac{1}{|x|} \log \frac{f^{(\mathrm{S})}(x)}{f^{(\mathrm{H})}(x)}, \tag{13}$$

$$\kappa(x) = \begin{cases} \mathrm{H}, & \text{if } q(x, \theta) < \alpha; \\ \mathrm{S}, & \text{if } q(x, \theta) \geq \alpha. \end{cases} \tag{14}$$

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

$$f(x, \theta) \propto \frac{g(x)}{|x|} \tag{15}$$

is an exact solution of the problem (11). Because 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 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 works as a simple frequency estimator, setting

$$\theta_{ij} \approx \frac{N_{ij}}{N_i}, \text{ for } N_{ij} = \sum_{x \in T} n_{ij}(x), \; N_i = \sum_{a_j \in A} N_{ij}. \tag{16}$$

For versions of PPM estimators and the details of their implementation, see [12, 13, 27, 34].

4. ADVERSARIAL ATTACKS

4.1 Problem Definition

As we have seen above, in the classifier problem (11), the goal was to find an optimal statistical model $f(x, \theta)$ for messages of some class, given a fixed input source induced by probabilities $g(x)$ observed through a sample T. More strictly, the function $g(x)$ was fixed (although unknown), while the probability distribution $f(x, \theta)$ was known up to the vector of parameters θ.

It is 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 favorable for certain classification outcome. In this setting, $g(x)$ becomes the function in question, while $f(x, \theta)$ is fixed through known 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 after a certain combining transformation $\psi(z, x_t)$ they satisfy 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 will assume that 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. These strings are then used to obtain new messages $u_t = \psi(z, x_t(\tau))$ from a given base 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 any particular transformation, for illustration purposes, we will be using concatenation as the transformation ψ.

The objective of the inverse problem has the same form:

$$G(\tau) \equiv -\sum_{x\in X} \frac{1}{|x|} g(x,\tau) \log f(x) \to \min_{\tau}, \tag{17}$$

but with the optimization being 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 (11), it is well known that, in the non-parametrical form, the inverse problem (17) also has an analytical solution. Any function $g(x)$ such that

$$\sum_{x\in X_{f_{\max}}} g(x) = 1, \text{ where } X_{f_{\max}} = \arg\max_x \frac{f(x)}{|x|}, \tag{18}$$

$$g(x) = 0 \text{ for all } x \in X \setminus X_{f_{\max}}, \tag{19}$$

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 (17) is

$$\widehat{G}(\tau) \equiv \sum_{x_k\in T} \frac{1}{|x|} g(x_k,\tau) \to \min_{\tau}, \tag{20}$$

where a sample T is obtained from the distribution $P(\xi = x) \propto \log \frac{1}{f(x)}$. Therefore, in order to approach the inference problem in the form (20), 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.

4.2 Instrumental Sampling Approach

Let us introduce new parameters w_{ij} such that

$$\tau_{ij} = \frac{\exp(w_{ij})}{\sum_{a_j\in A} \exp(w_{ij})}, \tag{21}$$

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:

$$0 < \tau_{ij} < 1, \text{ and } \sum_{a_j\in A} \tau_{ij} = 1 \tag{22}$$

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

For the new parameters, probability (5) changes to

$$g(x,\tau(w)) = \prod_{c_i\in A^K} \left(\frac{1}{Z_i} \exp\left(\sum_{a_j\in A} w_{ij} \frac{n_{ij}(x)}{n_i(x)} \right) \right)^{n_i(x)}, \tag{23}$$

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

$$n_i(x) = \sum_{a_j\in A} n_{ij}(x), \tag{24}$$

$$Z_i(w) = \sum_{a_j\in A} \exp(w_{ij}). \tag{25}$$

Since

$$\begin{aligned} \log g(x,\tau(w)) &= \sum_{\substack{c_i\in A^K \\ a_j\in A}} n_{ij}(x) \log \left(\frac{\exp(w_{ij})}{Z_i(w)} \right) \\ &= \sum_{\substack{c_i\in A^K \\ a_j\in A}} w_{ij} n_{ij}(x) - \sum_{\substack{c_i\in A^K \\ a_j\in A}} n_{ij}(x) \log Z_i(w), \end{aligned} \tag{26}$$

it is clear that

$$\begin{aligned} \frac{\partial g(x,\tau(w))}{\partial w_{lk}} &= g(x,\tau(w)) \left(n_{lk}(x) - \frac{n_l(x)}{Z_l} \exp(w_{lk}) \right) \\ &= g(x,\tau(w))\, n_l(x) \left(\widehat{\tau}_{lk}(x) - \tau_{lk}(w) \right), \end{aligned} \tag{27}$$

for

$$\widehat{\tau}_{lk}(x) = \frac{n_{lk}(x)}{n_l(x)}. \tag{28}$$

Now, consider the problem

$$\mathbf{E}_\xi \left[F(\xi, w) \right] \approx \sum_{x_k\in T} F(x_k, w) \to \min_{w}, \tag{29}$$

where both the random variable $\xi(w)$ and strings x_k from the instrumental sample T are distributed according to the probabilities $g(x,\tau(w))$. The problem (20) that has motivated us to consider this approach is a special case for

$$F(x,w) = F(x) = \frac{1}{|x|} \log \frac{1}{f(x)}. \tag{30}$$

Taking derivatives of (29) and keeping in mind (27), we obtain the following necessary condition of extremum:

$$\mathbf{E}_\xi \left[\frac{\partial F(\xi,w)}{\partial w_{lk}} + F(\xi,w)\, n_l(\xi) \left(\widehat{\tau}_{lk}(\xi) - \tau_{lk}(w) \right) \right] = 0, \tag{31}$$

for all $c_l \in A^K$, $a_k \in A$. In the case when function $F(x,w) = F(x)$ is independent of parameters, it simplifies to

$$\mathbf{E}_\xi \left[F(\xi)\, n_l(\xi) \left(\widehat{\tau}_{lk}(\xi) - \tau_{lk}(w) \right) \right] = 0. \tag{32}$$

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. Therefore, attempts to iteratively optimize (29) will turn into random walks around initial values of w_{ij}.

Moreover, for many practical generation procedures,

$$\mathbf{E}_\xi \left[\widehat{\tau}_{lk}(\xi) \right] = \tau_{lk}(w). \tag{33}$$

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 some fixed length of x, the equation (32) 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 (32) 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 (32) almost independent of F.

For these reasons, we do not regard approaching problem (17) in form (29) as promising.

4.3 Importance Sampling Approach

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

$$\mathcal{F}_D[q(u,\theta)] \equiv \mathcal{F}[q(u,\theta) \mid u = \psi(z,x),\, x \in D] \to \max_{\tau}, \quad (34)$$

where D is a set of text strings, and $\mathcal{F}(\cdot)$ is an ensemble operation defined on the domain 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 adversarial generator can also be taken as a 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. In this work, we consider the following two cases.

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

$$\mathcal{F}_{D_\tau}[q(u,\theta)] = -\sum_{x \in D_\tau} q(u,\theta) \to \max_{\tau}. \quad (35)$$

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

$$\mathcal{F}_{D_\tau}[q(u,\theta)] = \frac{1}{|D_\tau|} \sum_{x \in D_\tau} \mathbf{1}^{(\mathrm{H})}(u) \to \max_{\tau}, \quad (36)$$

where $\mathbf{1}^{(\mathrm{H})}(u) \equiv \mathbf{1}[q(u,\theta) < \alpha]$.

4.3.1 Entropy-Based Criterion

Empirical criterion (35) is equivalent to the problem

$$R(\tau \,|\, z) \equiv \sum_{x \in X} q(u,\theta)\, g(x \,|\, z,\tau) = \mathbf{E}_\xi\big[q(u,\theta)\big] \to \min_{\tau}, \quad (37)$$

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 (37) using the well-known technique of importance sampling:

$$R(\tau \mid z) = \sum_{\kappa \in \{\mathrm{H},\mathrm{S}\}} p^{(\kappa)}\, \mathbf{E}_\xi^{(\kappa)} \left[\gamma^{(\kappa)}\, q(\xi \mid z,\theta)\, \frac{g(\xi \mid z,\tau)}{p^{(\kappa)}\, f^{(\kappa)}(\xi)} \right]$$
$$= \mathbf{E}_{\xi,\kappa}\big[W^{(\kappa)}(\xi \mid z,\theta)\, g(\xi \mid z,\tau) \big] \to \min_{\tau}, \quad (38)$$

for

$$W^{(\kappa)}(x \mid z,\theta) = \frac{\gamma^{(\kappa)}\, q(x \mid z,\theta)}{p^{(\kappa)}\, f^{(\kappa)}(x)}, \quad (39)$$

where for each class $\kappa \in \{\mathrm{H},\mathrm{S}\}$, expected value $E_\xi^{(\kappa)}[\cdot]$ denotes conditional expectation $E_\xi[\,\cdot \,|\, \xi \sim f^{(\kappa)}(x)]$, $p^{(\kappa)}$ 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 — is used:

$$R(\tau \,|\, z) \approx \widehat{R}(\tau \,|\, z) = \sum_{(x_k,\kappa_k) \in T} W^{(\kappa_k)}(x_k \,|\, z,\, \theta)\, g(x_k \,|\, z,\tau). \quad (40)$$

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

From the necessary condition of extremum,

$$\frac{\partial}{\partial \tau_{ij}} R(\tau \mid z) = \mathbf{E}_{\xi,\kappa} \left[W^{(\kappa)}(\xi \mid z,\theta)\, \frac{\partial g(\xi \mid z,\tau)}{\partial \tau_{ij}} \right] = 0. \quad (41)$$

Since it has the form $\mathbf{E}[\,\cdot\,] = 0$, we may apply the method of stochastic optimization [30]. Switching to the parameters w_{ij} that were introduced in (21), we obtain the algorithm

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \gamma_t W^{(\kappa_{k(t)})}(z, x_{k(t)})\, g(x_{k(t)} \mid z, \tau(w^{(t)}))$$
$$\cdot\, n_i(x_{k(t)} \mid z)\, \big(\widehat{\tau}_{ij}(x_{k(t)} \mid z) - \tau_{ij}(w^{(t)})\big), \quad (42)$$

where $\gamma_t \geq 0$ is a sequence satisfying the properties

$$\sum_{t=0}^{\infty} \gamma_t = \infty, \qquad \sum_{t=0}^{\infty} \gamma_t^2 < \infty, \quad (43)$$

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

4.3.2 Probability-Based Criterion

Objective function (36) is nothing else but an empirical version of the criterion

$$R^{(\mathrm{H})}(\tau) \equiv \sum_{x \in X} \mathbf{1}^{(\mathrm{H})}(u)\, g(x \mid z,\tau) = \mathbf{E}_\xi\big[\mathbf{1}^{(\mathrm{H})}(u)\big], \quad (44)$$

where $\xi \sim g(x \mid z,\tau)$ and, as previously, $g(x \mid z,\tau)$ is generational probability distribution for a 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:

$$R(\tau) = \Pr\big[\mathbf{1}^{(\mathrm{H})}(\psi(z,\xi)) \mid z,\tau\big] \to \max_{\tau}. \quad (45)$$

As we have only two classes, maximization of the criterion (44), is equivalent to minimization of the dual criterion

$$R^{(\mathrm{S})}(\tau) \equiv \sum_{x \in X} \mathbf{1}^{(\mathrm{S})}(\psi(z,x))\, g(x \mid z,\tau), \quad (46)$$

for $\mathbf{1}^{(\mathrm{S})}(u) = 1 - \mathbf{1}^{(\mathrm{H})}(u)$. Combining (44) and (46), we have

$$R(\tau) \equiv \gamma^{(\mathrm{H})} R^{(\mathrm{H})}(\tau) - \gamma^{(\mathrm{S})} R^{(\mathrm{S})}(\tau) \to \max_{\tau}, \quad (47)$$

where $\gamma^{(\mathrm{H})} + \gamma^{(\mathrm{S})} = 1$ are some splitting weights. Let us now consider this problem in the context of both supervised and unsupervised learning.

Supervised Learning.

Formally rearranging the criterion function (47) into two sums and applying the importance sampling for the distribution of the pair (ξ,κ), we see that

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

$$= \sum_{\kappa\in\{\mathrm{H},\mathrm{S}\}} p^{(\kappa)} \sum_{x\in X} W^{(\kappa)}(z,x) f^{(\kappa)}(x)\, g(x \mid z,\tau)$$
$$= \mathbf{E}_\xi\big[W^{(\kappa)}(z,\xi)\, g(\xi \mid z,\tau)\big] \to \max_\tau, \tag{48}$$

where the variable ξ is distributed according to $f^{(\kappa)}(x)$, and

$$W^{(\kappa)}(z,x) = \frac{\gamma^{(\mathrm{H})}\mathbf{1}^{(\mathrm{H})}(\psi(z,x)) - \gamma^{(\mathrm{S})}\mathbf{1}^{(\mathrm{S})}(\psi(z,x))}{f^{(\kappa)}(x)}$$
$$= \frac{\mathbf{1}^{(\mathrm{H})}(\psi(z,x)) - \gamma^{(\mathrm{S})}}{f^{(\kappa)}(x)}. \tag{49}$$

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

$$R(\tau) \approx \widehat{R}(\tau) \equiv \sum_{(x_k,\kappa_k)\in T} W^{(\kappa_k)}(z,x_k)\, g(x_k \mid z,\tau). \tag{50}$$

For the parameters w_{ij} from (21), we obtain the following necessary condition of extremum:

$$\frac{\partial R(w)}{\partial w_{ij}} = \mathbf{E}_\xi\Big[W^{(\kappa)}(z,\xi)\, g(\xi \mid z,\tau(w))\, n_i(\xi \mid z)$$
$$\cdot \big(\widehat{\tau}_{ij}(\xi \mid z) - \tau_{ij}(w)\big)\Big] = 0, \tag{51}$$

where $\widehat{\tau}_{lk}(x)$ is defined as in (28), and $n_i(x \mid z)$, $n_{ij}(x \mid z)$ stand for the number of occurrences of the context $c_i \in A^K$ alone and followed by the character $a_j \in A$, respectively, in the text x. The corresponding stochastic approximation algorithm takes the form

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \gamma_t W^{(\kappa_{k(t)})}(z,x_{k(t)})\, g(x_{k(t)} \mid z,\tau(w^{(t)}))$$
$$\cdot\, n_i(x_{k(t)} \mid z)\big(\widehat{\tau}_{ij}(x_{k(t)} \mid z) - \tau_{ij}(w^{(t)})\big). \tag{52}$$

Unsupervised Learning.

In the case when true labelings κ_k of sampled messages $x_k \in T$ are unknown, we can alternatively apply importance sampling for the distribution

$$f(x) = p^{(\mathrm{H})} f^{(\mathrm{H})}(x) + p^{(\mathrm{S})} f^{(\mathrm{S})}(x). \tag{53}$$

Then,

$$R(\tau) = \mathbf{E}_\xi\big[W(z,\xi)\, g(\xi \mid z,\tau)\big] \to \max_\tau, \tag{54}$$

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

$$W(z,x) = \frac{\gamma^{(\mathrm{H})}\mathbf{1}^{(\mathrm{H})}(\psi(z,x)) - \gamma^{(\mathrm{S})}\mathbf{1}^{(\mathrm{S})}(\psi(z,x))}{f(x)}$$
$$= \frac{\mathbf{1}^{(\mathrm{H})}(\psi(z,x)) - \gamma^{(\mathrm{S})}}{p^{(\mathrm{H})} f^{(\mathrm{H})}(x) + p^{(\mathrm{S})} f^{(\mathrm{S})}(x)}. \tag{55}$$

Since the criteria (54) and (48) differ only in definitions of the weights $W(z,x)$ which are independent of w_{ij}, the resulting stochastic optimization algorithm is exactly the same as in (52) (again, up to differences between $W(z,x)$ and $W^{(\kappa)}(z,x)$).

4.4 Likelihood-Based Criterion

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

$$H(u \mid \tau) = -\frac{1}{|u|} \log\left(\prod_{l=1}^{|u|} g(u_l \mid u_{l-K}^{l-1},\tau)\right)$$
$$= -\frac{1}{|u|} \sum_{\substack{c_i\in A^K \\ a_j\in A}} n_{ij}(u) \log \tau_{ij}. \tag{56}$$

Averaged over random transformed messages u, it is equal to

$$H(\tau) = \mathbf{E}_u\big[H(u \mid \tau)\big] = -\sum_{c_i\in A^K} p_i\, H_i(\tau), \tag{57}$$

where

$$H_i(\tau) = \sum_{a_j\in A} p_{j|i} \log \tau_{ij}, \tag{58}$$

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 the character a_j occurring in u after the context c_i. On a single message u, $H_i(\tau)$ can be estimated as

$$H_i(\tau) \approx \widehat{H}_i(u \mid \tau) = \sum_{a_j\in A} \frac{n_{ij}(u)}{n_i(u)} \log \tau_{ij}. \tag{59}$$

Assuming a sample T of messages $x \in X$ is available, we can split T into auxiliary samples depending on the class $\kappa \in \{\mathrm{H},\mathrm{S}\}$ to which $u = \psi(z,x)$ is assigned by the classifier:

$$T^{(\kappa)} = \{x_k \in T \mid \mathbf{1}^{(\kappa)}(\psi(z,x_k) \mid \theta) = 1\}, \tag{60}$$

for $\mathbf{1}^{(\mathrm{H})}(x \mid \theta) = \mathbf{1}[q(x,\theta) < \alpha]$, and $\mathbf{1}^{(\mathrm{S})}(x \mid \theta) = \mathbf{1}[q(x,\theta) \ge \alpha]$. 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 $T^{(\mathrm{S})}$:

$$R^{(\kappa)}(\tau \mid z) = -\frac{1}{|T^{(\kappa)}|} \sum_{x_k\in T^{(\kappa)}} \sum_{c_i\in A^K} p_i\, \widehat{H}_i(\psi(z,x_k) \mid \tau). \tag{61}$$

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 both of these goals is to formalize them as a problem of minimizing the difference of the above objective functions:

$$R(\tau) = R^{(\mathrm{H})}(\tau \mid z) - R^{(\mathrm{S})}(\tau \mid z) \to \min_\tau, \tag{62}$$

subject to usual normalization requirements on τ.

Substituting the entropy estimation (59) definition into the criterion (62), we have:

$$R(\tau \mid z) = -\sum_{c_i\in A^K} p_i \left(\frac{1}{|T^{(\mathrm{H})}|} \sum_{u_k\in U^{(\mathrm{H})}} \widehat{H}_i(u \mid \tau) - \frac{1}{|T^{(\mathrm{S})}|} \sum_{u_k\in U^{(\mathrm{S})}} \widehat{H}_i(u \mid \tau)\right)$$
$$= -\sum_{c_i\in A^K} p_i \sum_{a_j\in A} (\nu_{ij}^{(\mathrm{H})} - \nu_{ij}^{(\mathrm{S})}) \log \tau_{ij}, \tag{63}$$

for

$$U^{(\kappa)} = \{\psi(z,x_k) \mid x_k \in T^{(\kappa)}\}, \tag{64}$$

$$\nu_{ij}^{(\kappa)} = \frac{1}{|T^{(\kappa)}|} \sum_{u_k\in U^{(\kappa)}} \frac{n_{ij}(u_k)}{n_i(u_k)}. \tag{65}$$

Since parameters τ_{ij} occur only in summands for the context c_i, optimization of (63) naturally falls into $|A|^K$ smaller

problems:

$$R_i(\tau \mid z) = R_i^{(\mathrm{H})}(\tau \mid z) - R_i^{(\mathrm{S})}(\tau \mid z) \to \min_{\{\tau_{i\bullet}\}}, \tag{66}$$

where $\{\tau_{i\bullet}\}$ stands for $\tau_{i1}, \tau_{i2}, \ldots, \tau_{i|A|}$, and

$$R_i^{(\kappa)}(\tau \mid z) = -\sum_{a_j \in A} \nu_{ij}^{(\kappa)} \log \tau_{ij}. \tag{67}$$

Unlike optimization problems (37) and (44) induced by the two approaches presented in Section 4.3, it is possible to solve problem (66) analytically:

THEOREM 1. *The criterion function*

$$R_i(\tau \mid z) = -\sum_{a_j \in A} (\nu_{ij}^{(\mathrm{H})} - \nu_{ij}^{(\mathrm{S})}) \log \tau_{ij}, \tag{68}$$

subject to constraints

$$\tau_{ij} \geq 0 \;\; \textit{and} \sum_{a_j \in A} \tau_{ij} = 1, \tag{69}$$

reaches its minimum value at

$$\tau_{ij}^* = \frac{\mu_{ij}}{\mu_i}, \tag{70}$$

where

$$\mu_{ij} = \max\left\{0, \nu_{ij}^{(\mathrm{H})} - \nu_{ij}^{(\mathrm{S})}\right\}, \tag{71}$$

$$\mu_i = \sum_{a_j \in A} \mu_{ij}. \tag{72}$$

Proof of Theorem 1 is provided in Appendix.

4.5 Multiple Base Messages

Throughout this section, 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 from Section 4.3 resulting in stochastic optimization, the only change caused by averaging over Z is that the variable z, like x, also runs over a sample on iterations:

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \gamma_t W^{(\kappa_{k(t)})}(z_{l(t)}, x_{k(t)})\, g(x_{k(t)} \mid z_{l(t)}, \tau(w^{(t)})) \cdot n_i(x_{k(t)} \mid z_{l(t)}) \left(\widehat{\tau}_{ij}(x_{k(t)} \mid z_{l(t)}) - \tau_{ij}(w^{(t)})\right). \tag{73}$$

The same is true for the likelihood-based approach discussed in Section 4.4. If criterion (62) 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 (63) but for

$$\nu_{ij}^{(\kappa)} = \frac{1}{|Z|\,|T^{(\kappa)}|} \sum_{z_l \in Z} \sum_{u_k \in T^{(\kappa)}} \frac{n_{ij}(\psi(z_l, x_k))}{n_i(\psi(z_l, x_k))}. \tag{74}$$

5. EVALUATION

5.1 Methodology

In order to validate the method proposed in this work, first we implemented the entropy classifier for the problem of spam filtering. Following the definition of the problem given in Section 3.4, our implementation uses PPM to learn the 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 4.3.1, 4.3.2, and 4.4.

Throughout all evaluations presented in this section, in the spirit of the works [7, 8, 18], we worked with character-based PPM models. Although it is possible to use a word-based alphabet instead, switching to it vastly increases the alphabet size (and, consequently, the number of parameters), ignores differences in punctuation and spacing, and makes the algorithm sensitive to tokenization. Furthermore, it does not make the classifier stronger: In our experiments, word-based PPM spam filters have comparable or worse accuracy than character-based ones (for the latter having greater context sizes than the former, of course).

Our numerical experiments were organized as follows. For each evaluation run, 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 [2]. Each message in $x_k \in T$ was accompanied with the true class labeling $\kappa_k \in \{\mathrm{H}, \mathrm{S}\}$.

The sample T was additionally temporarily split at random in proportion seven to three into the training and testing samples, correspondingly. 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 [7]). 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 (42), (52), and (70) were run on the combined sample T in order to obtain transformation parameters $\tau^{(\mathrm{E})}$, $\tau^{(\mathrm{P})}$, and $\tau^{(\mathrm{L})}$. 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 $g(x, \tau^{(t)})$. The generation was 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 $20\,|z|$, 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 (70) required less work since it provides the analytical solution as long as the values μ_{ij} are 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 run, so there was an auxiliary measure for comparing this algorithm with the other two and the baseline strategy.

The role of the baseline strategy in our experiments was played by the same generation procedure, but executed for the vector of parameters $\tau^{(\mathrm{H})} = \theta^{(\mathrm{H})}$ that were estimated

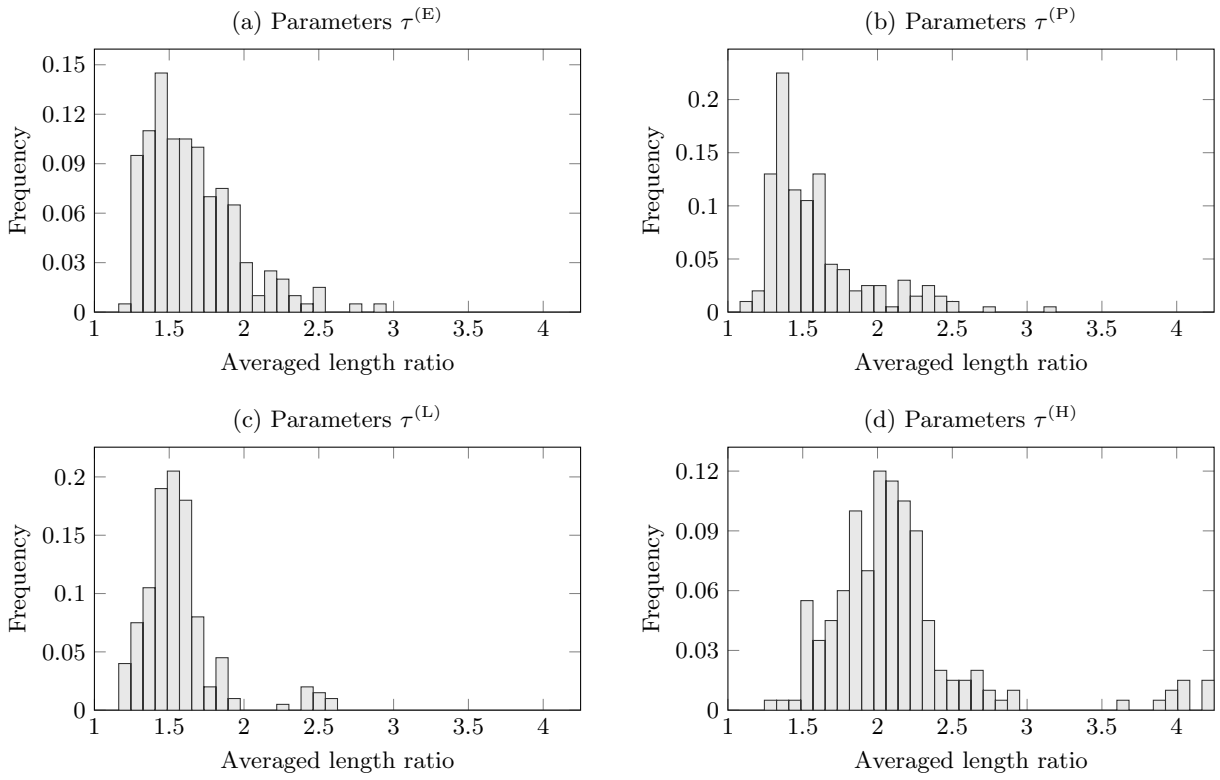

Figure 1: Histograms of the length ratio $|\psi(z_l, x_k)|/|z_l|$ averaged over appendices x_k generated for the order $K = 3$ on *1% datasets* using (a) the parameters $\tau^{(\mathbf{E})}$ optimized for the *entropy-based criterion* (38), (b) the parameters $\tau^{(\mathbf{P})}$ optimized for the *probability-based criterion* (47), (c) the optimal parameters $\tau^{(\mathbf{L})}$ (70) for the *likelihood-based criterion* (63), and (d) the *baseline parameters* $\tau^{(\mathrm{H})}$ estimated from $\theta^{(\mathrm{H})}$.

Table 1: Accuracy of the classifier on the full dataset

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

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.

5.2 Results

Due to limited computational resources, during all evaluation runs, Markov models' memory was set to three characters for both the classifier and the adversary. In practice, entropy-based spam filters demonstrate the best performance for the Markov models of orders between six and eight characters [7]. 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 1 shows statistics for one run of the classifier, when all 6046 bodies of email messages were randomly split into 3627 training and 2419 testing messages.

For $K = 3$, the space of parameters τ, representing conditional probabilities of one-byte characters given a context of at most K another one-byte characters, is bounded by

Table 2: Summary statistics for the performance of different generation parameters on *1% datasets*

Index	Optimization based on Entropy	Prob.	Likelih.	Ham baseline
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
Failure rate (%)	0	0	0	0

$256^1 + 256^2 + 256^3 + 256^4 \approx 2^{32}$. The number of character-context combinations that actually occur in the whole SpamAssassin dataset for $K = 3$ is approximately 524 000.

To avoid memory pressure and achieve faster convergence, algorithms (42) and (52) 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 run on a 1% dataset done for all three algorithms, as described in the previous section.

The failure rate of the chosen concatenation-based transformation was zero for all spam messages and parameters τ

Table 3: Summary statistics for the performance of generation with parameters optimal for the *likelihood-based criterion* on the *full dataset* for orders K from 3 to 10

Index	Likelihood-based optimization								Ham baseline							
	3	4	5	6	7	8	9	10	3	4	5	6	7	8	9	10
Averaged length ratio																
Minimum	1.00	1.01	1.01	1.02	1.02	1.02	1.01	1.01	1.00	1.01	1.04	1.04	1.02	1.03	1.02	1.05
5% quantile	1.04	1.04	1.04	1.05	1.05	1.04	1.04	1.04	1.25	1.34	1.40	1.45	1.48	1.51	1.52	1.54
Median	1.08	1.08	1.08	1.08	1.08	1.07	1.07	1.07	1.70	1.75	1.79	1.81	1.82	1.85	1.87	1.89
Mean	1.11	1.10	1.10	1.09	1.09	1.08	1.08	1.07	1.99	1.99	1.98	1.96	1.94	1.95	1.95	1.96
95% quantile	1.33	1.27	1.23	1.19	1.17	1.15	1.14	1.13	4.14	3.73	3.38	3.12	2.93	2.82	2.72	2.68
Maximum	2.63	1.56	1.51	1.39	1.27	1.46	1.66	1.84	8.50	7.53	7.81	7.94	5.44	6.19	7.72	6.19
Failure rate (%)	0	0	0	0	0	0	0	0	0.16	0	0.01	0.01	0.01	0.01	0	0

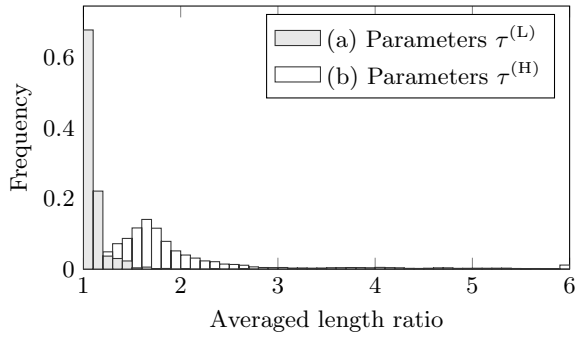

Figure 2: Histograms of the length ratio $|\psi(z_l, x_k)|/|z_l|$ averaged over appendices x_k generated for the order $K = 3$ on the *full dataset* using (a) the optimal parameters $\tau^{(L)}$ (70) for the *likelihood-based criterion* (63), and (b) the *baseline parameters* $\tau^{(H)}$ estimated from $\theta^{(H)}$.

obtained from all three algorithms as well as the Ham baseline $\tau^{(H)}$. 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 $|u_k|/|z_l|$ for each transformed message. Note that none of the methods proposed in this work was constructed to directly optimize this ratio of lengths.

Figures 1(a), 1(b), and 1(c) depict distributions of length ratios averaged over transformation appendices x_k generated according to the parameters $\tau^{(E)}$, $\tau^{(P)}$, and $\tau^{(L)}$ that were optimized for the entropy-based, probability-based, and likelihood-based criteria, respectively. As it is evident from the shapes of the histograms, 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 1(d), showing the histogram for the parameters $\tau^{(H)}$, we see that all three techniques provide a noticeably better performance than the baseline of generating Ham-like appendices. A short summary of the statistics from these figures is given in Table 2.

Figure 3 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 a few continuations generated according to optimal parameters (highlighted with gray background). Any of the presented appendices (by itself) is sufficient to make the corresponding spam message legitimate for the classifier trained on a 1% dataset, and cannot be shortened without changing the class of the transformed message back to spam.

Since the likelihood-based algorithm (70) 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, and also with higher values of the Markov-chain order K. Resulting distributions of the length ratio for the optimal parameters under the likelihood-based criterion, and for the baseline Ham parameters are presented in Figure 2. Table 3 lists the same five quantiles of the length ratio as well as its mean values. Comparing these statistics with the ones in Table 2, we conclude that the gap between the Ham-like generation and the likelihood-based algorithm is even greater on the full dataset.

6. 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 the 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 some implementation difficulties, after appropriate calibration, they showed comparable performance. Together, all three methods have shown the feasibility of attacking compression-based classifiers statistically using relatively limited extent of transformation.

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 for optimizing the contents of transformation texts directly on a per-character basis (e.g., genetic algorithms or Markov chain Monte Carlo). This problem setting can potentially increase the number of different criterion functions that can be considered to formalize the adversary problem.

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

```
  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.
```

```
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 3: Examples of original spam messages z_l (white background) and several appendices x_k corresponding to each z_l that are generated for the order $K = 3$ using parameters $\tau^{(\mathbf{E})}$ optimized on a *1% dataset* (gray background).

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

Finally, it is important to investigate ways of improving robustness of compression-based classifiers, given the knowledge of potential adversary attacks. For linear filters, it has been shown that additional regularization [37] or frequent retraining [26] can mitigate the severity of most attacks. The research of whether such methods may also be adaptable to compression-based classifiers is left for future work.

Acknowledgments

We thank the anonymous reviewers for helpful comments. This research was funded by NSF grant IIS-1451453.

7. REFERENCES

[1] L. Akoglu, R. Chandy, and C. Faloutsos. Opinion fraud detection in online reviews by network effects. In *Proceedings of the 7th International AAAI Conference on Weblogs and Social Media*. AAAI Press, 2013.

[2] Apache SpamAssassin Project. The SpamAssassin public mail corpus. Available at `https://spamassassin.apache.org/publiccorpus/`, 2005.

[3] M. Bailey, J. Oberheide, J. Andersen, Z. M. Mao, F. Jahanian, and J. Nazario. Automated classification and analysis of internet malware. In *Recent Advances in Intrusion Detection*, pages 178–197. Springer, 2007.

[4] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on Twitter. In *Proceedings of the 7th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, 2010.

[5] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases*, pages 387–402. Springer, 2013.

[6] V. Bobicev and M. Sokolova. An effective and robust method for short text classification. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pages 1444–1445, 2008.

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

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

[9] M. Brückner and T. Scheffer. Nash equilibria of static prediction games. In *Advances in Neural Information Processing Systems 22*, 2009.

[10] M. Brückner and T. Scheffer. Stackelberg games for adversarial prediction problems. In *Proceedings of the Seventeenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press, 2011.

[11] D. Chau, S. Pandit, and C. Faloutsos. Detecting fraudulent personalities in networks of online auctioneers. *Knowledge Discovery in Databases: PKDD 2006*, pages 103–114, 2006.

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

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

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

[15] O. Dekel, O. Shamir, and L. Xiao. Learning to classify with missing and corrupted features. *Machine Learning*, 81(2):149–178, 2010.

[16] H. Drucker, D. Wu, and V. N. Vapnik. Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, 10(5):1048–1054, 1999.

[17] T. Fawcett. "In vivo" spam filtering: A challenge problem for KDD. *SIGKDD Explorations*, 5(2):140–148, 2003.

[18] E. Frank, C. Chui, and I. H. Witten. Text categorization using compression models. In *Proceedings of DCC-00, IEEE Data Compression Conference*, pages 200–209. IEEE Computer Society Press, Los Alamitos, US, 2000.

[19] J. Goodman, D. Heckerman, and R. Rounthwaite. Stopping spam. *Scientific American*, 292(4):42–49, 2005.

[20] J. Graham-Cumming. How to beat an adaptive spam filter. In *MIT Spam Conference*, 2004.

[21] N. Jindal and B. Liu. Opinion spam and analysis. In *Proceedings of the International Conference on Web Search and Data Mining*, pages 219–230. ACM, 2008.

[22] Z. Jorgensen, Y. Zhou, and M. Inge. A multiple instance learning strategy for combating good word attacks on spam filters. *Journal of Machine Learning Research*, 9:1115–1146, 2008.

[23] A. Kantchelian, J. Ma, L. Huang, S. Afroz, A. Joseph, and J. D. Tygar. Robust detection of comment spam using entropy rate. In *Proceedings of 5th ACM Workshop on Artificial Intelligence and Security*, pages 59–70. ACM Press, 2012.

[24] J. Z. Kolter and M. A. Maloof. Learning to detect and classify malicious executables in the wild. *Journal of Machine Learning Research*, 7:2721–2744, 2006.

[25] D. Lowd and C. Meek. Adversarial learning. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 641–647. ACM, 2005.

[26] D. Lowd and C. Meek. Good word attacks on statistical spam filters. In *Proceedings of the Second Conference on Email and Anti-Spam*, 2005.

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

[28] B. Nelson, B. I. P. Rubinstein, L. Huang, A. D. Joseph, S. J. Lee, S. Rao, and J. D. Tygar. Query strategies for evading convex-inducing classifiers. *Journal of Machine Learning Research*, 13:1293–1332, 2012.

[29] K. Rieck, P. Trinius, C. Willems, and T. Holz. Automatic analysis of malware behavior using machine learning. *Journal of Computer Security*, 19(4):639–668, 2011.

[30] H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22:400–407, 1951.

[31] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A Bayesian approach to filtering junk e-mail. In *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*, Madison, WI, 1998. AAAI Press.

[32] D. Stevens and D. Lowd. On the hardness of evading combinations of linear classifiers. In *Proceedings on the 2013 ACM Workshop on Artificial Intelligence and Security (AISec)*, Berlin, Germany, 2013. ACM Press.

[33] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 1–9. ACM, 2010.

[34] W. J. Teahan. Probability estimation for PPM. In *New Zealand Computer Science Research Students' Conference*, 1995.

[35] C. H. Teo, A. Globerson, S. Roweis, and A. Smola. Convex learning with invariances. In *Advances in Neural Information Processing Systems 20*, 2007.

[36] Y. Vorobeychik and B. Li. Optimal randomized classification in adversarial settings. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, pages 485–492. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[37] H. Xu, C. Caramanis, and S. Mannor. Robustness and regularization of support vector machines. *The Journal of Machine Learning Research*, 10:1485–1510, 2009.

[38] C. Yang, R. C. Harkreader, and G. Gu. Die free or live hard? Empirical evaluation and new design for fighting evolving Twitter spammers. In *Recent Advances in Intrusion Detection*, pages 318–337. Springer, 2011.

[39] Y. Zhou and W. M. Inge. Malware detection using adaptive data compression. In *Proceedings of 1st ACM Workshop on Artificial Intelligence and Security*, pages 53–60. ACM, 2008.

APPENDIX

LEMMA 1. *The objective function*

$$R_i(\tau) = -\sum_{j\in J} \nu_{ij} \log \tau_{ij}, \tag{75}$$

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

$$\tau_{ij} \geq 0 \ \text{ and } \ \sum_{j\in J} \tau_{ij} = s > 0, \tag{76}$$

reaches its minimum value at

$$\tau^*_{ij} = s\frac{\nu_{ij}}{\nu_i}, \tag{77}$$

where

$$\nu_i = \sum_{j\in J} \nu_{ij}. \tag{78}$$

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

$$R_i(\tau^*) - R_i(\tau) = -\sum_{j\in J} \nu_{ij} \log \frac{s\nu_{ij}}{\nu_i} + \sum_{j\in J} \nu_{ij} \log \tau_{ij}$$

$$= \sum_{j\in J} \nu_{ij} \log \frac{\tau_{ij}\nu_i}{s\nu_{ij}} \leq \sum_{j\in J} \nu_{ij} \left(\frac{\tau_{ij}\nu_i}{s\nu_{ij}} - 1 \right)$$

$$= \frac{\nu_i}{s} \sum_{j\in J} \tau_{ij} - \sum_{j\in J} \nu_{ij} = 0, \tag{79}$$

by definition of ν_i and requirements (76). From the obtained relation it follows that $R_i(\tau^*) \le R_i(\tau)$ for any τ. □

LEMMA 2. *The objective function*

$$R_i(\tau) = -\sum_{j\in J} \nu_{ij} \log \frac{1}{\tau_{ij}}, \tag{80}$$

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

$$\tau_{ij} \ge 0 \ \text{ and } \ \sum_{j\in J} \tau_{ij} = s > 0, \tag{81}$$

reaches its minimum value at

$$\tau^*_{ij} = \begin{cases} \dfrac{s}{|J_i^{\min}|}, & \text{if } j \in J_i^{\min}; \\ 0, & \text{if } j \in J \setminus J_i^{\min}; \end{cases} \tag{82}$$

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

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

$$\tau^*_{ij}(\varepsilon) = \begin{cases} s\dfrac{1 - (|J| - |J_i^{\min}|)\varepsilon}{|J_i^{\min}|}, & \text{if } j \in J_i^{\min}; \\ s\varepsilon, & \text{if } j \in J \setminus J_i^{\min}. \end{cases} \tag{83}$$

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:

$$R_i(\tau^*(\varepsilon)) = -\sum_{j\in J_i^{\min}} \nu_{ij} \log \frac{|J_i^{\min}|}{s\big(1 - (|J| - |J_i^{\min}|)\varepsilon\big)} - \sum_{j\notin J_i^{\min}} \nu_{ij} \log \frac{1}{s\varepsilon}. \tag{84}$$

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

$$\lim_{\varepsilon\to 0} R_i(\tau^*(\varepsilon)) = -\infty, \qquad \lim_{\varepsilon\to 0} \tau^*(\varepsilon) = \tau^*. \tag{85}$$

Solution (82) 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. □

Now we can proceed to the proof of Theorem 1.

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 $\delta_{ij} \equiv \nu^{(\mathrm{H})}_{ij} - \nu^{(\mathrm{S})}_{ij}$:

$$R_i(\tau \mid z) = -\sum_{j\in J_i^+} \delta_{ij} \log \tau_{ij} - \sum_{j\in J_i^-} \delta_{ij} \log \tau_{ij} - \sum_{j\in J_i^0} \delta_{ij} \log \tau_{ij} = -\sum_{j\in J_i^+} \delta_{ij} \log \tau_{ij} - \sum_{j\in J_i^-} (-\delta_{ij}) \log \frac{1}{\tau_{ij}}, \tag{86}$$

where $J_i^\sigma = \{j \mid a_j \in A \wedge \operatorname{sgn}(\delta_{ij}) = \sigma\}$, and, similarly,

$$s_i^\sigma = \sum_{j\in J_i^\sigma} \tau_{ij}, \qquad s_i^+ + s_i^- + s_i^0 = 1. \tag{87}$$

Clearly, the problem of finding optimal τ_{ij} can be solved separately for each of the sums in (86).

- For the first sum

$$R_i^+(\tau) = -\sum_{j\in J_i^+} \delta_{ij} \log \tau_{ij}, \tag{88}$$

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

$$\tau^*_{ij} = \frac{s_i^+\, \delta_{ij}}{\sum_{l\in J_i^+} \delta_{il}}, \qquad j \in J_i^+. \tag{89}$$

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

- For the second sum

$$R_i^-(\tau) = -\sum_{j\in J_i^-} (-\delta_{ij}) \log \frac{1}{\tau_{ij}}, \tag{90}$$

conditions of Lemma 2 hold for $J = J_i^-$, $\nu_{ij} = -\delta_{ij}$, 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

$$\tau^*_{ij}(\varepsilon) = \begin{cases} s_i^- \dfrac{1 - (|A| - |J_i^{\min}|)\varepsilon}{|J_i^{\min}|}, & \text{if } j \in J_i^{\min}; \\ s_i^-\varepsilon, & \text{if } j \in J_i^- \setminus J_i^{\min}; \end{cases} \tag{91}$$

$$J_i^{\min} = \{j \in J_i^- \mid -\delta_{ij} = \min_{l\in J_i^-} (-\delta_{il})\}. \tag{92}$$

Notice that, since τ_{ij} occurs in $R_i^-(\tau)$ inverted, unlike $R_i^+(\tau)$, 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 s_i^0.

In order to combine independent solutions (89) and (91) optimizing separate sums, it is necessary to determine in which proportion should the probability mass be distributed between parameters belonging to J_i^+, J_i^-, and J_i^0. As we have seen above, for the minimal value (as a function of the bound ε) to be the least, 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

$$s_i^0 = |J_i^0|\,\varepsilon, \quad s_i^- = |J_i^-|\,\varepsilon, \quad s_i^+ = 1 - s_i^- - s_i^0. \tag{93}$$

The corresponding parameters are then

$$\tau^*_{ij}(\varepsilon) = \begin{cases} \dfrac{\big(1 - (|A| - |J_i^+|)\varepsilon\big)\delta_{ij}}{\sum_{l\in J_i^+} \delta_{il}}, & \text{if } j \in J_i^+; \\ \varepsilon, & \text{if } j \in J_i^- \cup J_i^0. \end{cases} \tag{94}$$

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

$$\tau^*_{ij} = \lim_{\varepsilon\to 0} \tau^*_{ij}(\varepsilon) = \begin{cases} \dfrac{\delta_{ij}}{\sum_{l\in J_i^+} \delta_{il}}, & \text{if } j \in J_i^+; \\ 0, & \text{if } j \in J_i^- \cup J_i^0; \end{cases} = \frac{\max\{0, \delta_{ij}\}}{\sum_{a_j\in A} \max\{0, \delta_{ij}\}} = \frac{\mu_{ij}}{\mu_i}. \quad \square \tag{95}$$