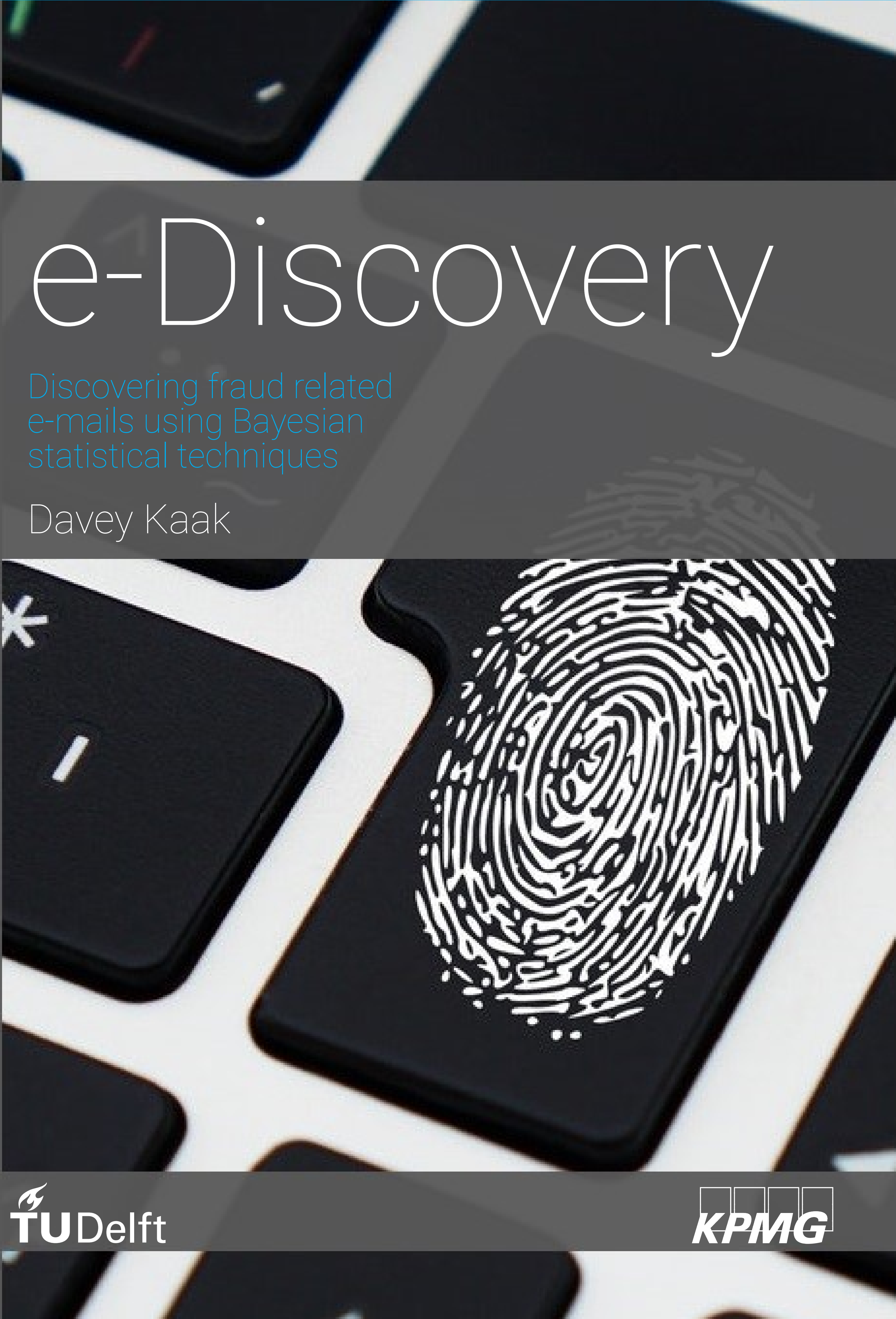


e-Discovery

Discovering fraud related e-mails using Bayesian statistical techniques

Davey Kaak

Delft University of Technology

e-Discovery

Discovering fraud related e-mails using Bayesian statistical techniques

by

Davey Kaak

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on June 6, 2019 at 15:00.

Student number:	4377966	
Project duration:	November 1, 2018 – June 6, 2019	
Thesis committee:	Dr. ir. J. Bierkens,	TU Delft, supervisor
	I. Dashorst MSc,	KPMG, supervisor
	Dr. D. Kurowicka	TU Delft

Abstract

During a digital fraud investigation the search for relevant information in mailboxes of custodians is like finding a needle in a haystack. This time consuming task can, on various levels, be improved and made more efficient. Technology Assisted Review (TAR) is already one of the available machine learning algorithms that helps speeding up the process of finding relevant information. In Technology Assisted Review a model is trained based on the classification of e-mails by expert review. During the review process TAR continuously gives back the (potentially) most relevant e-mails that still need to be given a classification. The downside of this algorithm is that a manual expert review is still needed before TAR can give recommendations. This thesis will focus on introductory research on models that give an initial sorting before the expert review is done. The hypothesis that will be used is that this sorting (or classification) can be done in a similar manner as spam e-mails are removed to the junk folder in a mailbox. Three different features have been used (word frequencies, word occurrences and length of an e-mail) on four different models for each feature (A generative and discriminative model, each with maximum likelihood estimation or Bayesian estimation). Each of these 12 different implementations have been tested on three different datasets (TREC, ENRON and a confidential dataset). Based on 5-fold cross validation the Bayesian generative model based on word frequencies has been shown to perform best on the confidential dataset. This model shows that a classification at the start of a digital fraud investigation can be helpful. Combining different models, and finding the best parameters for practical usage of the model is left for further research.

Keywords: classification, fraud, generative model, discriminative model, Naive Bayes, logistic regression, TAR, e-Discovery

Preface

When I started my bachelors in Applied Mathematics I would never have thought statistics and probability to be that interesting. During the various courses, gradually, the endless possibilities of analysing and modeling data became clear to me. At the start of my masters I was especially interested in the detection of anomalies and rare event data. I am very happy that KPMG Forensic Technology gave me the possibility to contribute to the detection of fraud, a subject that is not only theoretically interesting but also has major consequences for society.

I would like to thank my supervisor at the TU Delft, Joris Bierkens, for his support and input. Without the input and brainstorm sessions I do not think the results, and especially the report, would have turned out how it did. Furthermore, I would also like to thank my daily supervisor at KPMG, Ian Dashorst, for his help on the fraud related part of the thesis as well as his constructive feedback. Last, but not least, I would like to thank the Forensic Technology team for giving me (more) insight information on fraud prevention and detection, and for the great atmosphere during the thesis period.

Davey Kaak
Amstelveen, May 2019

"Outlier: Observation which deviates so much from other observations
as to arouse suspicion it was generated by a different mechanism."

Hawkins (1980)

Contents

List of Figures

List of Tables

List of Symbols

X, Y	random variables
$f_X(x)$	probability distribution function of X
$p(x)$	probability distribution function of X (Bayesian notation)
$f_{X\mid Y}(x \mid y)$	probability distribution function of X given Y
$p(x \mid y)$	probability distribution function of X given Y (Bayesian notation)
J	size of the dictionary of words
y	indication of being spam/fraud ($y = 1$) or not spam/fraud ($y = 0$)
ξ	same usage as y, used in some case for clarification purposes
N	number of e-mail messages
n	length of ordered sequence of words for a particular e-mail
t_i	i-th word in the dictionary of words
θ_j^y	relative frequency of j-th word in the dictionary corresponding to y
$\boldsymbol{\theta}^0, \boldsymbol{\theta}^1$	vector of relative frequencies θ_j^y of the complete dictionary
q_j^y	probability that word t_j corresponds to category y
$\boldsymbol{z}$	ordered sequence (length n) of words
x_j	feature of the j-th word in the dictionary (e.g. word counts)
u_j	a particular feature
$\boldsymbol{x}$	vector of features x_j
$\boldsymbol{u}$	vector of features u_j
$\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(N)}$	sequence of e-mails
$\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}$	sequence of features
$y^{(1)}, \ldots, y^{(N)}$	sequence of indications whether an e-mail is spam or not
$\boldsymbol{z}^{(N+1)}$	a new e-mail (to be given a classification)
$\boldsymbol{x}^{(N+1)}$	features corresponding to a new e-mail
$y^{(N+1)}$	classification of a new e-mail
$\boldsymbol{w}$	weights used in the model of Logistic Regression
$\alpha, \boldsymbol{\alpha}, \boldsymbol{\alpha}^y, \tilde{\boldsymbol{\alpha}}^y$	hyper parameters or vector of hyper parameters
$\beta, \boldsymbol{\beta}, \boldsymbol{\beta}^y, \tilde{\boldsymbol{\beta}}^y$	hype rparameters or vector of hyper parameters
η	hyper parameter
n_{tp}	Number of true positives
n_{tn}	Number of true negatives
n_{fp}	Number of false positives
n_{fn}	Number of false negatives

List of Abbreviations

ASCII	American Standard Code for Information Interchange
CPM	Count per message
DM	Discriminative Model
EDRM	Electronic Discovery Reference Model
FERC	Federal Energy Regulatory Commission
GM	Generative Model
LR	Logistic Regression
MAP estimate	Maximum a posteriori estimate
MIME	Multipurpose Internet Mail Extensions
MLE	Maximum Likelihood Estimation
n.a.	not available
NB	Naive Bayes
RFC	Request for Comments
SVM	Support Vector Machines
TAR	Technology Assisted Review
TREC	Text Retrieval Conference

1 Introduction

From time to time a case related to fraudulent activities is presented in the news. Several, at time of writing, recent examples are the Steinhoff accounting fraud [19], tax fraud in Curacao [18], or interest rate derivative fraud [17]. Consider that in each of these cases quite a lot of money is involved, but the case itself is only one of the many fraud related cases that take place every year. According to an article published in 2013 [10], and based on research done by PwC [12], fraudulent activities are costing the Dutch society 11 billion euro's per year. This gives companies and organisations more than enough reasons to try and combat this type of crime.

During an investigation that is related to (suspected) fraudulent activities many different stages must be completed before any actual evidence can be presented. Figure 1.1 shows an overview of the so called e-Discovery process, which is one of the means available. e-Discovery is the process related to the identification of relevant information in electronic material. This process includes all steps that are needed from the point of data collection to the presentation of actual results, but also includes the information governance. This process is in general done by hand (expert review), and uses different e-Discovery tools to help speed up the process. One of the parts that is especially time consuming is the investigation of the mailboxes of custodians that are suspected to have taken part in the case. In Figure 1.1 this part is visualised with the dark blue boxes. The aim of this thesis is to build a model that classifies the e-mails before the review process begins. This will be applied alongside the 'processing' part in Figure 1.1. With this classification beforehand, the number of e-mails that need to be reviewed before the most important and evidential e-mails are found will be reduced. In the next section the state of the art for fraud classification is discussed.

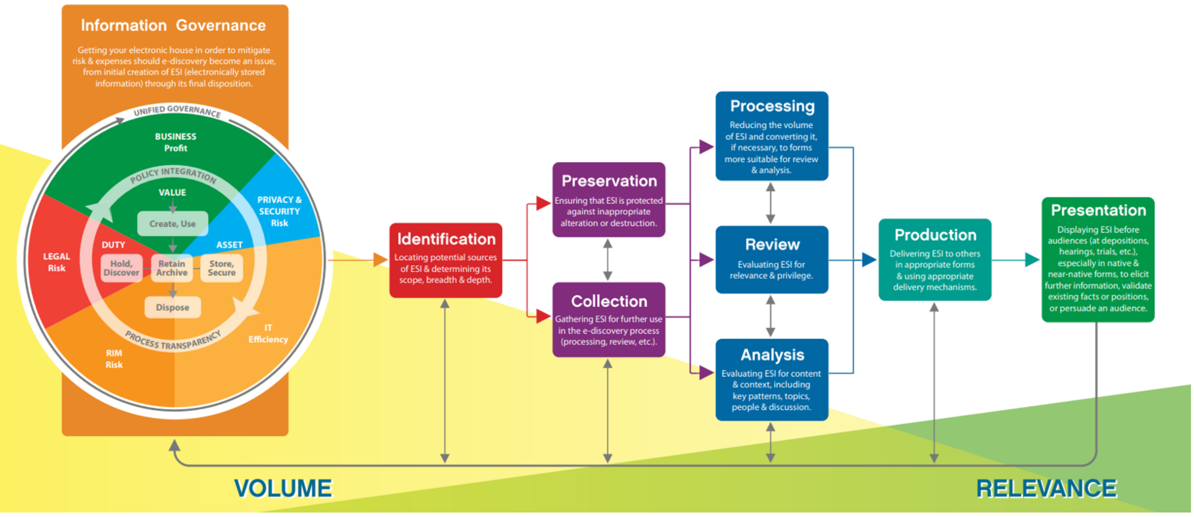

Information Governance
Getting your electronic house in order to mitigate risk & expenses should e-discovery become an issue, from initial creation of ESI (electronically stored information) through its final disposition.
UNIFIED GOVERNANCE
BUSINESS Profit
POLICY INTEGRATION
VALUE
Create, Use
PRIVACY & SECURITY Risk
LEGAL Risk
DUTY
ASSET
Hold, Discover
Retain Archive
Store, Secure
Dispose
PROCESS TRANSPARENCY
IT Efficiency
RIM Risk
Identification
Locating potential sources of ESI & determining its scope, breadth & depth.
Preservation
Ensuring that ESI is protected against inappropriate alteration or destruction.
Collection
Gathering ESI for further use in the e-discovery process (processing, review, etc.).
Processing
Reducing the volume of ESI and converting it, if necessary, to forms more suitable for review & analysis.
Review
Evaluating ESI for relevance & privilege.
Analysis
Evaluating ESI for content & context, including key patterns, topics, people & discussion.
Production
Delivering ESI to others in appropriate forms & using appropriate delivery mechanisms.
Presentation
Displaying ESI before audiences (at depositions, hearings, trials, etc.), especially in native & near-native forms, to elicit further information, validate existing facts or positions, or persuade an audience.
VOLUME
RELEVANCE

Figure 1.1: EDRM Reference model (version July 2018) [16], which summarises all the stages of an e-Discovery process. The investigation of potential proof of fraudulent activities is represented by the dark blue boxes. This thesis focuses on finding a classification model that can be applied alongside the 'processing' part.

1.1. State of the Art fraud classification

For the classification of fraud-related e-mail in the setting as mentioned in the previous section little to no research is available. This has partially to do with reason that not a lot of datasets are (publicly) available. Current methods to detect fraud-related e-mail are based on expert review or based on Technology Assisted Review (TAR). Furthermore, although not specifically applied in a similar setting, quite some research has been written on the topic of the classification of fraudulent financial statements.

1.1.1. Current methodology

Currently the methodology used for classification of fraud-related e-mail is, according to Lawton et al. [37] and the EDRM reference model [16], based on expert judgment. It is part of the e-Discovery process which is shown in Figure 1.1. During an e-Discovery investigation the aim is to "retrieve information from a digital archive in a systematic way." This process can be very time consuming, and due to the increasing size of data this process continues to be even more time consuming. The part of the classification of fraud-related e-mail in an e-Discovery process is visualised in Figure 1.1 with the box stating "review". As mentioned before, this research report focuses on classifying e-mails before the review process and therfore focusses on the blue box with label "processing" in Figure 1.1.

In the current process of expert review it is customary to use keyword based searching as well as using methods that decrease the size of the dataset (for example by deleting irrelevant e-mails or duplicates) [37].

1.1.2. Technology Assisted Review

Technology Assisted Review is the, currently, most used method to reduce the amount of time needed to review the e-mails in an e-Discovery case. According to the EDRM [15] TAR is "a process of having computer software electronically classify documents based on input from expert reviewers, in an effort to expedite the organization and prioritization of the document collection." In other words, TAR helps identifying the potentially most relevant documents, based on expert reviews. The process of Technology Assisted Review can be summarised in three steps:

1. Expert reviews a batch of documents
2. The TAR model is trained based on the expert reviewed documents and gives an indication of labels (classification) for the remaining documents
3. The expert reviewer, reviews the classification of its model, this is essentially step 1 repeated on the next batch.

In comparison with the current basic methodology and this research report, TAR is especially focused on the three blue boxes in Figure 1.1. It is a continuous training and classification process between those three steps.

1.1.3. Overview research results

The classification of fraud-related e-mail is not discussed in many research articles. Two articles were found that did conclude that deceptive communication in e-mail can be found by analysing specific word frequency [33, 34]. Both reports especially looked at first and third person pronouns. Although both of these research are not very elaborate, they do give us an indication that certain usage of words is more frequent in deceptive (and thus potentially fraudulent e-mail) compared to 'normal' e-mails.

More research has been conducted on the analysis of financial statements and annual reports on hidden fraudulent activities. Humpherys et al. [31] state that people "[...] crafting fraudulent disclosures use more activation language, words, imagery, pleasantness, group references, and less lexical diversity than non fraudulent ones. Writers of fraudulent disclosures may write more to appear credible while communicating less in actual content." With models containing various linguistic features they managed to get an accuracy of about 60-70% on the detection of fraud/non-fraud. Comparing these results to the application of classifying e-mails as fraud/non-fraud gives us also a positive expectation. However, it should be mentioned that financial statements often have a longer length than e-mails, and therefore linguistic features might be less distinctive in the case of e-mails.

Glancy and Yadav [26] as well as Chen et al. [9] found that a computational fraud detection model, tested on management's discussion and analysis also detected fraud quite well. They both state that their main criticism is that they used a dataset of (only) 69 companies. Glancy and Yadav do give the recommendation to do additional research of applying deception detection in e-mail.

Fissette [20] concluded that text analysis can contribute to the detection of fraudulent annual statements. She applied Naive Bayes models as well as Support Vector Machines and a Neural Network model. Although this is promising, she also concluded that it is desirable to further enhance the performance (about 75% with the Neural Network model).

Furthermore, as is pointed out in the previous section, Technology Assisted Review is also used nowadays. However, since most of the models used in TAR and that are applied in e-Discovery cases are developed by private companies, little can be found about the exact implementation or the used features.

1.2. Hypothesis

It is already concluded that little to no research is available on a similar classification problem as the one that is discussed in this thesis. This makes the application described in this report almost unique in its field, and also of importance for future use and research.

A similar problem as the classification of fraud related e-mail messages is the classification of spam e-mails. For spam classification a lot of research has already been published, and therefore many models that work well are known. The idea is that there must be unique characteristics of both relevant and not relevant e-mails as well as of both spam and ham (i.e. not spam) e-mails. Furthermore, the classification of spam is focused on giving as little ham e-mails as possible the classification spam. In the e-Discovery context the aim is to give as little relevant e-mails as possible the classification 'not relevant'. Last but not least, for the problem of classifying spam e-mails as well as for the classification of relevant e-mails in a fraud related case unbalanced datasets are common.

With this reason the research described in this thesis is based on the hypothesis that the classification of relevant messages in an e-Discovery case is similar to the well known classification of spam in a mailbox. Looking at the state of the art applied in the spam classification gives a general idea of potentially useful models and algorithms. Various models that are mentioned in the following section and that are eventually used are explained mathematically in Chapter 2.

It is important to state the definitions that have been used for spam and fraudulent activities that are taken into account. Regarding the spam classification, in Chapter 3 it is stated that the TREC dataset is used. Therefore, although various definitions for spam are available, for this thesis the definition given by the TREC has been used: "'unsolicited', unwanted email that was sent indiscriminately, directly or indirectly, by a sender having no current relationship with the user".

Regarding the definition of fraudulent e-mails, the definition of the Dutch public prosecutor has been used. On the website they state that fraud is the intentional act of deceiving or gaining unlawful advantage [47].

1.3. State of the Art spam classification

There has been performed quite a lot of research and various reviews have been written on the classification of e-mail messages as ham and spam [5, 6, 8, 27, 54, 58, 62]. These reviews all draw almost the same conclusion, namely that the most used and useful classification algorithms for spam detection are (in no particular order):

- Naive Bayes (NB)
- Support Vector Machines (SVM)
- Logistic Regression (LR)
- Language specific classification

According to the research of Guzella and Caminhas [27], Zhang et al. [62], Moon et al. [42], C. Lai and M. Tsai [35], and Blanzieri and Bryl [8] each of the listed classification algorithms perform good in certain situations. Guzella and Caminhas [27] stated that Logistic Regression was equal to or even better in terms of performance if compared to the best known filters. Moreover, Moon et al. [42] stated that SVM used together with n-gram indexing is superior when compared to others in terms of performance. Whilst Lai and Tsai [35] only concluded that NB and SVM perform better than the k-Nearest Neighbour algorithm. Furthermore, several reports and especially Blanzieri and Bryl [8] mentioned that the Naive Bayes classifier works computationally efficient and has good rates of classification. It is therefore often used as benchmark by many researchers.

Moreover, in the spam classification various features are used. According to e.g. Guzella and Caminhas [27] and Bhowmick and Hazarika [6] the most used features are:

- Word frequencies: takes into account the number of times a word occurs in the e-mail,
- Word occurrences: takes the absence of words into account (checks whether a word is, present)
- E-mail length: the number of words or characters in an e-mail,
- Presence of attachments,
- Content type: MIME type of the message.

Many other features can be thought of and have been studied. According to the earlier cited research papers these features have been used most (with promising results) in the classification of spam. Based on these outcomes, three features have been used in this thesis, namely: Word frequencies, word occurrences and e-mail length. It should be remarked that Naive Bayes classifiers are mostly based and get best results with word frequencies or word occurrences [40]. Furthermore, Bhowmick and Hazarika mentioned that so called non-content features (features that are not based on the content of the e-mail) have also led to promising results. These are the reasons to choose for three different features for the classification of spam and fraud in this thesis.

1.4. Research questions

Based upon the observation that Naive Bayes is cited in most articles as benchmark performance, it is logical to start the research on classification of fraud related messages in Section 5.1 from that point. The most simple forms of Naive Bayes are best starting points, i.e. Multinomial Naive Bayes (word frequencies, see Section 5.1) and Multinomial Naive Bayes with boolean attributes (word occurrences, see Section 5.2), since these are most often used to benchmark performance.

Furthermore Naive Bayes might be a good alternative to the current used method of detecting fraud-related messages since it makes a distinction based on the (relative) frequencies of words used in the e-mails, this has some connection to the keyword searches that are used in an e-Discovery process. Models for detecting fraudulent financial statements or annual reports have already shown positive results, which makes the possibility of detecting fraud related e-mail look promising. The question remains whether there is an analogy between these statements/reports and e-mail messages.

Taking into account the goal of the thesis, the research question can be formulated as follows:

Can spam filtering techniques be used as viable techniques for detecting fraud related (i.e. relevant) e-mails?

Answering the research question is done by taking into account the following subquestions:

- What are similarities/differences between fraud and spam datasets/problems?
- What information is available in an e-mail dataset?
- What techniques have already been used in this setting?
- Are Bayesian models useful for classification purposes?

The conclusion will be positive if a model (or multiple models) have been found that are able to classify fraud related e-mail messages with a recall of the category relevant of 100% (whilst also labeling e-mails as not relevant). There will be potential in a model (or multiple models) if the recall of the category relevant is lower than 100% but above random (meaning the recall percentage that would be obtained by randomly assigning labels). In all other cases it will be concluded that the investigated models are not useful for a classification before the review process.

1.5. Structure of the thesis

At first an explanation of the available and used datasets is given in Chapter 3. After that the required mathematical preliminaries are discussed . In Chapter 4 an exploratory analysis of the available data is done, in order to check which features most likely give the best results. The mathematical background of the models of the various features is discussed in Sections 5.1-5.3. Each of these sections focuses on one feature, and the results of the classification based on each feature are stated in Chapter 6. After that an additional model, AdaBoost, is given in Chapter 7. This additional model is added based on the results found. Finally a discussion and conclusion is given of which feature and model works best, and further research is given as recommendation.

The structure of the thesis, as well as the influence of one chapter on another is visually presented in Figure 1.2.

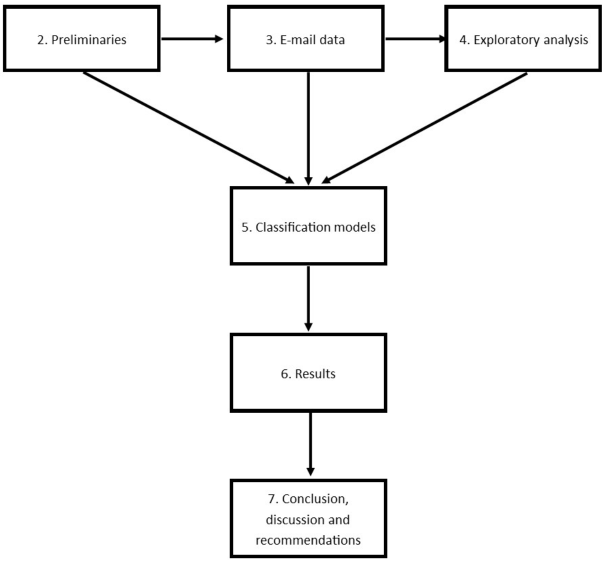


Figure 1.2: Structure of the report. Arrows indicate influence of one chapter on another.

At the start of the thesis, before the Table of Contents, lists of figures, tables, abbreviations and mathematical notation can be found. At the end of the thesis, after the bibliography, various appendices have been added. In these appendices additional results and more detailed mathematical explanations can be found. Reference to the appendices are given throughout the report. Furthermore, the used code can be found on Github[1].

[1] https://github.com/DKaak/Master-Thesis-TU-Delft/

2 Preliminaries

The previous chapter introduced the problem that is faced and gave the state of the art research on the topic discussed in this thesis. In this chapter the needed theoretical background information, as well as other theory that is used in the remainder of this research report is given. First the notion of classification is discussed, after that Bayesian Statistics is introduced. Moreover, Logistic Regression and Maximum Likelihood estimation are explained in Section 2.2.3 and 2.2.6. Finally the background information on various performance measures are given. A list of all basic background information of the used distributions throughout the report is given in Appendix A. Most theory discussed in this chapter can be found in more detail in the book written by Gelman [25] or Bishop [7]. Both these books have been used as reference for the explanations given in this chapter.

2.1. Classification

For the task of classification features $\boldsymbol{X}$ of a given instance are used in order to predict a label Y. Features known information about the instance (e.g. frequency of words, letters or whether a word can be found in the text), the label might for example be a binary variable to indicate whether the instance belongs to the class 'A' or 'B'. By using classification, it is typical that Y only takes discrete values. In the classification of spam and fraud related e-mails Y is in this research assumed to be binary.

In mathematical terms the concept of classification is formulated by finding the optimal function Φ such that (for $1 \leq j \leq J$ e-mails $\boldsymbol{z}^{(j)}$ and its classification $y^{(j)}$) if $\Phi(\boldsymbol{z}^{(j)}) = y^{(j)}$, it indicates that the e-mail $\boldsymbol{z}^{(j)}$ belongs to the category $y^{(j)}$. And the opposite is also true: When $\Phi(\boldsymbol{z}^{(j)}) \neq y^{(j)}$, it indicates that the e-mail $\boldsymbol{z}^{(j)}$ does not belong to the category $y^{(j)}$. The definition of most optimal function may be based on the context and differ per situation, i.e. in some cases it is better to have less false positives (spam) while in other cases it is better to have less false negatives (fraud).

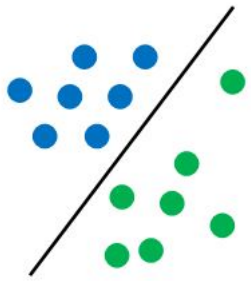


Figure 2.1: Visualisation of the idea of best possible classification. The circles represent e-mails, and the blue and green colour each represent a different class. The black line shows a possible classification.

2.2. Models

In the previous chapter various models and features that are mentioned in other research reports have been stated. Furthermore, in Chapter 5 a few of the features are used to create mathematical models. The mathematical theory of all the models mentioned in Section 1.1 are explained in general in this section as well as theoretical background information that is needed in the remaining chapters.

2.2.1. Generative vs. discriminative models

The two main approaches of statistical classification are generative classification and discriminative classification. Both approaches are useful to classify new instances, but the way the model learns to classify differs.

Given a variable X and outcome Y, the generative classification uses a generative model, which is based on the joint distribution $f(x,y)$. Whilst the discriminative model only uses the conditional distribution $f(y \mid x)$. This means that a generative model uses the distribution of the observed variables, and a discriminative model only uses the observed data.

One of the advantages of a generative model is that it is possible (as the name suggests) to generate new instances of the data which a similar to the existing data. The discriminative models make fewer assumptions, but are more influenced on the quality of the data. On the other hand it is possible to express a generative model analytically, whilst a discriminative model should often be approximated.

Examples of discriminative models are:

- Logistic regression
- Support vector machines
- Linear regression
- Neural Networks

Examples of generative models are:

- Naive Bayes
- Hidden Markov Model
- Latent Dirichlet Allocation

2.2.2. Bayesian Statistics

A field within statistics is related to Bayesian statistics. Bayesian statistics interprets probability as the degree of belief in an event. This belief is based on prior knowledge and update by observations. Bayesian statistics is based on Bayes' rule, which can be stated in mathematical notation.

Let X, Y be random variables. The probability density function of X is denoted as $f_X(x)$ or shorthand $f(x)$.The probability distribution function of X given Y is generally written as $f_{X|Y}(x|y)$ or shorthand $f(x|y)$.
Bayes' rule can then be expressed as

$$f_{X|Y}(x|y) = \frac{f_{Y|X}(y|x)f_X(x)}{f_Y(y)}. \tag{2.1}$$

In Bayesian statistics, to make inference on a parameter θ in the distribution of a random variable X, θ is assumed to be a random variable itself. A prior distribution $g(\theta)$ is assumed on θ. Using Bayes' formula the posterior distribution of θ given the data X can be found by:

$$g(\theta|x) = \frac{f(x|\theta)g(\theta)}{f(x)}. \tag{2.2}$$

Bayesian statistics can also be used to predict a new $\tilde{y}$ based on the observed information y. This posterior predictive distribution is given by

$$h(\tilde{y} \mid y) = \int f(\tilde{y} \mid \theta)g(\theta \mid y)d\theta. \tag{2.3}$$

It should be noted that in Bayesian statistics it is common to use Bayesian notation. In Bayesian notation all densities are denoted by p and the argument denotes both the random variable, as well as the value at which the density is evaluated, i.e., $p(x) = f_X(x)$ and $p(y \mid x) = f_{Y|X}(y|x)$.

Hierarchical models

In a Bayesian context it is possible to create hierarchical models. These type of models involve multiple parameters that are related to each other. It therefore implies that a joint probability model for these parameters reflects their dependence. In a hierarchical model the observable outcomes are modeled conditionally on certain parameters, which are specified based on so called hyper-parameters. The above formulation of a Bayesian model becomes a hierarchical model when in Equation (2.2) the distribution over θ is made dependent on for example hyper-parameter α, this would yield:

$$g(\theta \mid x, \alpha) = \frac{f(x \mid \theta, \alpha) g(\theta \mid \alpha)}{f(x \mid \alpha)}. \tag{2.4}$$

The distribution of α is called the hyper-distribution.

Conjugate priors

A full Bayesian model makes use of prior distributions on the parameters. The prior is often criticized by non-Bayesian statisticians, as it can be very subjective.

In order to take away this subjectivity as much as possible, a flat prior can be chosen. A flat prior includes as little subjective input as possible about the values the parameter should take.

Another possibility is to choose the prior to be a conjugate prior of the distribution over the data X. In Bayesian statistical models it is customary to use such a conjugate prior, since it simplifies the inference and therefore in most cases also computational complexity. Conjugate priors can sometimes also be chosen in such a way that they are flat priors.

Naive Bayes

In literature many researchers use a model to which they refer to as Naive Bayes. Naive Bayes is a generative classifier based on Bayes' Theorem:

$$\mathbb{P}(y \mid \boldsymbol{u}) = \frac{\mathbb{P}(\boldsymbol{u} \mid y)\mathbb{P}(y)}{\mathbb{P}(\boldsymbol{u})},$$

in which $\boldsymbol{u}$ represents the vector of features of the document and y represents the to be predicted class to which the document belongs to.

In the equation above, $\mathbb{P}(\boldsymbol{u})$ is the probability to observe the feature vector and $\mathbb{P}(y)$ the prior probability of a random document to belong to category y. The probability $\mathbb{P}(\boldsymbol{u} \mid y)$ is in Naive Bayes often assumed (hence the name Naive Bayes) to be

$$\mathbb{P}(\boldsymbol{u} \mid y) = \prod_{i=1}^{|\boldsymbol{u}|} \mathbb{P}(u_i \mid y).$$

With this assumption it is basically assumed that each feature occurs in an e-mail independently of each other feature, meaning that features do not influence the other features. In applications of Naive Bayes to e-mails word counts are often taken as feature. Although the assumption of words being independent of each other is often not true in real world data, various researchers have shown that the classifier performs quite good [21, 27, 40].

It should be noted that Naive Bayes is often not implemented as a full Bayesian model. Most implementations of Naive Bayes use a Maximum Likelihood Estimator. The distinction is therefore made between Naive Bayes implemented as a full Bayesian model and Naive Bayes implemented using a Maximum Likelihood Estimator. The former is also referred to as Extended Naive Bayes, the latter as Classical Naive Bayes.

Related Definitions and Theorems

Definitions related to Bayesian statistics that are used are:

Definition 1 (Posterior mean) *Let X be a random variable, and θ the parameter which is assumed to be a random variable itself. Then the posterior mean of θ is defined by:*

$$\mathbb{E}[\theta \mid X] = \int \theta p(\theta \mid x) d\theta.$$

Definition 2 (MAP Estimate) *A maximum a posteriori probability (MAP) estimate is an estimate that equals the mode of the posterior distribution. The MAP estimate can be used to obtain a point estimate of an unobserved quantity on the basis of empirical data. It is defined as:*

$$\hat{\theta}_{MAP} = \underset{\theta}{\arg\max} f(\theta \mid x).$$

Definition 3 (Exchangeability) *A finite set $X_1, \ldots, X_n$ of random quantities is said to be exchangeable if every permutation of $(X_1, \ldots, X_n)$ has the same joint distribution as every other permutation. An infinite collection is exchangeable if every finite subcollection is exchangeable.*

It should be noted that assuming exchangeability is a weaker assumption than assuming independence.

2.2.3. Logistic Regression

Logistic regression is a discriminative model and uses the following formula to determine the probability of an e-mail with feature vector $\boldsymbol{x}$ to belong to class y:

$$p(y|\boldsymbol{x}) = \frac{1}{1 + \exp(w_0 + \sum_{j=1}^{J} w_j x_j)}.$$

The probability of not belonging to class y thus becomes:

$$1 - p(y|\boldsymbol{x}) = \frac{\exp(w_0 + \sum_{j=1}^{J} w_j x_j)}{1 + \exp(w_0 + \sum_{j=1}^{J} w_j x_j)}.$$

The weights $\boldsymbol{w} = \left(w_1, \ldots, w_J\right)$ are based on the available training data, and can be calculated with various methods.

Available approximation methods are for example [52]:

- liblinear
- lbfgs
- newton-cg
- SAG

Of these approximation methods the first three have the major disadvantage that they are not faster for larger datasets. SAGA is a incremental gradient algorithm with fast linear convergence rates, and based on SAG. It has been shown that SAGA is one of the methods that is most efficient for high dimensional data [13], as is the case in the application described in this thesis. A detailed description of SAGA is given by Defazio et al. [13].

2.2.4. Support Vector Machines

A support-vector machine is a supervised learning algorithm that constructs a hyperplane in a high-dimensional space. This algorithm can then be used for classification. Support Vector Machines solve an optimization problem, which can in general be described as finding the hyperplane that has the largest distance to the nearest training-data point of any class (in general it holds that the larger the margin, the lower error of the classifier).

According to Zhang, Zhu and Yao [62] the optimization problem can be written as

$$\min_{\omega,\beta,\epsilon} \frac{1}{2} W^T \cdot W + C \sum_{i=1}^{M} \epsilon_i,$$

subject to $y_i(W^T \phi(x_i) + \beta) \geq 1 - \epsilon_i, \epsilon_i \geq 0$. Where $\{(x_1, y_1), \ldots, (x_M, y_M)\}$ is given training data. ϕ is a mapping function, W a weight vector, ϵ_i are slack variables and $C \geq 0$ a constant.

Support vector machines give a classification based on the outcome of the SVM. The outcome is the signed distance between the data point and the hyperplane. If this distance is positive the data point belongs to one class, if it is negative it belongs to the other class.

2.2.5. Language specific

Many implemented algorithms focus on specific features of the language, for example using N-grams (N-gram language models are based on the assumption that the probability of a certain word occurring at a certain position in a sequence depends only on the previous N-1 words), behaviour analysis (e.g. user's past activity and recipient frequency) or identifying e-mail authorship.

2.2.6. Maximum Likelihood Estimate

Maximum Likelihood estimation (MLE) is a method of estimating the parameters of a statistical model, given observations. As the name indicates, a maximization of the likelihood function is done. If the likelihood function is given by $\mathcal{L}(\theta; x)$, with θ the parameter and x the given data, then the maximum likelihood estimate is defined by

$$\hat{\theta} \in \left\{ \underset{\theta \in \Theta}{\arg\max} \mathcal{L}(\theta; x) \right\}.$$

Often the log-likelihood is used, i.e.

$$l(\theta; x) = \ln(\mathcal{L}(\theta; x)).$$

Note that the Maximum Likelihood Estimate is a point estimate.

2.3. Performance statistics

For the performance analysis of the classification of the different models there are various statistics available. The statistics that are used in this research are Bayes Factors and various specific performance measures.

2.3.1. Bayes Factors

In Bayesian statistics Bayes Factor is often used as an alternative to hypothesis testing. With Bayes Factor the certainty of the classification of one label over another can be expressed.

Bayes Factor (B) is defined by

$$B = \frac{p(y \mid H_0)}{p(y \mid H_1)} \tag{2.5}$$

in which y represents the to be predicted data, H_1 the alternative hypothesis and H_0 the null hypothesis.

According to Gelman et al. [25] Bayes Factors work well in the decision between models when they are discrete and when it makes sense to consider one model over another as good description of the data. Jeffreys [32] gave Table 2.1 for the interpretation of a Bayes Factor. Although the interpretations might be disputable, this interpretation of Bayes factors gives us the possibility to give some 'strength of evidence' [61].

Bayes factor B	Interpretation
$B > 1$	Evidence supports H_0
$1 > B > 10^{-\frac{1}{2}}$	Slight evidence against H_0
$10^{-\frac{1}{2}} > B > 10^{-1}$	Substantial evidence against H_0
$10^{-1} > B > 10^{-\frac{3}{2}}$	Strong evidence against H_0
$10^{-\frac{3}{2}} > B > 10^{-2}$	Very strong evidence against H_0
$10^{-2} > B$	Decisive evidence against H_0

Table 2.1: Interpretation of Bayes Factor, which gives the possibility to give 'strength of evidence' to the outcomes of Bayes Factor.

2.3.2. Measures of performance

Spam, as well as fraud, is in this research dealt with as a binary classification problem. A given e-mail can either get the classification spam/fraud/relevant (positive) or not spam/not fraud/not relevant/unlabeled (negative).

Furthermore:

- A true positive is defined as the outcome that should have been positive and is positive
- A true negative is defined as the outcome that should have been negative and is negative
- A false positive is defined as the outcome that is positive but should have been negative
- A false negative is defined as the outcome that is negative but should have been positive

		Actual Class	
		A	B
Predicted Class	A	True positives	False positives
	B	False negatives	True negatives

Table 2.2: Confusion matrix indicating the true/false positives/negatives.

In various research articles the same measures of filtering performance are used. Blanzieri and Bryl have listed one of the most comprehensive lists that correspond to a binary classification problem [8]. In table 2.3 the most useful performance measures that are used in this report have been summarised. These performance measures are first briefly explained. The number of true positives is denoted with n_{tp}, the number of true negatives with n_{tn}, the number of false negatives with n_{fn} and the number of false positives with n_{fp}.

Important performance measures for binary outcomes are precision and recall. Precision is defined as the ratio of true positives over the total number of positives:

$$\text{Precision} = \frac{n_{tp}}{n_{tp} + n_{fp}}.$$

The recall is defined as the ratio of true positives over the number of outcomes that should have been positive:

$$\text{Recall} = \frac{n_{tp}}{n_{tp} + n_{fn}}.$$

Furthermore, the accuracy is defined as the number of true outcomes over the total number of observations:

$$\text{Accuracy} = \frac{n_{tp} + n_{tn}}{n_{tp} + n_{fp} + n_{fn} + n_{tn}}.$$

It should be noticed that it is possible to get a good accuracy while having a high number of incorrect predictions for the smaller class.

The error rate is defined as the number of false outcomes over the total number of observations:

$$\text{Error rate} = \frac{n_{fn} + n_{fp}}{n_{tp} + n_{fn} + n_{fp} + n_{tn}}.$$

Note that the error rate is the same as one minus the accuracy.

The F-score considers both the precision and the recall, it is an average of the precision and recall and has its best value at 1 (and worst at 0).

$$\text{F-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

In the case of spam classification the most interesting performance measures are accuracy and ham recall. The accuracy shows how many e-mails are given a correct classification, and the ham recall shows how many ham e-mails are correctly given the ham classification. In this case the most interesting performance measure is the ham recall, because it is considered to be worse to give a ham e-mail the classification spam than the other way around. Intuitively, missed important information has a bigger impact than having to delete a spam e-mail manually from time to time.

In the case of fraud related e-mail classification, an interesting performance measure is the accuracy (with the same reasoning). The most interesting performance measure is, however, the recall of the 'relevant' e-mails. This is different to the case of spam classification, since this time the goal is to give a correct classification to as much relevant e-mails as possible, at first not minding very much if a lot of not relevant e-mails are given the classification as relevant. This is because it is not preferred to discard any important information, since there is an ability to go through unimportant information.

Measure	Formula
Accuracy	$\frac{n_{tp}+n_{tn}}{n_{tp}+n_{fp}+n_{fn}+n_{tn}}$
Error rate	$\frac{n_{fn}+n_{fp}}{n_{tp}+n_{fn}+n_{fp}+n_{tn}}$
Spam/Fraud/relevant recall	$\frac{n_{tp}}{n_{fn}+n_{tp}}$
Spam/Fraud/relevant precision	$\frac{n_{tp}}{n_{fp}+n_{tp}}$
Spam/Fraud/relevant F-score	$2 \cdot \frac{\text{precision}\cdot\text{recall}}{\text{precision}+\text{recall}}$

Table 2.3: Performance measures that are used throughout the report, based on the comprehensive list made by Blanzieri and Bryl [8]. The most important performance measure for fraud detection is the recall corresponding to the category relevant.

3
E-mail data

For the classification of spam as well as of fraud related messages, e-mail messages are taken into account as input data. To get a better overview of the available data and the used datasets, this chapter first focuses on the primary characteristics of e-mail. After that the used datasets (TREC, ENRON and a confidential dataset) are introduced. Together with the mathematics described in the previous chapter, the information in this chapter is used as background information for the analysis in the subsequent chapters.

3.1. E-mail characteristics

For e-mail data a general format is used throughout the Internet. Request for Comments (RFC) documents are publicly available and often serve as a way to describe standardized internet protocols. As per RFC2822 [49] a standard is described for the syntax of the transmission of electronic text messages. It is noted in the document that characteristics of the transmission of other structured data is described by Freed in RFC2049 [22]. Although it is out of the scope of this project to discuss all characteristics of the transmission of messages, various characteristics of e-mails that are used in the research later in the report are touched upon. A more detailed analysis is for example given in RFC2822 [49] and RFC2049 [22].

Generally e-mail is formatted with a header and body. The header must consist of the date on which the message was sent and the address of the sender. Other fields such as ‘to’, ‘cc’, ‘bcc’ and ‘subject’ are optional, but can be valuable for analysis. Fields that are marked as ’informational’, are: ‘subject’, ‘comments’ and ‘keywords’. These are, however, optional and it is noted in RFC2822 that the subject field is the most common to use. A summary of the way an e-mail message is structured is given in Figure 3.1.

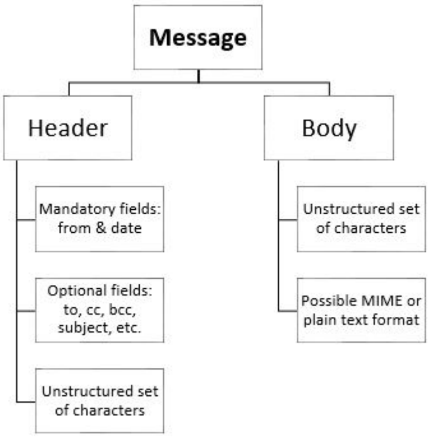
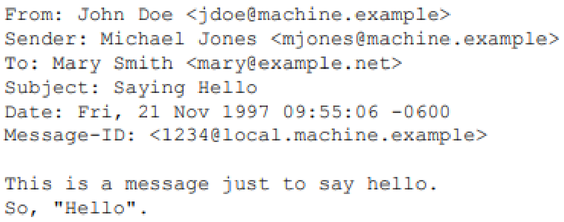

Message
Header
Body
Mandatory fields: from & date
Optional fields: to, cc, bcc, subject, etc.
Unstructured set of characters
Unstructured set of characters
Possible MIME or plain text format

Figure 3.1: Overview of the structure of an e-mail message. The fields that are always available are the mandatory from and date fields as well as the unstructured set of characters in the body (which might be empty).

If the body is just a text message it contains an unstructured set of ASCII characters. It is, however, possible that the body is formatted in a different way. For example bodies based on Multipurpose Internet Mail Extensions (MIME) extend the standard kind of message bodies. MIME allows the format of messages to include:

- textual messages
- non-textual messages, i.e. images, audio, video, html, etc.
- multi-part messages: messages that (may) contain multiple different kinds of body. This allows for parallel display of more objects (e.g. a picture and an audio fragment)

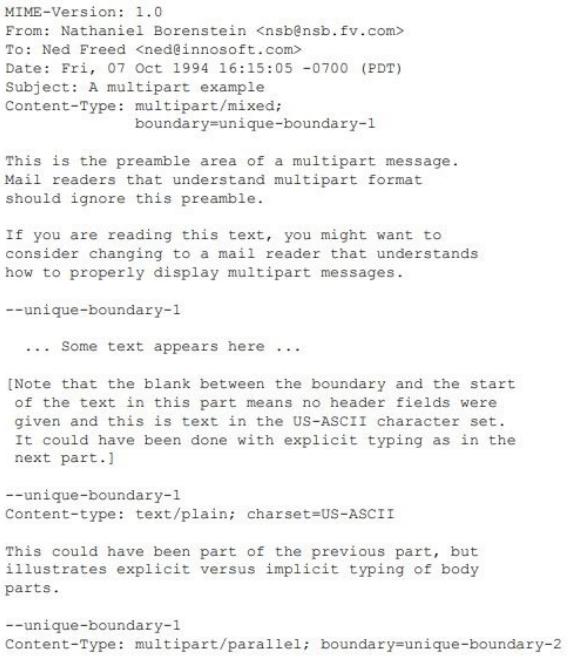
In Figure 3.2 an example e-mail message of text format is shown [49]. Figure 3.3 shows a multi-part MIME formatted e-mail message [22]. Both figures are taken from the corresponding RFC documents.



```
From: John Doe <jdoe@machine.example>
Sender: Michael Jones <mjones@machine.example>
To: Mary Smith <mary@example.net>
Subject: Saying Hello
Date: Fri, 21 Nov 1997 09:55:06 -0600
Message-ID: <1234@local.machine.example>

This is a message just to say hello.
So, "Hello".
```

Figure 3.2: Sample plain text e-mail message [49]. In this sample all mandatory fields summarised in Figure 3.1 can be seen.



```
MIME-Version: 1.0
From: Nathaniel Borenstein <nsb@nsb.fv.com>
To: Ned Freed <ned@innosoft.com>
Date: Fri, 07 Oct 1994 16:15:05 -0700 (PDT)
Subject: A multipart example
Content-Type: multipart/mixed;
              boundary=unique-boundary-1

This is the preamble area of a multipart message.
Mail readers that understand multipart format
should ignore this preamble.

If you are reading this text, you might want to
consider changing to a mail reader that understands
how to properly display multipart messages.

--unique-boundary-1

  ... Some text appears here ...

[Note that the blank between the boundary and the start
 of the text in this part means no header fields were
 given and this is text in the US-ASCII character set.
 It could have been done with explicit typing as in the
 next part.]

--unique-boundary-1
Content-type: text/plain; charset=US-ASCII

This could have been part of the previous part, but
illustrates explicit versus implicit typing of body
parts.

--unique-boundary-1
Content-Type: multipart/parallel; boundary=unique-boundary-2
```

Figure 3.3: Sample multi-part e-mail message based on the MIME format [22]. Especially the way different parts of the multi-part e-mail message are separated is important to take into account when analysing the e-mail messages.

As can be seen from the two example e-mail messages, the major difference between the plain text e-mail and the multi-part formatted e-mails is boundary between the various parts of the e-mail. For later analysis of spam and relevent fraud related e-mail it is good to keep in mind that an e-mail can consist of a plain text part and a html text part that both contain the same information. Furthermore, if a message has the structure multi-part any unreadable attachments might have to be discarded.

3.2. Data collection and preparation

As is stated in the Introduction, the research on fraud classification is started by examining the state of the art spam classification. In the following subsections more details on the used datasets are given, both for the spam and fraud related e-mail classification. As is stated earlier, there is more information and data available corresponding to spam than to fraud. This has mainly to do with the concerns of privacy in fraud-related cases and private companies doing the research.

3.2.1. Spam

For the analysis of spam already quite a lot of research is done (see also Chapter 1.1). There are various datasets on which this state of the art research has been based [43], the most cited are:

- Ling-Spam
- SpamAssassin
- PU corpora
- TREC

Each dataset has its own advantages and disadvantages, since each of the datasets are created differently. Using mails in public analysis threatens privacy, and therefore it is hard to find or create a dataset of a real life mailbox that includes spam.

Ling-Spam tries to overcome this burden by using messages from the publicly available archives of Linguist list. There are 2893 messages in the corpus (2412 ham, 481 spam). As Androutsopoulos et al. [40] state the disadvantage of this dataset is that the ham messages are more specified to certain topics than the messages in most mailboxes. This might cause a performance that is too optimistic. Furthermore, since the number of e-mails in the corpus is relatively small, this might also influence results[50].

For the SpamAssassin dataset this disadvantage works the other way around. Since the ham messages are included from multiple different users, the topics might be too diversified leading to unrepresentative results.

The PU corpora tries to solve the disadvantages of not using a real world mailbox by using encrypted personal emails [1, 2]. In this way the content is not publicly available, whilst the mails are, to some extent, usable. The disadvantage with this dataset is that the original words are not available, and therefore no context or length of documents/words can be used.

The TREC is based on chronically ordered e-mail and is labeled by 'human-adjudicated gold standard' [11]. This gold standard means that based on human adjudication each e-mail message is labeled spam or not spam. In general, the gold standard is assumed to be the truth. It should be noted that a human process of labeling is prone to errors. Since these are real life messages, with the classification of people, this dataset would be the most representing reality and further information about this dataset is given in the next section.

TREC

As is stated in the previous section, the TREC Public Corpus [57] is one of the best representing reality. It is also one of the datasets on which the most research is available. Although there are 3 different corpora, the 2007 corpus is the most recent, and thus the most applicable to today's data. This corpus contains of 75419 messages, of which:

- 25220 ham
- 50199 spam

Furthermore, the corpus is dived into 3 subcorpora:

- Full: all messages
- Delay: contains only the first 10,000 messages
- Partial: contains the 30,388 messages of 1 recipient

For the analysis of spam classification in the subsequent chapters the full TREC 2007 corpora has been used. The TREC 2007 dataset is based for almost half on one user, therefore the disadvantage of being too specific might still be in place. However the dataset is larger when compared to the other available datasets.

TREC data preparation

All the e-mails in the TREC corpora are text files in the MIME format (which is explained in Section 3.1). Therefore, no decryption or change of file type is needed in order to read these files. However, some preparation of the documents needed to be done before the analysis can be done or models can be applied.

For the pre-processing the following pre-process specific python modules are used:

- BeautifulSoup [36]
- email [48]

The following list of tasks to each e-mail has been applied (the corresponding code can be found on GitHub), which is also visually presented in Figure 3.4:

1. Check whether the e-mail contains an ’text/html’ part.
   - If this is the case use the html parser from BeautifulSoup, and continue with step 2
   - If this is not the case, check if the e-mail contains an ’text/plain’ part.
     - If this is the case use that part for the analysis, and continue with step 2
     - If this is not the case, the e-mail does not contain any relevant information, and continue with the next e-mail, step 1.
2. Either from the parsed html text or from the plain text any tabs or enters are removed and all text is made lower case
3. From the remaining message any characters which have an ASCII number below 97 or above 122 are removed
4. The remaining message is then split into words (the space is used as split character)
5. The required information (e.g. number of occurrences of each unique word) in the e-mail message is stored. The used features are listed in Section 1.3

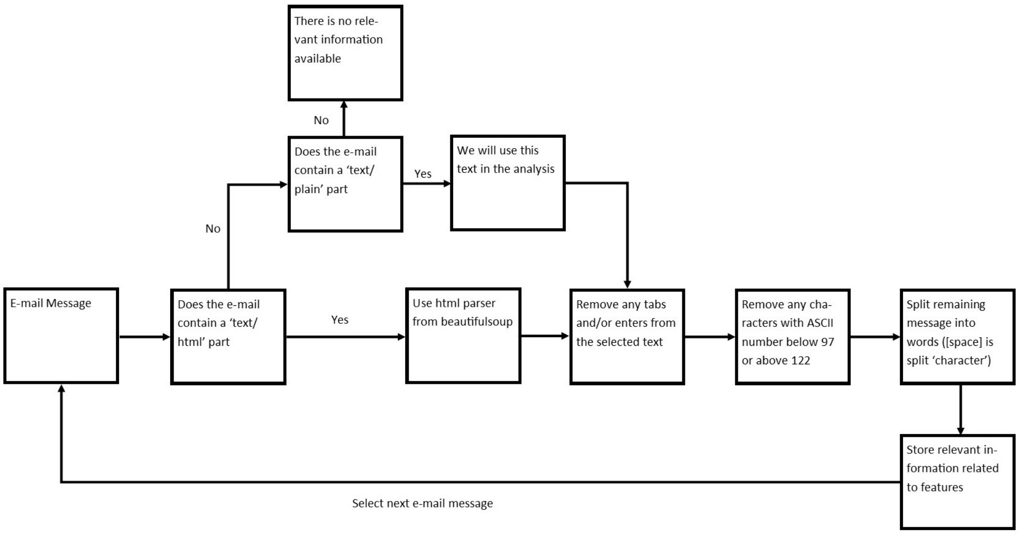

There is no relevant information available
No
Does the e-mail contain a ‘text/ plain’ part
Yes
We will use this text in the analysis
No
E-mail Message
Does the e-mail contain a ‘text/ html’ part
Yes
Use html parser from beautifulsoup
Remove any tabs and/or enters from the selected text
Remove any characters with ASCII number below 97 or above 122
Split remaining message into words ([space] is split ‘character’)
Store relevant information related to features
Select next e-mail message

Figure 3.4: Flow diagram of the pre-processing TREC data. The diagram shows how the features are extracted from the e-mail messages. It is important to note that any characters other than normal letters are removed.

After data preparation the remaining information is randomly assigned to five different batches. These batches are used for cross validation. Table 3.1 shows the statistics of each of the batches.

	Set 1	Set 2	Set 3	Set 4	Set 5
Spam	65.7%	65.0%	65.9%	66.2%	65.6%
Ham	34.3%	35.0%	34.1%	33.8%	34.4%
Total (amount)	15000	15000	15000	15000	15000

Table 3.1: Information on the five batches used performance checks of the TREC dataset.

3.2.2. Fraud

As is already stated earlier, the number of available datasets for analysis of fraud-related e-mail messages is low. The only publicly available and most used and cited dataset is the ENRON dataset, which is based on a major fraud case in 2001 related to the ENRON company. The reason there are not many datasets available is the nature of the data. Most fraud cases in which an analysis of e-mail messages is done are specifically focused on only a few number of people, which makes the data not suitable for public analysis due to privacy. In the following sections the public available ENRON dataset is explained, and as much detail as possible on the other used datasets is given.

ENRON

The ENRON fraud case is one of the largest fraud scandals in United States history. ENRON was an American energy company, and has declared bankruptcy in December 2001. Various employees have had trials related to e.g. fraud, money laundering, inside trading, etc. Two main suspects were Enron's former chief executive, Jeffrey Skilling and the ex-chairman Kenneth Lay [38].

Since the Federal Energy Regulatory Commission (FERC) has made the database of mails public, it is one of the biggest publicly available datasets of real world data. This dataset consists of 517401 documents (e-mails, calender items and attachments) from 150 different persons [14]. Many researchers have used this database to analyse behaviour, spam and fraud [4, 28, 39, 51, 55].

Besides the available dataset that consists of the mails of various employees, the United States Department of Justice has also released numerous document related to the ENRON trials of the chief executive and the ex-chairman [56, 59]. These available documents have been used to make a labeled ENRON dataset (with labels indicating if an e-mail is related to fraud, and therefore 'relevant', or otherwise 'unlabeled'). Every e-mail that is published on the archive page of the United States Department of Justice webpage [56] is marked as a relevant e-mail in the ENRON database. This way of labeling is based on the idea that during an e-Discovery investigation it would be best to label an equal amount or more e-mails as related to fraud than will eventually be used in a court case. This also means that many more e-mails in the remaining (unlabeled) part of the dataset might be relevant as well. It should be kept in mind that this (the data quality) can have influence on the classification performance.

ENRON data preparation

The latest available ENRON dataset has been used from the EDRM [14], which is the v1 data set cleansed of private, health and financial information. The methodology used in this cleansing process is described in a report written by Nuix and EDRM [46]. It consists of the mailboxes of 130 ENRON employees (in .msg file type). It should be noted that ENRON had many more employees, but only these 130 mailboxes are nowadays publicly available, the reason why other mailboxes are not (any more) available is not known. The available mailboxes do include the people that were part of the higher management.

As is stated in the previous section, the e-mails published on the United States Department of Justice webpage [56] have been used to create a labeled dataset. The trials on this webpage are related to Kenneth Lay and Jeffrey Skilling. The mailboxes of Jeffrey Skilling are unfortunately not part of the public available EDRM dataset, but the mailbox of various other employees mentioned on this webpage including Kenneth Lay are. On the webpage of the United States Department of Justice 123 e-mails are published. For each of these e-mails information of the sender and the addressees of the e-mail have been gathered. The mailboxes of the people that are in the public EDRM dataset and are on this list of names are used to search for the identified mails. For clarification purposes, the process described in this paragraph is also visually summarised in Figure 3.5.

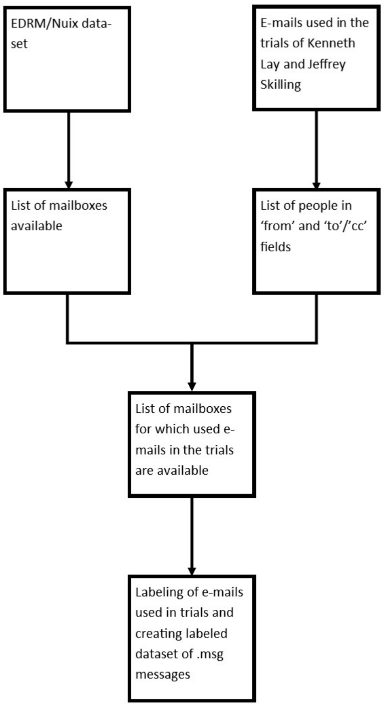

EDRM/Nuix data-set
E-mails used in the trials of Kenneth Lay and Jeffrey Skilling
List of mailboxes available
List of people in 'from' and 'to'/'cc' fields
List of mailboxes for which used e-mails in the trials are available
Labeling of e-mails used in trials and creating labeled dataset of .msg messages

Figure 3.5: Flow diagram of the pre-processing ENRON trial messages to create a labeled dataset. As has been noted in the text no labeled ENRON dataset is available. This diagram shows how the labeled dataset is created, based on the trials of Kenneth Lay and Jeffrey Skilling (former employees of ENRON).

All the e-mails in the useful mailboxes of the ENRON dataset are .msg files with text format (which is explained in Section 3.1). Therefore, it is again possible to read them directly with the right packages. However, some preparation of the documents needed to be done before the analysis can be done or models can be applied.

For the pre-processing the python module aspose.email.mapi [3] is used.

The following list of tasks has been applied to each e-mail (the corresponding code can be found on GitHub), which is also visually presented in Figure 3.6:

1. Check whether the e-mail is empty (except for the final sentence which is added by Nuix and EDRM, and which is removed from every e-mail in step 3).
   - If the e-mail does not contain text it is discarded
   - If the e-mail does contain text, continue with step 2
2. From the .msg file any tabs or enters are removed and all text is made lower case. Furthermore remove the following sentence which is present in all e-mails:
   ************ EDRM Enron Email Data Set has been produced in EML, PST and NSF format by ZL Technologies, Inc. This Data Set is licensed under a Creative Commons Attribution 3.0 United States License <http://creativecommons.org/licenses/by/3.0/us/> . To provide attribution, please cite to "ZL Technologies, Inc. (http://www.zlti.com)." ************
3. From the remaining message any characters which have an ASCII number below 97 or above 122 are removed
4. The remaining message is then split into words (the space is used as split character)

5. The needed information (e.g. number of occurrences of each unique word) in the e-mail message is stored. The used features are listed in Section 1.3

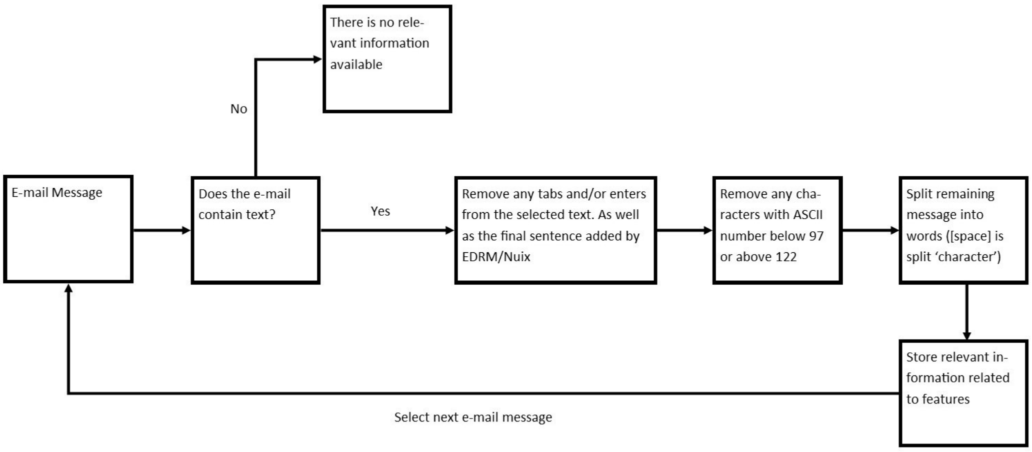

There is no relevant information available
No
E-mail Message
Does the e-mail contain text?
Yes
Remove any tabs and/or enters from the selected text. As well as the final sentence added by EDRM/Nuix
Remove any characters with ASCII number below 97 or above 122
Split remaining message into words ([space] is split 'character')
Store relevant information related to features
Select next e-mail message

Figure 3.6: Flow diagram of the pre-processing ENRON data. The diagram shows how the features are extracted from the e-mail messages. It is important to note that any characters other than normal letters are removed.

After data preparation the remaining information is randomly assigned to five different batches. These batches are used for cross validation. Table 3.2 shows the statistics of each of the batches.

	Set 1	Set 2	Set 3	Set 4	Set 5
Relevant	1.7%	1.5%	1.8%	1.0%	1.5%
Not relevant	98.3%	98.5%	98.2%	99.0%	98.5%
Total (amount)	1621	1621	1621	1621	1621

Table 3.2: Information on the five batches used for cross validation of the ENRON dataset.

Confidential dataset

In order to tackle the problem of the limited number of datasets available and to test the method in the "real world", a confidential dataset provided by KPMG is used. Due to legal rules and privacy concerns not much information is allowed to be shared. This might not be in favour of academic research, in which results and methods are presented in a transparent way, but in order to test whether the models will work in real cases there is no alternative choice at this moment in time.

It can be stated that the labeling of the relevant (fraud-related) e-mails is done based on expert review, i.e. experts have manually labeled the e-mails in the dataset.

All the e-mails in the useful mailboxes of the confidential dataset are .msg files with text format (which is explained in Section 3.1). Therefore, it is again possible to read them directly with the right packages. However, some preparation of the documents needed to be done before the analysis can be done or models can be applied.

For the pre-processing the python module aspose.email.mapi [3] is used.

The following list of tasks has been applied to each e-mail (the corresponding code can be found on GitHub), which is also visually presented in Figure 3.7:

1. Check whether the e-mail is empty.

- If the e-mail does not contain text it is discarded
- If the e-mail does contain text, continue with step 2

2. From the .msg file any tabs or enters are removed and all text is made lower case.
3. From the remaining message any characters which have an ASCII number below 97 or above 122 are removed
4. The remaining message is then split into words (the space is used as split character)
5. The needed information (e.g. number of occurrences of each unique word) in the e-mail message is stored. The features used are listed in Section 1.3

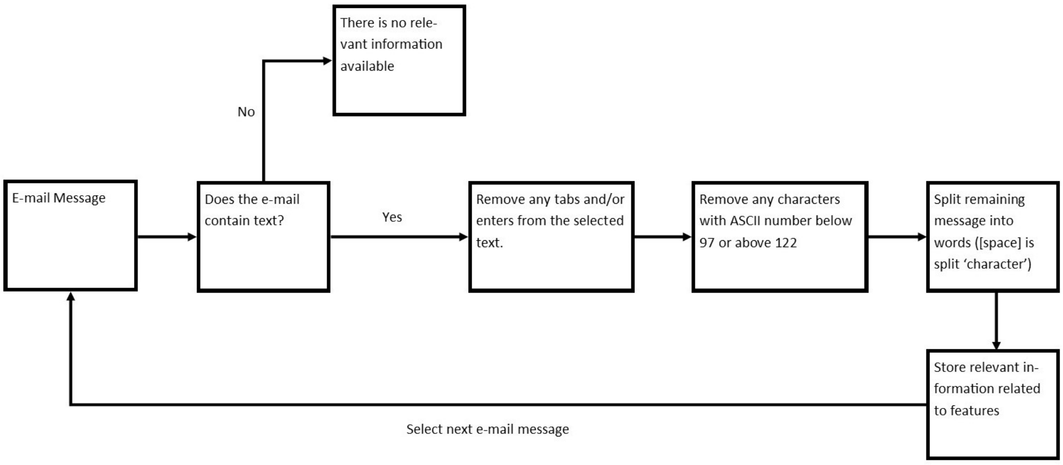

There is no relevant information available
No
E-mail Message
Does the e-mail contain text?
Yes
Remove any tabs and/or enters from the selected text.
Remove any characters with ASCII number below 97 or above 122
Split remaining message into words ([space] is split 'character')
Store relevant information related to features
Select next e-mail message

Figure 3.7: Flow diagram of the pre-processing confidential data. The diagram shows how the features are extracted from the e-mail messages. It is important to note that any characters other than normal letters are removed.

After data preparation the remaining information is randomly assigned to five different batches. These batches are used for cross validation. Table 3.3 shows the statistics of each of the batches.

	Set 1	Set 2	Set 3	Set 4	Set 5
Relevant	8.0%	5.0%	7.8%	5.5%	7.8%
Not relevant	92.0%	95.0%	92.2%	94.5%	92.2%
Total (amount)	399	399	399	399	399

Table 3.3: Information on the five batches used for cross validation of the confidential dataset.

Results based on the classified datasets have been included in the report in a similar manner as the results of ENRON en TREC are presented. However, no word specific analysis, or word specific references are included due to the confidentiality.

4

Exploratory data analysis

In order to test whether the earlier mentioned features (based on literature study, word frequencies, word occurrences and e-mail length) are feasible, an analysis of the available data has been conducted. In Section 4.1 the data of the TREC dataset has been analysed. In section 4.2 and 4.3 a similar analysis is presented, but then on the ENRON and confidential dataset respectively.

4.1. Spam analysis

The differences in spam have been modeled most often by using features related to word usage. For example by looking at the different words used in spam and ham e-mails.

4.1.1. Word counts and occurrences

Although not all unique words that are present in the TREC dataset can be analysed, some statistical analysis on the words that are highly identifiable for spam and ham has been done. In Table 4.1 the top 10 words that are identifiable for ham e-mail in the TREC dataset are shown. In this table for both categories both the unnormalised counts are shown, these are the number of times the word occurs in spam and ham e-mails respectively. The last column shows a ratio of the normalised counts per message. This ratio is calculated according to Equation (4.2). This ratio is based on the counts per message (CPM) as explained in Equation (4.1), in which $x_j^{(m)}$ is the number of times word j occurs in e-mail m. In this equation y indicates the category for which the counts per message are calculated, J the total number of unique words and $n^{(m)}$ is the number of words in e-mail m. Intuitively this count per message is the number of times the word occurs on average per message within the category, and 1 count for every word and J the total number of words is added in order to avoid division by 0 in Equation (4.2), this modification will also be used in the Maximum Likelihood Estimation described in Chapter 5.1.

$$CPM_j^y = \frac{\sum\limits_{m=1,\ldots,N:y^{(m)}=y} x_j^{(m)} + 1}{\sum\limits_{m=1,\ldots,N:y^{(m)}=y} n^{(m)} + J}. \tag{4.1}$$

$$Ratio_j = \frac{CPM_j^{spam}}{CPM_j^{ham}}. \tag{4.2}$$

Word	Count spam	Count ham	Ratio counts per message ($\cdot 10^{-5}$)
reproducible	0	6642	7.58
ntstatus	0	2690	18.7
speakup	0	2463	20.4
uint	0	1741	28.9
committer	0	1468	34.2
ndr	0	1401	35.9
revno	0	1278	39.4
zonk	0	873	57.6
tridge	0	759	66.3
accuweather	0	648	77.6

Table 4.1: Top 10 words which are identifiable for ham e-mail messages from TREC dataset according to the ratio of words per message. As can be noted these 10 words are only used in the ham e-mails, and therefore these words are assumed to be representative for the category ham e-mails. The ratio of counts per message shows how the words are sorted.

Table 4.1 shows us that there are quite a number of words that are used more often in ham e-mails than in spam e-mails. Table 4.2 shows similar values, but for words that are identifiable with spam e-mails. From both these tables it is suspected that there is an possibility to distinguish spam from ham e-mails based on word frequencies. Furthermore, it should be noted that almost all words presented in the tables only occur in one of the categories, which might imply that it should be able to distinguish spam and ham e-mails by looking at the word occurrences. Another observation is that the words in Table 4.1 are mainly English words whilst the words in Table 4.2 are mainly French. This is indeed true, although English words are also identifiable (but not in the top 10) as spam words, and vice versa.

Word	Count spam	Count ham	Ratio counts per message
desjardins	31244	1	7866.06
accsd	6787	0	3417.81
anatrim	4526	0	3084.99
mouvement	4516	0	2274.99
scuris	2613	0	1316.17
soyez	1610	0	1314.66
comptable	2509	0	1314.16
caisses	2411	0	1214.46
membre	2400	0	1208.92
svp	2365	0	1191.3

Table 4.2: Top 10 words which are identifiable for spam e-mail messages from TREC dataset according to the ratio of words per message. As can be noted these 10 words are mainly used in the spam e-mails, and therefore these words are assumed to be representative for the category spam e-mails. The ratio of counts per message shows how the words are sorted.

4.1.2. Length of e-mail

As can be seen in Figure 4.1, but even better in Table 4.3, there is not much, but still a little difference in terms of the length of spam e-mails versus the length of ham e-mails. Figure 4.1 shows the boxplots of the logarithm of the lengths of the e-mails per category. Although the differences are small between the categories, note that in general there are more smaller and bigger ham e-mails than spam e-mails. This might imply that especially for the outlying lengths of e-mails a good classification can be given.

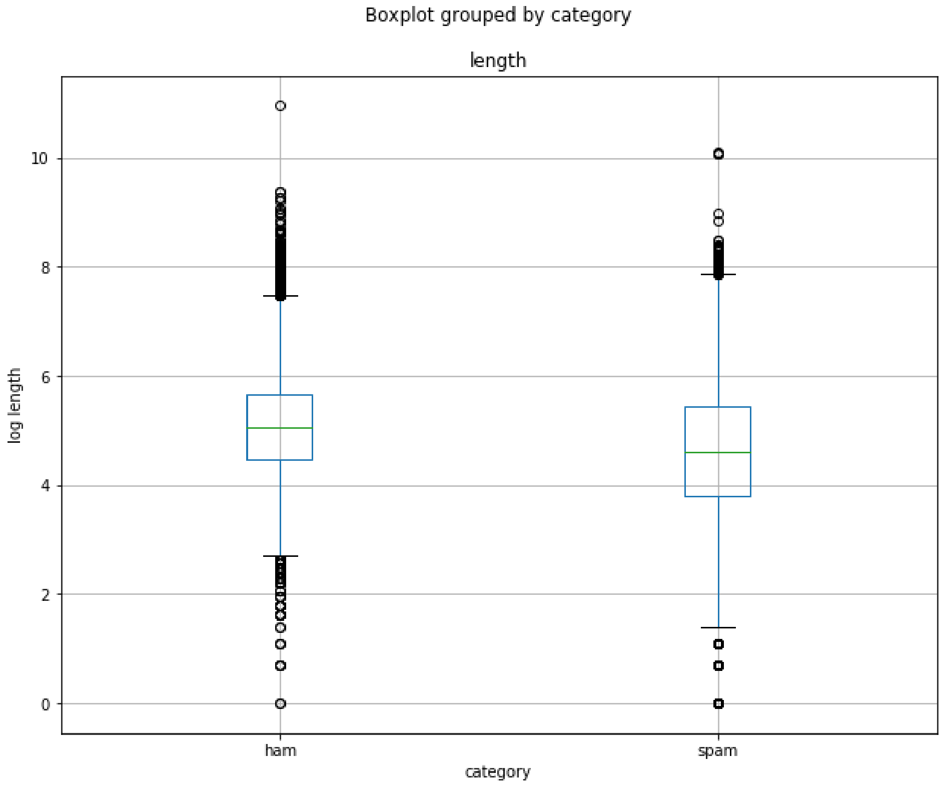


Figure 4.1: Boxplot of length spam e-mails TREC (in which the length is defined as the number of words). As can be noted the lengths are more or less similar, although especially the outlying values of the ham category might be useful for classification.

Table 4.3, gives more information. As can be seen, the mean as well as the standard deviation are quite different between the two categories. This also gives us the impression that these differences might be of value to distinguish spam and ham e-mails. Furthermore, Figures 4.2-4.3 show histograms of the lengths of the e-mails shorter than 1000 words (if the longer e-mails are also presented, no informative figure can be made). As can be noted from these figures it is expected that the lengths of the e-mails follow a Pareto distribution. This distribution is often used for lengths, and it is therefore expected that this is observed [44].

	Spam	Ham
Mean	175.91	287.91
Standard deviation	335.34	643.81
Q1	45	88
median	101	159
Q3	231	293

Table 4.3: Basic statistical features of lengths of e-mails from TREC dataset (divided into the classification spam and ham)

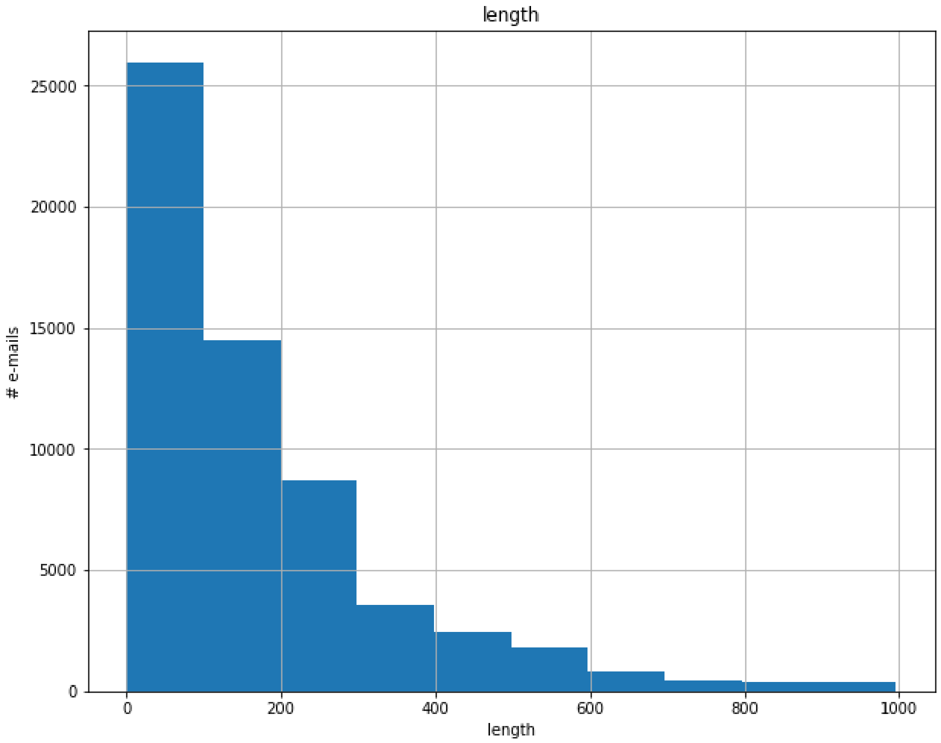


Figure 4.2: Histogram of length e-mails TREC dataset (in which length is defined as the number of words). As can be noted many e-mails shorter than 5000 words are present. Figure E.2 shows the same histogram but only for the smaller e-mail messages.

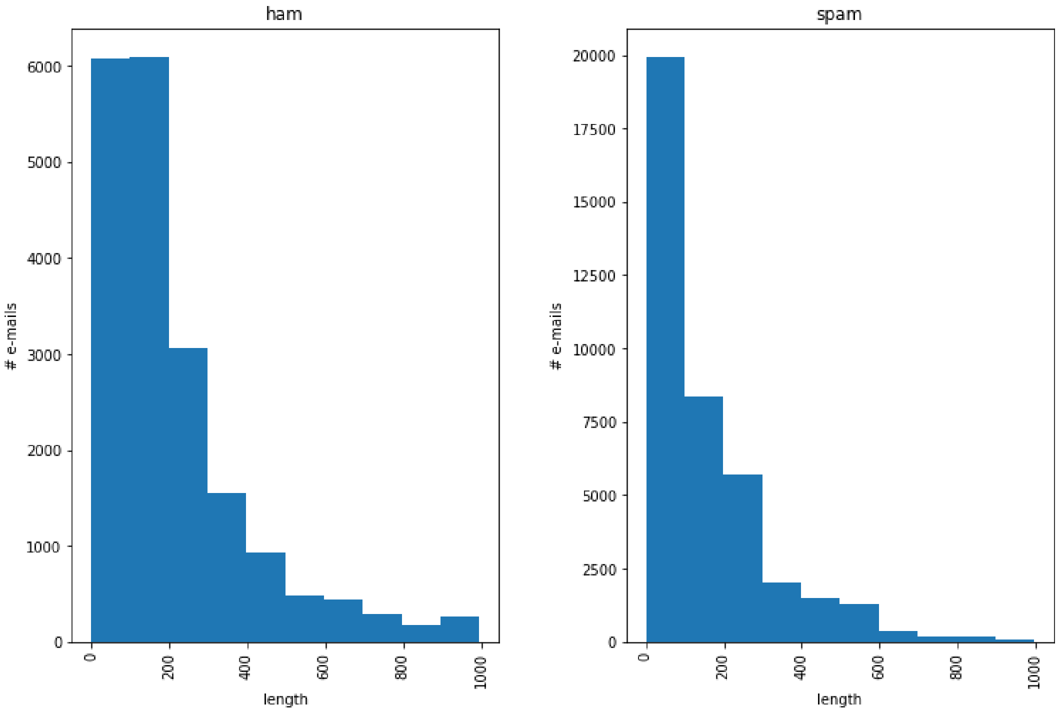


Figure 4.3: Histogram of length e-mails TREC dataset, shown for both categories (in which length is defined as the number of words). As can be seen both categories are distributed similarly.

4.2. Fraud analysis ENRON

As has been explained in Section 1.1.3, several research papers have found ways to identify fraudulent financial statements and annual reports. The models that were found in the research reports were mostly based on linguistic features, i.e. usage of words. In this section the same analysis will be done as in the previous section on spam. Based on the conclusions from other research reports, it is expected to find some differences in the word counts between relevant (fraud related) and unlabeled (possibly legit or fraud related) e-mails.

4.2.1. Word counts and occurrences

In a similar manner as is described in Section 4.1.1 the e-mails in the labeled ENRON dataset have also been analysed. Although the ENRON dataset is limited in size (i.e. number of e-mails), some careful conclusions have been made. In Table 4.5 the 10 words that are most identifiable for relevant e-mail are shown. Table 4.4 shows the 10 words that are most identifiable for unlabeled e-mail. The values in the column in which the ratio counts per message is shown are calculated according to Equations (4.1) and (4.2). From both these tables it can be concluded that there are again various differences in word usage, however the difference is smaller than in the similar spam analysis. This smaller difference might lead to results that have a lower performance than in spam classification. Looking at the specific words in the tables, it should be noted that there are a number of specific ENRON related words in Table 4.5. It should be kept in mind that these words will most likely not be useful if a model will be trained with the labeled ENRON dataset and apply the model to another case, because words that are company specific are (often) not used by other companies and therefore not present in other cases.

Word	Count relevant	Count not labeled	Ratio counts per message
consumers	0	2795	0.023
bankruptcy	0	1945	0.033
retirement	0	1860	0.034
bills	0	1856	0.034
declared	0	1812	0.035
donate	0	1808	0.035
basic	0	977	0.065
americans	0	975	0.066
buying	0	958	0.067
thousands	0	951	0.067

Table 4.4: Top 10 words which are identifiable for not labeled e-mail messages from ENRON dataset according to the ratio of words per message. As can be noted these 10 words are only used in the unlabeled e-mails, and therefore these words are assumed to be representative for this category. The ratio of counts per message shows how the words are sorted.

Word	Count relevant	Count not labeled	Ratio counts per message
epe	23	1	767.52
developmentenron	8	0	575.64
ccbn	8	0	575.64
pref	6	0	447.72
holdco	5	0	383.76
barone	4	0	319.8
fraudulent	4	0	319.8
deconsolidate	4	0	319.8
ivers	4	0	319.8
writedowns	4	0	319.8

Table 4.5: Top 10 words which are identifiable for relevant e-mail messages from ENRON dataset according to the ratio of words per message. As can be noted these 10 words are mainly used in the relevant e-mails, and therefore these words are assumed to be representative for this category. The ratio of counts per message shows how the words are sorted. Moreover, it should be mentioned that the words are not used many times (i.e. counts are low), this might have an influence on the classification results.

4.2.2. Length of e-mail

As can be seen in Figure 4.4 as well as in Table 4.6, there is a bigger difference in terms of length between relevant and not labeled e-mails when compared to the difference in lengths between spam and ham e-mails (as has been done in the previous section). Figure 4.4 shows the logarithm of the length of the e-mails in the two different categories, whilst in Table 4.6 the original values are presented. Furthermore, Table 4.6 shows us that especially the standard deviation of relevant mails is higher.

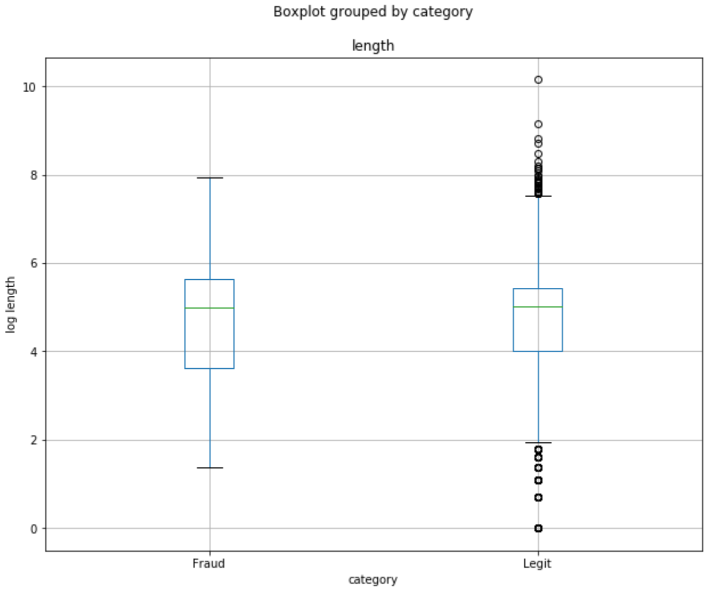


Figure 4.4: Boxplot of the logarithm of the length of e-mails of the two categories in the ENRON dataset (in which the length is defined as the number of words). As can be noted the lengths are more or less similar, although especially the outlying values of the unlabeled (Legit) category might be useful for classification.

	Relevant	Not Labeled
Mean	237.62	214.80
Standard deviation	374.27	324.97
Q1	42.5	55
median	139	152.5
Q3	281	227

Table 4.6: Basic statistical features of lengths of e-mails from ENRON dataset (divided into the classification relevant and unlabeled). From this table it can be concluded that not labeled e-mails have a smaller standard deviation, when compared to the relevant e-mails.

In order to get a better view on the distribution the lengths of e-mails, some histograms have been added. Figure 4.5 shows the histogram of all e-mails regardless of the category. This histogram includes a very long e-mail. Figure E.1 gives a better view of the distribution over the e-mails that are shorter in length, for this figure the e-mails with a maximum length of 1000 words have been used. Figure 4.6 (and also Figure E.2 for the smaller e-mails only) shows the histogram of all e-mails in both categories. Based on these figures it would be suspected that the length of e-mails, in terms of number of words, follow a Pareto distribution.

Based on the analysis done in this section, it can again be concluded that no major differences are present between the categories relevant and not labeled in terms of the length of e-mails. However, especially for small and large e-mails small differences are spotted.

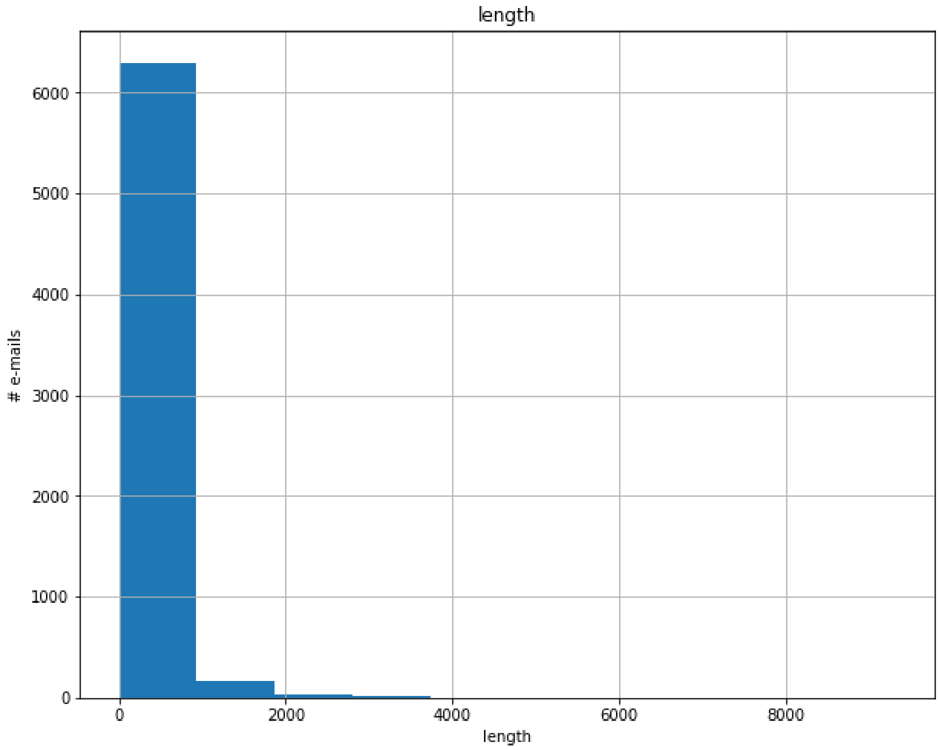


Figure 4.5: Histogram of length e-mails ENRON dataset (in which length is defined as the number of words). As can be noted many e-mails shorter than 2000 words are present. Figure E.2 shows the same histogram but only for the smaller e-mail messages.

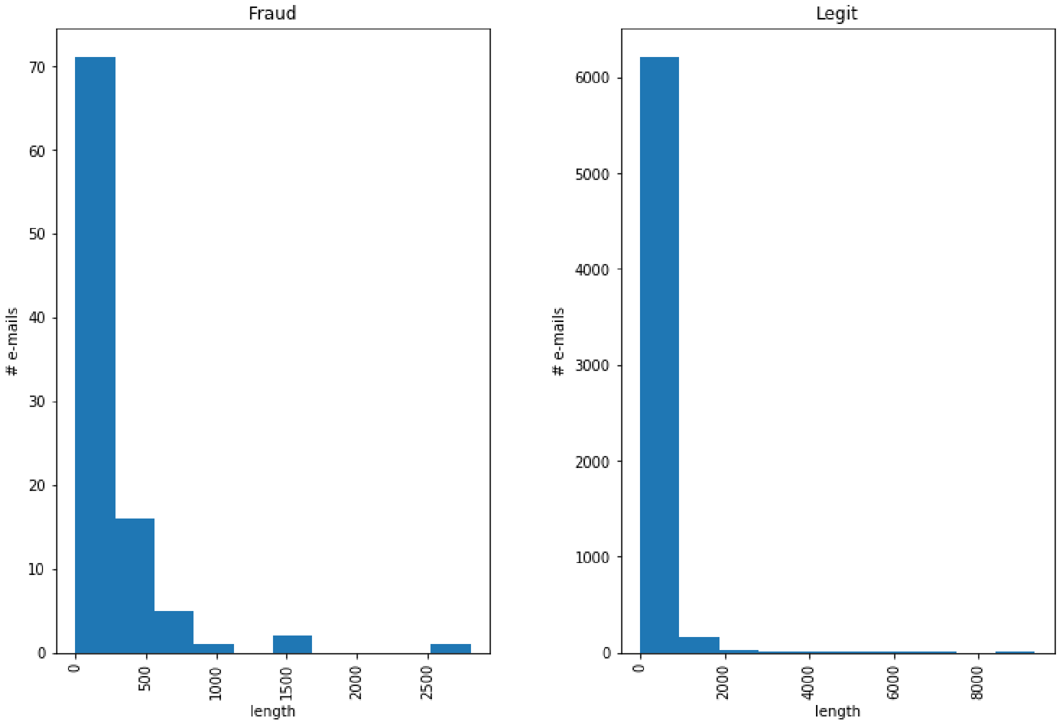


Figure 4.6: Histogram of length e-mails ENRON dataset, shown for both categories (in which length is defined as the number of words). As can be seen both categories are distributed similarly.

4.3. Fraud analysis confidential dataset

As in the previous two sections, this section will be usef for the analysis of the data of the confidential dataset. Since this dataset is a recent 'real life' dataset, it is expected that differences between relevant and not relevant e-mails will be seen.

4.3.1. Word counts and occurrences

In a similar manner as is described in Section 4.1.1 the e-mails in the labeled confidential dataset have also been analysed. In Table 4.8 the counts of the 10 words that are most identifiable for relevant e-mail are shown. Table 4.7 shows the counts of the 10 words that are most identifiable for not relevant e-mail. The reason that the original words are not shown is because of the confidentiality of the dataset. The values in the column in which the ratio counts per message is shown are calculated according to Equations (4.1) and (4.2). From both these tables it can be concluded that there are again various differences in word usage, however the difference is smaller than in the similar spam analysis. This smaller difference might lead to results that have a lower performance than in spam classification. Furthermore, it is remarkable to see that there is a word which is identified as a 'relevant' word, that also occurs multiple times in e-mails of the category not relevant. This shows how words that are used in fraud related e-mails are not only words that are specifically related to fraud, but also part of the language used in normal e-mails.

Word	Count relevant	Count not relevant	Ratio counts per message
word0	0	947	0.015
word1	0	477	0.030
word2	0	291	0.049
word3	0	290	0.049
word4	0	249	0.057
word5	0	203	0.070
word6	0	189	0.075
word7	0	187	0.076
word8	0	164	0.086
word9	0	143	0.099

Table 4.7: Top 10 words which are identifiable for not relevant e-mail messages from confidential dataset according to the ratio of words per message. The original words are not shown due to the confidentiality of the dataset. As can be noted these 10 words are only used in the not relevant e-mails, and therefore these words are assumed to be representative for this category. The ratio of counts per message shows how the words are sorted.

Word	Count relevant	Count not relevant	Ratio counts per message
word0	23	0	340.8
word1	12	0	184.6
word2	12	0	184.6
word3	23	1	170.4
word4	11	0	170.4
word5	89	7	159.8
word6	10	0	156.2
word7	10	0	156.2
word8	9	0	142.0
word9	9	0	142.0

Table 4.8: Top 10 words which are identifiable for relevant e-mail messages from confidential dataset according to the ratio of words per message. The original words are not shown due to the confidentiality of the dataset. As can be noted these 10 words are mainly used in the relevant e-mails, and therefore these words are assumed to be representative for this category. The most remarkable word is numbered 5, and occurs a number of times in both categories. The ratio of counts per message shows how the words are sorted.

4.3.2. Length of e-mail

As can be seen in Figure 4.7, and in Table 4.9, there is not much difference in terms of length between relevant and not relevant e-mails. Figure 4.7 shows the logarithm of the length of the e-mails in the two different categories, whilst in Table 4.9 the original values are presented. Furthermore, Table 4.9 shows us that especially the standard deviation of not relevant e-mails is much higher.

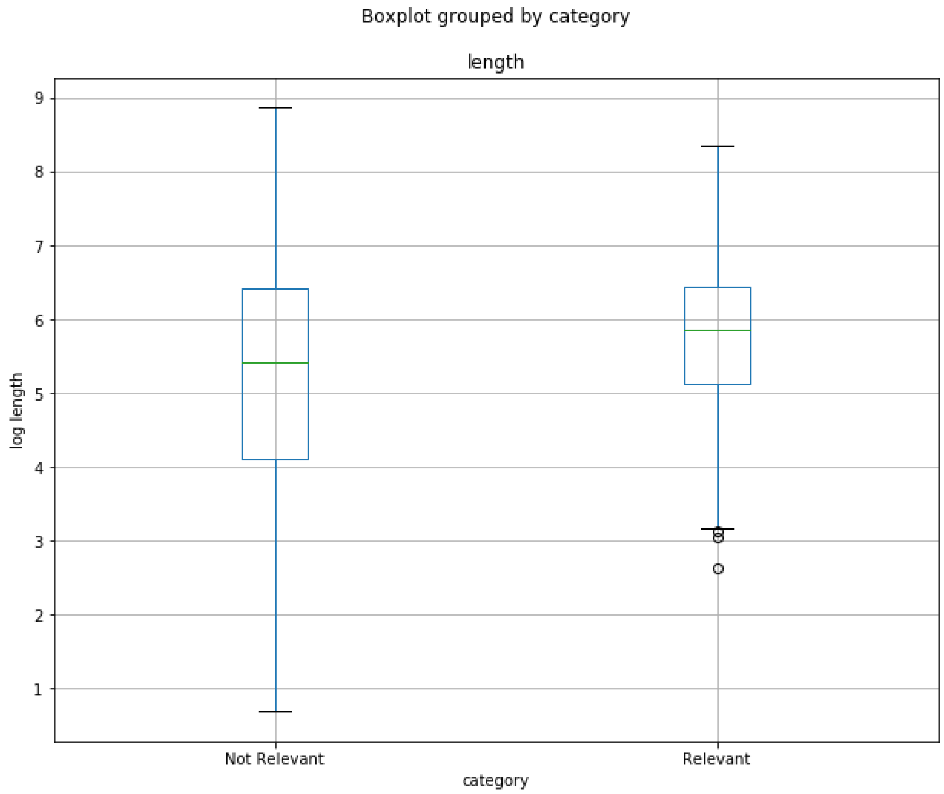


Figure 4.7: Boxplot of the logarithm of the length of e-mails of the two categories in the confidential dataset (in which the length is defined as the number of words).

	Relevant	Not Relevant
Mean	534.99	519.20
Standard deviation	608.14	788.48
Q1	168.0	66.0
median	350.0	232.0
Q3	631.0	622.0

Table 4.9: Basic statistical features of lengths of e-mails from confidential dataset (divided into the classification relevant and not relevant)

In order to get a better view on the distribution the lengths of e-mails, some histograms have been added. Figure 4.9 shows the histogram of all e-mails for both categories. Figure 4.8 shows the histogram of all e-mails regardless of their category. Based on these figures it would be suspected that the length of e-mails, in terms of number of words, follow a Pareto distribution.

Furthermore, the same conclusion as has been made based on the TREC and ENRON dataset is made. Only small differences in length are present between the categories.

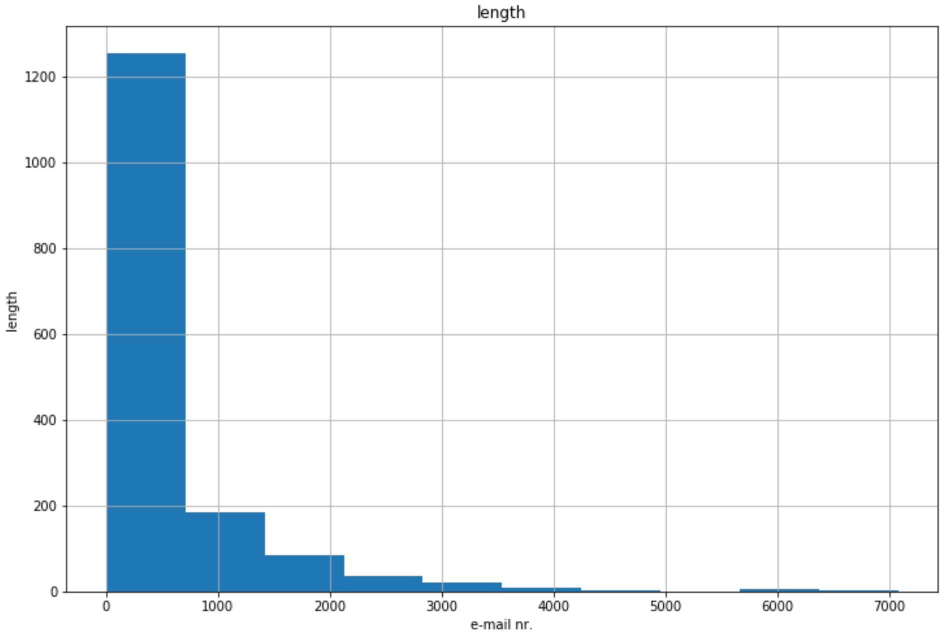


Figure 4.8: Histogram of length e-mails confidential dataset (in which length is defined as the number of words). Partially based on this histogram the lengths are expected to follow a Pareto distribution.

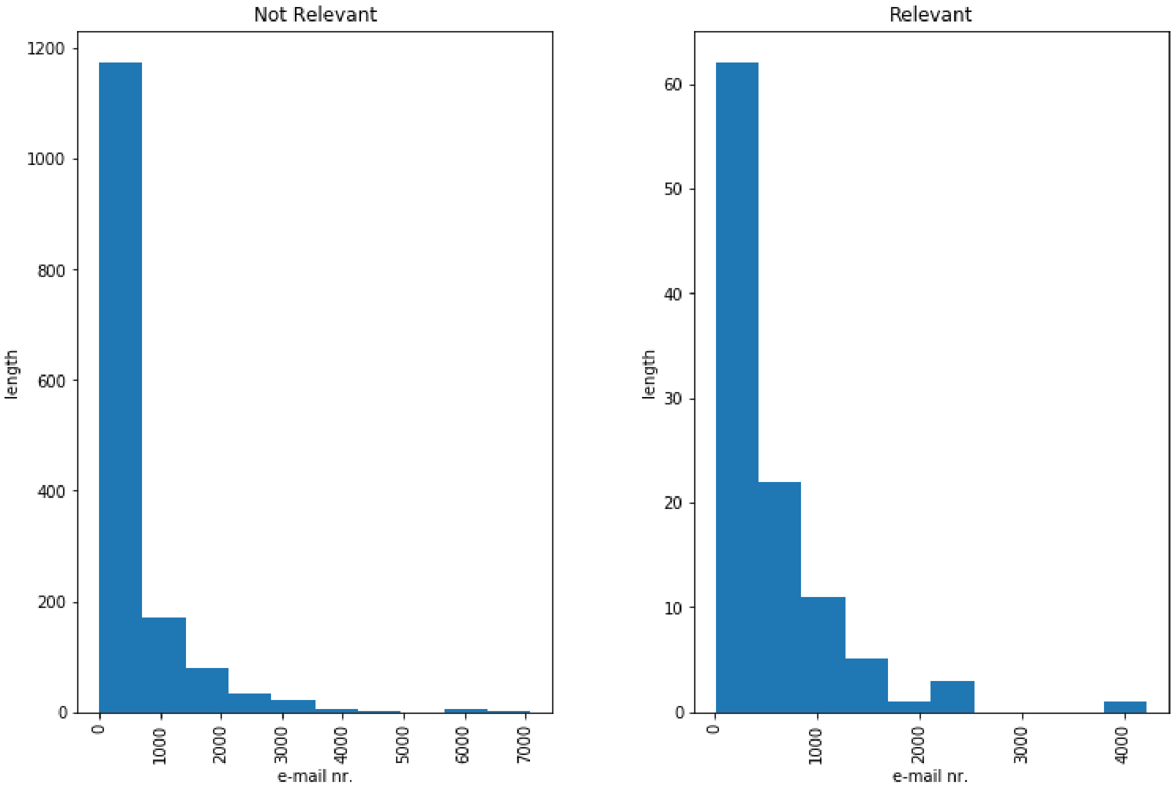


Figure 4.9: Histogram of length e-mails confidential dataset, shown for both categories (in which length is defined as the number of words). As can be seen both categories are distributed similarly. Partially based on this histogram the lengths are expected to follow a Pareto distribution.

4.4. Conclusion

Based on the analysis performed in this chapter, small differences have been found in the word usage between spam and ham e-mails as well as between relevant and unlabeled/not relevant e-mails. It should be remarked that the difference in word usage between relevant and unlabeled/not relevant e-mails is smaller, and it would therefore be expected that a model has lower performance compared to spam classification

The difference in length of e-mails in both categories (in terms of number of words) has also been looked at. This analysis concluded both for the spam/ham as well as the relevant/not labeled classification that the difference can especially be found in the outliers. The length of e-mails can therefore potentially be a good characteristic to identify certain e-mails.

This means that indeed the features taken from the research reports discussed in Chapter 1 are expected to be able to give a good classification. There might, however, be a difference in performance between spam classification and the classification of fraud related messages. Furthermore, the features on word frequencies and word occurrence are expected to perform better than the feature on e-mail lengths, since the differences between the categories for those two features is bigger than the difference for the feature of e-mail lengths.

5
Classification models

In the previous chapters the background information on the used mathematics (Chapter 2) and datasets (Chapter 3) have been discussed. In the previous chapter it has been confirmed that the proposed features, based on the literature study, show characteristics suitable for classification. Based on these observations, this chapter will focus on the mathematical models that will be applied to the data discussed in Chapter 3. This mathematical background will be given for the features: Word frequences, word occurrences and e-mail length. This chapter will, per feature, mainly discuss two different models: a generative and a discriminative model. For both models an estimation of the parameter based on MLE and Bayesian estimates is given. For the generative model this comes down to Classical Naive Bayes and Extended Naive Bayes. For the discriminative model these model names do not yet exists in literature. The mathematical explanations in this chapter are kept brief and only focused on the assumptions and way to calculate the probability that a new e-mail belongs to a certain category. More detailed mathematical derivations are given in Appendices B-D.

5.1. Word frequencies

In this section the generative and discriminative models based on the feature of word frequencies are given. The overview of the various models discussed is shown in Figure 5.1

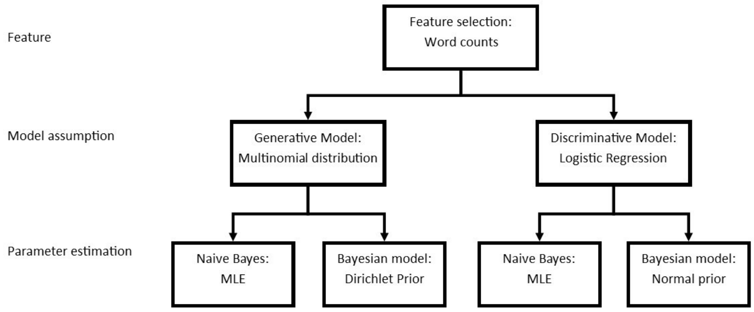

Feature
Model assumption
Parameter estimation
Feature selection: Word counts
Generative Model: Multinomial distribution
Discriminative Model: Logistic Regression
Naive Bayes: MLE
Bayesian model: Dirichlet Prior
Naive Bayes: MLE
Bayesian model: Normal prior

Figure 5.1: Overview of the model based on word frequencies discussed in this section. Based on the feature word counts to different models are discussed, each with two different parameter estimations.

5.1.1. Generative model

A simple generative model for e-mails will be described. Let $\left(t_1,\dots,t_J\right)$ be a given dictionary of words. Let $\boldsymbol{\theta}^y = \left(\theta_j^y\right)_{j=1}^J$ be relative frequencies for these words as they occur in e-mails, in which y is indicates whether the e-mail is relevant $(y = 1)$ or not relevant $(y = 0)$. Furthermore, let $\boldsymbol{z} = (z_1,\dots,z_n)$ be an ordered sequence of words.

The notation $x_j(\boldsymbol{z}) = \#\left\{i : z_i = t_j\right\}$ will be used, i.e. x_j is the word count of the j-th word in the dictionary. The posterior distribution over $\boldsymbol{x}$ is given by:

$$p(\boldsymbol{x} \mid y) = p(n)\binom{n}{x_1 \cdots x_J}\prod_{j=1}^{J}\left(\theta_j^y\right)^{x_j}. \tag{5.1}$$

Furthermore, let $\boldsymbol{z}^{(1)},\ldots,\boldsymbol{z}^{(N)}$ denote a sequence of N e-mails (or for the purpose of this model, equivalently: word counts $\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(N)}$). Moreover, $y^{(1)},\ldots,y^{(N)}$ denotes whether an e-mail is relevant or not relevant. All e-mails are assumed to be independently generated according to the model. Our available parameters are $\boldsymbol{\theta}^0$ and $\boldsymbol{\theta}^1$, and it is assumed that the probability distributions $p(y)$ (the distribution on whether an e-mail is relevant or not) and $p(n)$ (the distribution on the length of an e-mail) are fixed.

The assumption that e-mails are independently generated according to the model might not be valid when looking at certain threads of conversations, but is a generalisation of a mailbox. After all, not every e-mail sent and received will be about the same subject. Assuming the distributions $p(y)$ and $p(n)$ to be fixed is a simplification, the length n of an e-mail might depend on the category the e-mail belongs to and the classification y might be dependent on time. However, assuming fixed distributions will lead to a less computationally expensive model.

In the next two sections the parameters are assumed to be estimated by using Maximum Likelihood Estimation (Classical NB), or by applying a full Bayesian model (Extended NB). The assumption that the parameters can be estimated by Maximum Likelihood Estimation, can be valid since it is the computation that is least expensive and this makes the classification of new e-mails in general faster. The assumption that the parameters can be estimated by a full Bayesian model, can be valid since it takes (partially) into account the uncertainty about the parameter estimation. A full Bayesian model might be able to give better predictions.

Classification based on MLE of θ^y

The Maximum Likelihood Estimation of the generative model is the model which is referred to as Classical Naive Bayes. In this model the parameters are estimated with the following point estimates:

$$\theta_j^y = \frac{\sum\limits_{m=1,\ldots,N:y^{(m)}=y} x_j^{(m)}}{\sum\limits_{m=1,\ldots,N:y^{(m)}=y} n^{(m)}}. \tag{5.2}$$

For the classification of a new e-mail, Bayes' law is used and the obtained MLE estimators for $\boldsymbol{\theta}^y$ are plugged in:

$$p(y \mid \boldsymbol{z}) = \prod_{j=1}^{J}\left(\theta_j^y\right)^{x_j(\boldsymbol{z})} \cdot p^y(1-p)^{1-y}, \tag{5.3}$$

It is likely that with this formula a probability of 0 is given to both situations $y = 0$ and $y = 1$. This is the case since our dataset corresponding to the training of $\boldsymbol{\theta}$ does not have to have the situation in which every word occurs in an e-mail for both cases. Therefore Laplace smoothing will be applied to the MLE estimations of θ_j^y:

$$\tilde{\theta}_j^y = \frac{\sum\limits_{m=1,\ldots,N:y^{(m)}=y}\left(x_j^{(m)}\right)+1}{\sum\limits_{m=1,\ldots,N:y^{(m)}=y}\left(n^{(m)}\right)+J}. \tag{5.4}$$

Classification based on Bayesian estimation of θ^y

In the full Bayesian model prior on the parameters is used. In this case the conjugate prior of the Multinomial distribution is used, namely the Dirichlet distribution. This yields that

$$p\left(\boldsymbol{\theta}^\xi \mid \boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(N)},y^{(1)},\ldots,y^{(N)}\right) \sim Dir\left(\alpha^\xi\right), \tag{5.5}$$

with $\alpha_j^\xi = \sum\limits_{m=1,\ldots,N:y^{(m)}=\xi}\left(x_j^{(m)}\right)+\alpha_j$ and $\xi = 0,1$. The hyperparameter is chosen to be $\alpha_j = 1$ for all $1 \leq j \leq J$. This means that as little distinction as possible is made between the word occurrences beforehand, i.e. an uninformative prior is used.

Therefore, the conditional distribution of $y^{(N+1)}$ for our new e-mail is written as:

$$p\left(y^{(N+1)} \mid \boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(N)}, y^{(1)},\dots,y^{(N)}, \boldsymbol{x}^{(N+1)}\right) \propto p^{y^{(N+1)}}(1-p)^{1-y^{(N+1)}} \cdot \frac{1}{B\left(\boldsymbol{\alpha}^{y^{(N+1)}}\right)} \cdot B\left(\tilde{\boldsymbol{\alpha}}^{y^{(N+1)}}\right), \tag{5.6}$$

with

$$\tilde{\alpha}_j^{y^{(N+1)}} = x_j^{(N+1)} + \sum_{m=1,\dots,N: y^{(m)}=y^{(N+1)}} \left(x_j^{(m)}\right) + \alpha_j.$$

Which needs to be calculated for $y^{(N+1)} = 0$ and $y^{(N+1)} = 1$ since this expression holds up to the proportionality constant.

5.1.2. Discriminative model

A simple discriminative model for e-mails will be described. As in the generative model, let $(t_1,\dots,t_J)$ be a given dictionary of words. Let y indicate whether the e-mail is relevant $(y = 1)$ or not relevant $(y = 0)$.

Again let $\boldsymbol{z} = (z_1,\dots,z_n)$ be an ordered sequence of words. Also the notation $x_j(z) = \#\{i : z_i = t_j\}$ will be used, i.e. x_j is the word count of the j-th word in the dictionary.

Logistic Regression as discriminative model will be used, therefore assume that

$$p(y=1|\boldsymbol{x}) = \frac{1}{1+\exp\left(w_0 + \sum_{j=1}^{J} w_j x_j\right)}; \tag{5.7}$$

$$p(y=0|\boldsymbol{x}) = \frac{\exp\left(w_0 + \sum_{j=1}^{J} w_j x_j\right)}{1+\exp\left(w_0 + \sum_{j=1}^{J} w_j x_j\right)}. \tag{5.8}$$

Furthermore, let $\boldsymbol{z}^{(1)},\dots,\boldsymbol{z}^{(N)}$ denote a sequence of e-mails (or for the purpose of this model, equivalently: word counts $\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(N)}$). Moreover, let $y^{(1)},\dots,y^{(N)}$ denote whether an e-mail is relevant or not relevant. Assume all e-mails are independently generated according to the model. Let $\boldsymbol{w} = \left(w_1,\dots,w_J\right)$ be the weights needed in Formula (5.8) and (5.7). With the assumption that the outcomes follow a fixed Bernoulli distribution.

The assumption that e-mails are independently generated according to the model might not be valid when looking at certain threats of conversations, but is a generalisation of a mailbox. After all, not every e-mail sent and received will be about the same subject. Assuming the distributions $p(y)$ to be a fixed Bernoulli distribution is a simplification, the classification y might be dependent on time. However, the assumption will lead to a less computationally expensive model.

In the next two sections the parameters are assumed to be either estimated by using Maximum Likelihood Estimation, or by applying a full Bayesian model. The assumption that the parameters can be estimated by Maximum Likelihood Estimation, can be valid since it is the computation that is least expensive. The assumption that the parameters can be estimated by a full Bayesian model, can be valid since it takes (partially) into account the uncertainty about the parameter estimation.

Classification based on MLE of for w

In the Maximum Likelihood Estimation of the discriminative model will will use the following log likelihood:

$$l(\boldsymbol{w}) = \sum_{m=1}^{N} \left(y^{(m)} \ln\left(p\left(y=1 \mid \boldsymbol{x}^{(m)}\right)\right) + \left(1-y^{(m)}\right) \ln\left(p\left(y=0 \mid \boldsymbol{x}^{(m)}\right)\right)\right) \tag{5.9}$$

The only steps left to get the best parameters $\boldsymbol{w}$ is to maximize Equation (5.9) with respect to this parameter (in other words setting the derivative w.r.t. $\boldsymbol{w}$ to zero and solving this equation). Note that no closed form of the maximization of the log likelihood is available. Various algorithms are available to approximate the maximization, SAGA has been used, which was made available by the function LogisticRegression in the package sklearn.linear_model.

For the classification of a new e-mail, Equations (5.8) and (5.7) have been used. The classification property of logistic regression comes down to giving e-mail $\boldsymbol{z}^{(N+1)}$ (and corresponding data $\boldsymbol{x}^{(N+1)}$) label $y^{(N+1)} = 0$ if $0 < w_0 + \sum_{j=1}^{J} w_j x_j^{(N+1)}$, and label $y = 1$ otherwise.

Classification based on Bayesian estimation of w

In the full Bayesian model a prior on the parameters has been used. In this case a weakly informative normally distributed prior is used. This gives us the following posterior distribution for $\boldsymbol{w}$ (a detailed description is given in Appendix B.2):

$$p\left(\boldsymbol{w} \mid \boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(N)}, y^{(1)}, \dots, y^{(N)}\right) \propto$$

$$\prod_{m=1}^{N} \left(\frac{1}{1 + \exp\left(w_0 + \sum_{j=1}^{J} w_j x_j^{(m)}\right)} \cdot y^{(m)} + \frac{\exp\left(w_0 + \sum_{j=1}^{J} w_j x_j^{(m)}\right)}{1 + \exp\left(w_0 + \sum_{j=1}^{J} w_j x_j^{(m)}\right)} \cdot \left(1 - y^{(m)}\right) \right) \cdot \prod_{i=0}^{J} \frac{1}{\sqrt{2\pi \cdot 10^{12}}} \exp\left(-\frac{w_i^2}{2 \cdot 10^{12}}\right).$$

Therefore, the conditional distribution of $y^{(N+1)}$ for our new e-mail can be written as (S^J is the J dimensional simplex):

$$p\left(y^{(N+1)} \mid \boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(N+1)}, y^{(1)}, \dots, y^{(N)}\right) \propto$$

$$\int_{S^J} \prod_{m=1}^{N+1} \left(\frac{y^{(m)}}{1 + \exp\left(w_0 + \sum_{j=1}^{J} w_j x_j^{(m)}\right)} + \frac{\exp\left(w_0 + \sum_{j=1}^{J} w_j x_j^{(m)}\right)}{1 + \exp\left(w_0 + \sum_{j=1}^{J} w_j x_j^{(m)}\right)} \cdot \left(1 - y^{(m)}\right) \right) \cdot \prod_{i=0}^{J} \frac{1}{\sqrt{2\pi \cdot 10^{12}}} \exp\left(-\frac{w_i^2}{2 \cdot 10^{12}}\right) d\boldsymbol{w}.$$

Which needs to calculated for $y^{(N+1)} = 0$ and $y^{(N+1)} = 1$ since this expression holds up to the proportionality constant.

5.2. Word occurrences

In this section the generative and discriminative models based on the feature of word occurrences are discussed. The overview of the various models discussed is shown in Figure 5.2

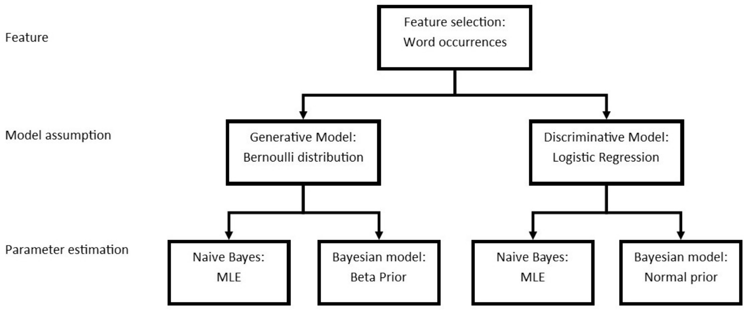

Feature
Feature selection: Word occurrences
Model assumption
Generative Model: Bernoulli distribution
Discriminative Model: Logistic Regression
Parameter estimation
Naive Bayes: MLE
Bayesian model: Beta Prior
Naive Bayes: MLE
Bayesian model: Normal prior

Figure 5.2: Overview of the model based on word occurrences discussed in this section. Based on the feature word counts to different models are discussed, each with two different parameter estimations.

5.2.1. Generative model

A simple generative model for e-mails will be described. Let $\boldsymbol{z} = (z_1, \ldots, z_n)$ be a sequence of words of an e-mail. Let $\left(t_1, \ldots, t_J\right)$ be a given dictionary of words. Let $\boldsymbol{q}^y = \left(q_j^y\right)_{j=1}^J$, in which q_j^y represents the probability that word t_j corresponds to the category relevant $(y = 1)$ or not relevant $(y = 0)$, i.e. $p\left(t_i \mid y\right) = q_j^y$. Furthermore, it holds that $q_j^1 = 1 - q_j^0$.

Let $\boldsymbol{x} = \left(x_1, \ldots, x_J\right)$ be the corresponding features of the given dictionary, in which $x_i = 1$ if word t_i is present in the e-mail (i.e. $t_i \in \boldsymbol{z}$) and $x_i = 0$ if word t_i is not present (i.e. $t_i \not\in \boldsymbol{z}$). For simplicity, assume that the distribution of n does not depend on y. The posterior distribution over $\boldsymbol{x}$ will then become:

$$p(\boldsymbol{x} \mid y) = p(n) \prod_{j=1}^{J} \left(q_j^y\right)^{x_j} \cdot \left(1 - q_j^y\right)^{1-x_j}. \tag{5.10}$$

Furthermore, let $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}$ denote word occurrences of e-mails. Moreover, let $y^{(1)}, \ldots, y^{(N)}$ denote whether an e-mail is relevant or not. All e-mails are assumed to be independently generated according to the model. Our available parameters are $\boldsymbol{q}^0$ and $\boldsymbol{q}^1$, and the probability distributions $p(y)$ and $p(n)$ are assumed to be fixed.

These assumptions are based on the same reasoning as is given in Chapter 5.1. For completeness this reasoning is also stated here.

The assumption that e-mails are independently generated according to the model might not be valid when looking at certain threats of conversations, but is a generalisation of a mailbox. After all, not every e-mail sent and received will be about the same subject. Assuming the distributions $p(y)$ and $p(n)$ to be fixed is a simplification, the length n of an e-mail might depend on whether an e-mail is relevant or not relevant and the classification y might be dependent on time. However, assuming fixed distributions will lead to a less computationally expensive model.

In the next two sections the parameters are assumed to be either estimated by using Maximum Likelihood Estimation, or by applying a full Bayesian model. The assumption that the parameters can be estimated by Maximum Likelihood Estimation, can be valid since it is the computation that is least expensive. The assumption that the parameters can be estimated by a full Bayesian model, can be valid since it takes (partially) into account the uncertainty about the parameter estimation.

Classification based on MLE of q^y

The Maximum Likelihood Estimation of the generative model gives us the following point estimates:

$$q_j^y = \frac{\sum\limits_{m=1,\ldots,N:y^{(m)}=y} x_j^{(m)}}{\sum\limits_{m=1,\ldots,N:y^{(m)}=y} 1}.$$

For the classification of a new e-mail, Bayes' law is used and the obtained MLE estimators for $\boldsymbol{\theta}^y$ are plugged in:

$$p(y \mid \boldsymbol{x}) \propto \prod_{j=1}^{J} \left(\left(q_j^y\right)^{x_j} \cdot \left(1 - q_j^y\right)^{1-x_j} \right) \cdot p^y (1-p)^{1-y}, \tag{5.11}$$

It is likely that with this formula probability of 0 is given in both situations $y = 0$ and $y = 1$. This is the case since our dataset corresponding to the training of $\boldsymbol{q}$ does not have to have the situation in which every word occurs in an e-mail for both cases. Therefore Laplace smoothing will be applied to the MLE estimations of q_j^y:

$$\tilde{q}_j^y = \frac{\sum\limits_{m=1,\ldots,N:y^{(m)}=y} \left(x_j^{(m)}\right) + 1}{\sum\limits_{m=1,\ldots,N:y^{(m)}=y} (1) + J}. \tag{5.12}$$

Classification based on Bayesian estimation of q^y

In the full Bayesian model a prior on the parameters has been used. In this case the conjugate prior of the Bernoulli distribution is used, namely the Beta distribution. This yields that

$$p\left(\boldsymbol{q}^\xi \mid \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}, y^{(1)}, \ldots, y^{(N)}\right) \propto \prod_{j=1}^{J} \frac{\left(q_j^\xi\right)^{\alpha_j - 1 + \sum\limits_{m=1,\ldots,N:y^{(m)}=\xi} x_j^{(m)}} \cdot \left(1 - q_j^\xi\right)^{\beta_j - \sum\limits_{m=1,\ldots,N:y^{(m)}=\xi} x_j^{(m)}}}{B\left(\alpha_j, \beta_j\right)}, \tag{5.13}$$

with $\xi = 0, 1$.

Therefore, the conditional distribution of $y^{(N+1)}$ for our new e-mail can be written as:

$$p\left(y^{(N+1)} \mid \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}, y^{(1)}, \ldots, y^{(N)}, \boldsymbol{x}^{(N+1)}\right) \quad \propto \quad \prod_{j=1}^{J} \frac{p^{y^{(N+1)}} (1-p)^{1-y^{(N+1)}}}{B\left(\alpha_j, \beta_j\right)} \cdot B\left(\tilde{\alpha}_j, \tilde{\beta}_j\right)$$

in which $\tilde{\alpha}_j = \alpha_j + \sum\limits_{m=1,\ldots,N+1:y^{(m)}=y^{(N+1)}} x_j^{(m)}$ and $\tilde{\beta}_j = \beta_j + \sum\limits_{m=1,\ldots,N+1:y^{(m)}=y^{(N+1)}} \left(1 - x_j^{(m)}\right)$.
Which need to be calculated for $y^{(N+1)} = 0$ and $y^{(N+1)} = 1$ since this expression holds up to the proportionality constant.

5.2.2. Discriminative model

The discriminative model for e-mails based on word occurrences is similar as the discriminative model for e-mails based on word frequencies, which is described in Section 5.1.2 and B.2. Let $\left(t_1, \ldots, t_J\right)$ be a given dictionary of words. Let y indicate whether the e-mail is relevant $(y = 1)$ or not relevant $(y = 0)$. Furthermore let $\boldsymbol{x} = \left(x_1, \ldots, x_J\right)$ be a sequence that indicates whether word t_i is in e-mail message $\boldsymbol{z}$ then $x_i = 1$ or else $x_i = 0$.

The model of Logistic Regression is the same as is described by Equations (5.7) and (5.8), but in this case with the variable $\boldsymbol{x}$ as described above.

The MLE as well as the Bayesian estimate for $\boldsymbol{w}$ used in Equations (5.7) and (5.8) is approximated with the same methods as described in Section B.2. The way new data is given a classification is also described in this section, please note that the feature vector $\boldsymbol{x}$ is used as described above, instead of the word frequencies used in Section B.2.

5.3. E-mail length

In this section the generative and discriminative models based on the feature of e-mail length are discussed. Please recall that the length of an e-mail is defined as the number of words that are present in the e-mail after preparation. The overview of the various models discussed is shown in Figure 5.3

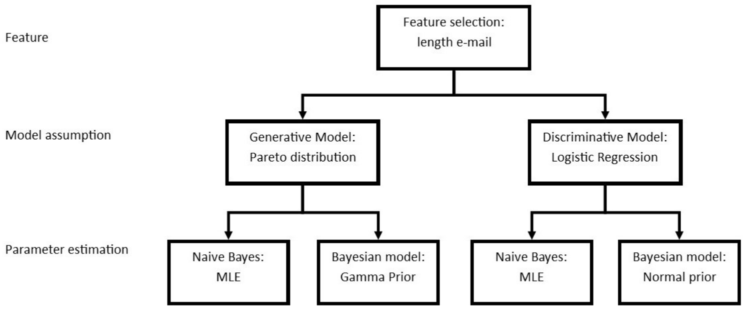

Feature
Feature selection: length e-mail
Model assumption
Generative Model: Pareto distribution
Discriminative Model: Logistic Regression
Parameter estimation
Naive Bayes: MLE
Bayesian model: Gamma Prior
Naive Bayes: MLE
Bayesian model: Normal prior

Figure 5.3: Overview of the model based on length of an e-mail discussed in this section. Based on the feature word counts to different models are discussed, each with two different parameter estimations.

5.3.1. Generative model

A simple generative model for e-mails will be described. Let $\boldsymbol{z} = (z_1, \ldots, z_n)$ be a sequence of words of an e-mail. Let $x = n$ denote the length of an e-mail. The length of an e-mail is defined by the number of words present in the e-mail. As is noted in Chapter 4, the lengths of an e-mail are Pareto distributed (observed based on the data as well as by other literature [44]). The distribution over x will then for both categories of y become:

$$p(x \mid y) = \frac{\alpha^y \cdot \eta^{\alpha^y}}{x^{\alpha^y + 1}}. \tag{5.14}$$

Furthermore, let $x^{(1)}, \ldots, x^{(N)}$ denote the lengths of N e-mails. Moreover, let $y^{(1)}, \ldots, y^{(N)}$ denote whether an e-mail is relevant or not. All e-mails are assumed to be independently generated according to the model. Our available parameters are η, α^0 and α^1, and the probability distribution $p(y)$ is assumed to be fixed.

The assumptions are based on the same reasoning as is given in Chapter 5.1. For completeness this reasoning is also stated here.

The assumption that e-mails are independently generated according to the model might not be valid when looking at certain threats of conversations, but is a generalisation of a mailbox. After all, not every e-mail sent and received will be about the same subject. Assuming the distribution $p(y)$ to be fixed is a simplification, the classification y might be dependent on time. However, assuming a fixed distribution will lead to a less computationally expensive model.

In the next two sections the parameters will be assumed to be either estimated by using Maximum Likelihood Estimation, or by applying a full Bayesian model. The assumption that the parameters can be estimated by Maximum Likelihood Estimation, can be valid since it is the computation that is least expensive. The assumption that the parameters can be estimated by a full Bayesian model, can be valid since it takes (partially) into account the uncertainty about the parameter estimation.

Classification based on MLE of α^y

The Maximum Likelihood Estimation of the generative model gives us the following point estimates:

$$\hat{\alpha}^y = \frac{\sum\limits_{m=1,\ldots,N: y^{(m)}=y} 1}{\sum\limits_{m=1,\ldots,N: y^{(m)}=y} \ln\left(\frac{x^{(m)}}{\hat{\eta}}\right)},$$

in which $\hat{\eta} = \min\limits_{m=1,\ldots,N}(x^{(m)})$.
For the classification of a new e-mail, Bayes' law is used and the obtained MLE estimators for α^0, α^1 and η are plugged in:

$$p(y \mid \boldsymbol{x}) \propto \frac{\hat{\alpha}^y \cdot \hat{\eta}^{\hat{\alpha}^y}}{x^{\hat{\alpha}^y+1}} \cdot p^y(1-p)^{1-y}. \tag{5.15}$$

Classification based on Bayesian estimation of α^y

In the full Bayesian model a prior on the parameters will be used. In this case the conjugate prior of the Pareto distribution, namely the Gamma distribution, is used. This yields that

$$p\left(\alpha^\xi \mid \boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(N)}, y^{(1)},\ldots,y^{(N)}\right) \sim Gamma(\hat{a},\hat{b}), \tag{5.16}$$

with $\xi = 0,1$. Furthermore, $\hat{a}^\xi = a + \sum\limits_{m=1,\ldots,N: y^{(m)}=\xi} 1$ and $\hat{b}^\xi = \sum\limits_{m=1,\ldots,N: y^{(m)}=\xi} \left(\ln(\eta) - \ln(x^{(m)}) - b\right)$ in which a, b hyperparameters of the prior Gamma distribution (details can be found in Appendix D).

Therefore, the conditional distribution of $y^{(N+1)}$ for our new e-mail can be written as:

$$p\left(y^{(N+1)} \mid \boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(N)}, y^{(1)},\ldots,y^{(N)}, \boldsymbol{x}^{(N+1)}\right) \quad \propto \quad p^{y^{(N+1)}}(1-p)^{y^{(N+1)}} \cdot \frac{\Gamma(\hat{\alpha}^{y^{(N+1)}}+1)}{\left(\hat{b}^{y^{(N+1)}} - \ln\left(\frac{\eta}{x^{(N+1)}}\right)\right)^{\hat{\alpha}^{y^{(N+1)}}+1}},$$

in which η is assumed to be a fixed valued hyperparameter, of which the value equals 1.
This equation needs to be calculated for $y^{(N+1)} = 0$ and $y^{(N+1)} = 1$ since this expression holds up to the proportionality constant.

5.3.2. Discriminative model

A simple discriminative model for e-mails will be described. As in the generative model, let y indicate whether the e-mail is relevant ($y = 1$) or not relevant ($y = 0$).

Again let $\boldsymbol{z} = (z_1,\ldots,z_n)$ be a sequence of words of an e-mail. Let $x = n$ denote the length of an e-mail. The length of an e-mail will be defined by the number of words present in the e-mail.
Logistic Regression will be used as discriminative model. Therefore it will be assumed that

$$p(y=1|x) = \frac{1}{1+\exp(w_0 + w_1 x)}; \tag{5.17}$$

$$p(y=0|x) = \frac{\exp(w_0 + w_1 x)}{1+\exp(w_0 + w_1 x)}. \tag{5.18}$$

Furthermore, let $\boldsymbol{z}^{(1)},\ldots,\boldsymbol{z}^{(N)}$ denote a sequence of e-mails (or for the purpose of this model, equivalently: the length of the e-mails $x^{(1)},\ldots,x^{(N)}$). Moreover, let $y^{(1)},\ldots,y^{(N)}$ denote whether an e-mail is relevant or not relevant. All e-mails are assumed to be independently generated according to the model. Let $\boldsymbol{w} = (w_0, w_1)$ be the weights needed in Formula (5.18) and (5.17). With the assumption that the outcomes follow a fixed Bernoulli distribution.

The assumption that e-mails are independently generated according to the model might not be valid when looking at certain threats of conversations, but is a generalisation of a mailbox. After all, not every e-mail sent and received will be about the same subject. Assuming the distribution $p(y)$ to be fixed is a simplification, the classification y might be dependent on time. However, assuming a fixed distribution will lead to a less computationally expensive model.

In the next two sections the parameters are either assumed to be estimated by using Maximum Likelihood Estimation, or by applying a full Bayesian model. The assumption that the parameters can be estimated by Maximum Likelihood Estimation, can be valid since it is the computation that is least expensive. The assumption that the parameters can be estimated by a full Bayesian model, can be valid since it takes (partially) into account the uncertainty about the parameter estimation.

Classification based on MLE of w

In the Maximum Likelihood Estimation of the discriminative model will will use the following log likelihood:

$$l(\boldsymbol{w}) = \sum_{m=1}^{N} \left(y^{(m)} \ln\left(p\left(y=1 \mid \boldsymbol{x}^{(m)}\right)\right) + \left(1-y^{(m)}\right) \ln\left(p\left(y=0 \mid \boldsymbol{x}^{(m)}\right)\right)\right). \tag{5.19}$$

The only steps left is to get the best parameters $\boldsymbol{w}$ is to maximize Equation (5.19) with respect to this parameter (in other words setting the derivative w.r.t. $\boldsymbol{w}$ to zero and solving this equation). Note that no closed form of the maximization of the log likelihood is available. Various algorithms are available to approximate the maximization, SAGA has been used which was made available by the function LogisticRegression in the package sklearn.linear_model.

For the classification of a new e-mail, Equations (5.18) and (5.17) will be used. The classification property of logistic regression comes down to giving e-mail $\boldsymbol{z}^{(N+1)}$ (and corresponding length $x^{(N+1)}$) label $y^{(N+1)} = 0$ if $0 < w_0 + w_1 x^{(N+1)}$, and label $y = 1$ otherwise.

Classification based on Bayesian estimation of w

In the full Bayesian model a prior on the parameters will be used. In this case a weakly informative normally distributed prior has been used. This gives us the following posterior distribution for $\boldsymbol{w}$:

$$p\left(\boldsymbol{w} \mid x^{(1)}, \dots, x^{(N)}, y^{(1)}, \dots, y^{(N)}\right)$$

$$\propto \prod_{m=1}^{N} \left(\frac{1}{1+\exp\left(w_0 + w_1 x^{(m)}\right)} \cdot y^{(m)} + \frac{\exp\left(w_0 + w_1 x^{(m)}\right)}{1+\exp\left(w_0 + w_1 x^{(m)}\right)} \cdot \left(1 - y^{(m)}\right) \right) \cdot \prod_{i=0}^{1} \frac{1}{\sqrt{2\pi \cdot 10^{12}}} \exp\left(-\frac{w_i^2}{2 \cdot 10^{12}}\right).$$

Therefore, the conditional distribution of $y^{(N+1)}$ for our new e-mail can be written as (S^2 is the 2 dimensional simplex):

$$p\left(y^{(N+1)} \mid x^{(1)}, \dots, x^{(N+1)}, y^{(1)}, \dots, y^{(N)}\right)$$

$$\begin{aligned} \propto \int_{S^2} & \left(\frac{y^{(N+1)}}{1+\exp\left(w_0 + w_1 x^{(N+1)}\right)} + \frac{\exp\left(w_0 + w_1 x^{(N+1)}\right)}{1+\exp\left(w_0 + w_1 x^{(N+1)}\right)} \cdot \left(1 - y^{(N+1)}\right) \right) \\ & \cdot \prod_{m=1}^{N} \left(\frac{y^{(m)}}{1+\exp\left(w_0 + w_1 x^{(m)}\right)} + \frac{\exp\left(w_0 + w_1 x^{(m)}\right)}{1+\exp\left(w_0 + w_1 x^{(m)}\right)} \cdot \left(1 - y^{(m)}\right) \right) \cdot \prod_{i=0}^{1} \frac{1}{\sqrt{2\pi \cdot 10^{12}}} \exp\left(-\frac{w_i^2}{2 \cdot 10^{12}}\right) d\boldsymbol{w}. \end{aligned}$$

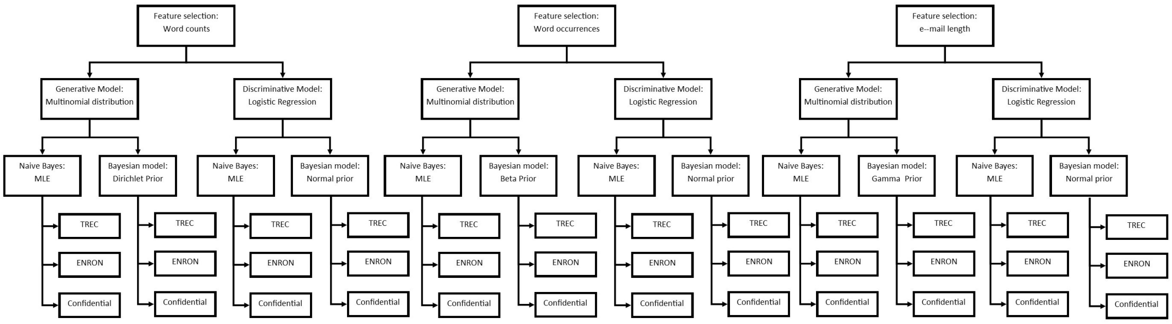
Which need to be calculated for $y^{(N+1)} = 0$ and $y^{(N+1)} = 1$ since this expression holds up to the proportionality constant.

6
Results

The models described in the previous chapter have been applied to the datasets described in Chapter 3. This chapter will analyse the results of each of these models. Per model, at first, in order to check for possible problems in the models, a parameter analysis will be done. After that the classification statistics will be explained and a conclusion is drawn. The analysis has mainly be focused around the results based on the confidential dataset. The TREC dataset has been used as check with other research reports as well as a reference point for comparison of the classification performance of the ENRON dataset and the confidential dataset of benchmark performances were not available. The ENRON dataset will be mentioned in the analysis, however due to the reasons mentioned in Section 3.2.2 (e.g. data quality) the dataset is most likely not a representative dataset. As has been explained, the ENRON dataset has been included as it is the only publicly available dataset that contains fraudulent e-mails, it is therefore useful for future research to be able to compare results.

Please note that the models explained in the previous chapter have been based on three features and each feature has 4 different parameter estimations. Furthermore, there are three different datasets. This gives in total 36 different results, which is also visualised in Figure 6.1. 5-fold cross validation has been applied to the ENRON and confidential dataset, as has been explained in Chapter 3. Due to computational limitations the performance of the TREC dataset is only measured on set 1. Overall this gave a lot of results, this chapter will therefore only elaborate on results that are worth mentioning, additional results can be found in the Appendix. Furthermore, due to computational limitations the discriminative models have only been applied with 1000 parameters. The 1000 parameters have been based on the top 500 words identifiable with relevant e-mail messages and the top 500 words identifiable as not relevant e-mail messages. The way these words are chosen is explained in Chapter 4. In order to be able to compare models the generative models have been applied on all available words as well as the top 1000 words used by the discriminative models.

Furthermore, before an analysis of the results and parameters is done, the classification rules will be explained in Section 6.1.



Feature selection: Word counts
Generative Model: Multinomial distribution
Discriminative Model: Logistic Regression
Naive Bayes: MLE
Bayesian model: Dirichlet Prior
Naive Bayes: MLE
Bayesian model: Normal prior
Feature selection: Word occurrences
Generative Model: Multinomial distribution
Discriminative Model: Logistic Regression
Naive Bayes: MLE
Bayesian model: Beta Prior
Naive Bayes: MLE
Bayesian model: Normal prior
Feature selection: e--mail length
Generative Model: Multinomial distribution
Discriminative Model: Logistic Regression
Naive Bayes: MLE
Bayesian model: Gamma Prior
Naive Bayes: MLE
Bayesian model: Normal prior
TREC
ENRON
Confidential

Figure 6.1: Overview of all available models, parameter estimations and datasets. In total this gives 36 different classification results that will be presented in this Chapter or the Appendix

6.1. Classification rules

The models explained in Chapter 5 are meant to perform a classification. In Chapter 3 it has already been explained which categories are available per dataset (TREC: spam and ham, ENRON: Fraud and unlabeled and in the confidential dataset: relevant vs and relevant). In this section will be explained what the classification rules for each dataset, meaning at what probabilities an e-mail will be classified to a certain category. The classification rules are similar for all three datasets, and summarised in Figure 6.2.

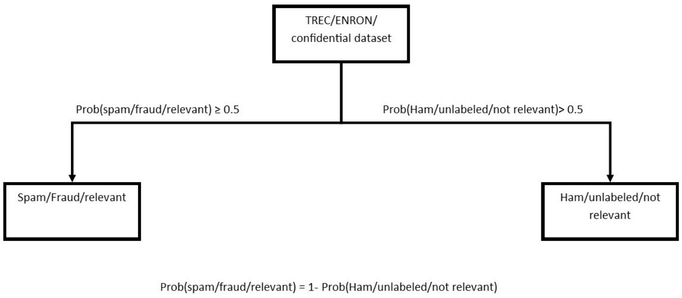

TREC/ENRON/
confidential dataset
Prob(spam/fraud/relevant) ≥ 0.5
Prob(Ham/unlabeled/not relevant)> 0.5
Spam/Fraud/relevant
Ham/unlabeled/not relevant
Prob(spam/fraud/relevant) = 1- Prob(Ham/unlabeled/not relevant)

Figure 6.2: Classification rules used to classify e-mails.

6.1.1. TREC

The spam e-mails in the TREC dataset are classified with label $y = 1$ and ham (not spam) e-mail with label $y = 0$. The probability of a particular category needs to be above 0.5 in order for the e-mail to get the classification of this category. If the probabilities are equal (i.e. both 0.5) the classification 'spam' is given. This classification for equal probabilities is chosen with the notion of having to review the category manually. Furthermore, if the probabilities are equal this can mean that either the e-mail is empty (meaning no words are present) or that the e-mail does not contain any known words (meaning that the words in the new e-mail are not in the training set). In both cases this means that it is likely a spam e-mail and that a closer look needs to be taken.

6.1.2. ENRON

In case of the labeled ENRON dataset there are two categories, namely relevant and unlabeled. Relevant e-mail is given a classification with label $y = 1$ and unlabeled e-mail with label $y = 0$. The probability of a particular category needs to be above 0.5 in order for the e-mail to get the classification of this category. If the probabilities are equal (i.e. both 0.5) the classification relevant will be given. This classification for equal probabilities is chosen with the notion of having to review the category relevant manually. In this way if the model is in doubt about the classification (i.e. probabilities equal to 0.5) no e-mails are removed without being sure of it being not relevant.

6.1.3. Confidential dataset

In case of the confidential KPMG dataset there are two categories, namely relevant and not relevant. Relevant e-mail is given a classification with label $y = 1$ and not relevant e-mail with label $y = 0$. The probability of a particular category needs to be above 0.5 in order for the e-mail to get the classification of this category. If the probabilities are equal (i.e. both 0.5) the classification relevant will be given. This classification for equal probabilities is chosen with the notion of having to review the category relevant manually. In this way if the model is in doubt about the classification (i.e. probabilities equal to 0.5) no e-mails are removed without being sure of it being not relevant.

6.2. Word frequencies

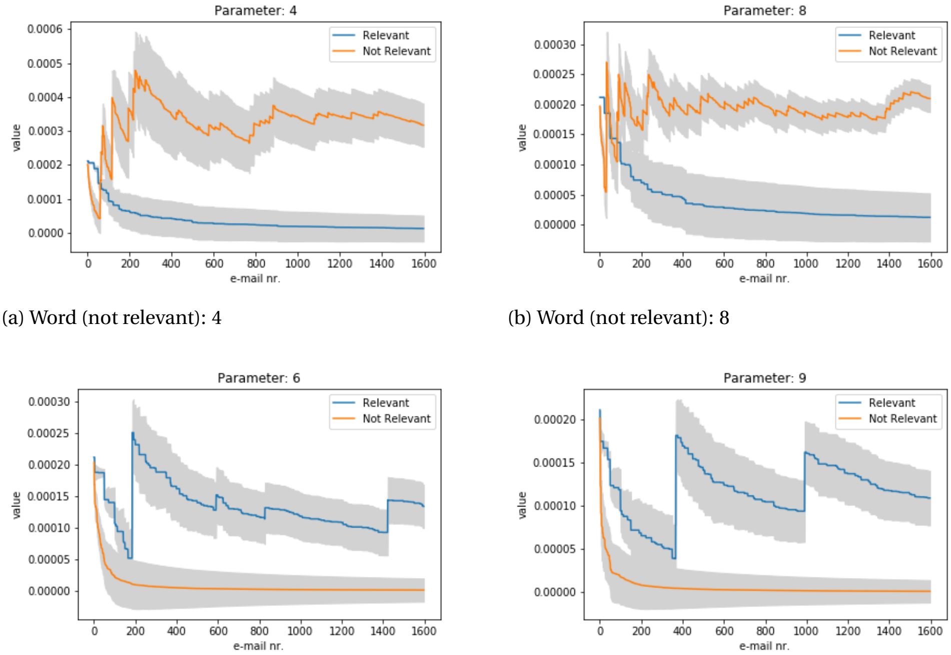
As is stated in the introduction of this chapter, the results corresponding to the models described in the previous chapter will be explained in the following sections. Moreover, in Attachment F.1 more results are presented. Especially results corresponding to the TREC and ENRON dataset, as well as additional results of the 5-fold cross validation.

6.2.1. Parameter analysis

In this section an analysis on the behaviour of the parameters in the various models is given. By doing this analysis any possible problems of the model can be identified before even looking at the results.

Generative model - MLE

For various words it can be checked how the corresponding parameter is trained. Figure 6.3 show this training process for two words that are identifiable with the two categories (four different words in total, in Section 4.2.1 it has been explained how the words are identifiable with each category). In the Appendix the same kind of figures can also be found for several other words. In the figures two lines can be seen, one line is for the training process of the parameter related to the category relevant, the other to the parameter related to the category not relevant. Furthermore, the grey area around these lines represent one standard deviation. From these figures it can be concluded that especially the parameters of the words related to the category relevant show spiky behaviour (e.g. the words numbered 6 and 9 in the category relevant), which might be an indication that the model is not trained well enough on this part. After all, two gradually stabilizing lines would be best. This behaviour is present with words that are identifiable as not relevant e-mails (words 4 and 8 in the category not relevant), but generally speaking not with the category relevant. During the analysis of the results it should be kept in mind that this spiky behaviour is present, the results might increase in performance if more information is available about these words (meaning if more e-mails in the category relevant are used).



(a) Word (not relevant): 4

(b) Word (not relevant): 8

(c) Word (relevant): 6

(d) Word (relevant): 9

Figure 6.3: Training process generative model with MLE (confidential dataset). The two lines indicate how the value of the parameter corresponding to the words are trained. In these figures 'Legit' is the category of the unlabeled e-mails.

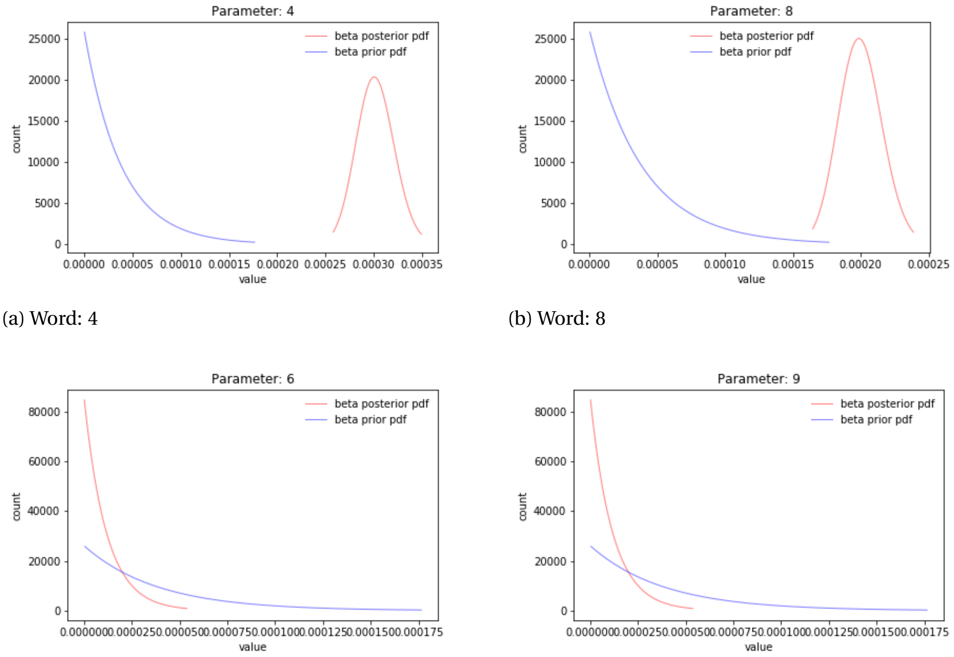
When looking at the differences of the training process between the two fraud related datasets no major differences between the ENRON dataset (Figures F.1 and F.2) and the confidential dataset (Figures F.6 and F.7) are seen. They both show stabalizing lines for parameters related to the category not relevant, whilst the parameters related to the category relevant show in both cases spiky behaviour. This spiky behaviour can be a cause for a potential low recall of the category relevant, which needs to be taken into account when analysing the results.

It should be remarked that these figures are unique for the way the model is trained, if the e-mails are shuffled the figures will be slightly different.

Furthermore, as can be seen in Figures F.1 and F.2 there are various non-words present (e.g. developmentenron), these words can occur due to the preparation of the e-mails (removing various characters). Another reason can of course be that the non-word is just used in this way.

Generative model - Bayesian estimation

For various words the difference in prior and posterior marginal distributions is checked. Figure 6.4 shows this difference for 2 words identifiable with each category of the confidential dataset. In the figures two distribution lines can be seen, one line is for the prior marginal distribution, the other for the posterior distribution. From these figures it can be concluded that especially the marginal distributions of the words related to the category 'not relevant' have a more fixed distribution (in Figure 6.4 words numbered 4 and 8). This is the case since the marginal posterior distributions are centered around specific values, this might be an indication that the model is very certain about their posterior distributions, and therefore trained well enough on this part. Moreover, the marginal posterior distributions of the words related to the category relevant in the confidential dataset (Figure 6.4, and words 6 and 9) also show centered behaviour (towards the value zero). This might indicate that the classification performance on the confidential dataset with the generative Bayesian model might be good.



(a) Word: 4

(b) Word: 8

(c) Word: 6

(d) Word: 9

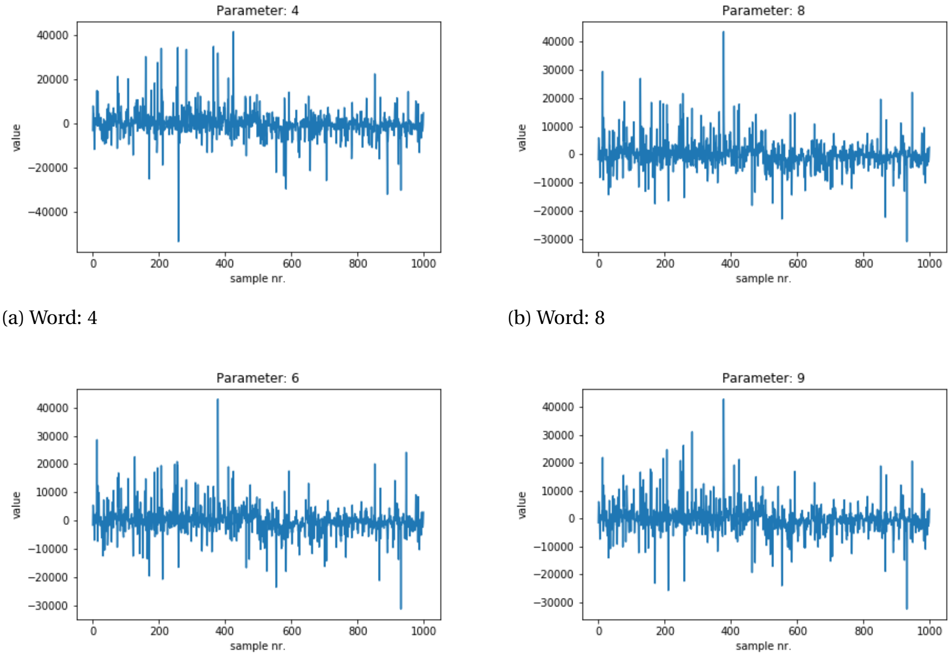
Figure 6.4: Training process generative model with Bayesian estimation (confidential dataset). The two lines indicate the prior and posterior marginal distribution of the corresponding word.

When comparing the marginal posterior distributions of the confidential dataset (Figures F.8 and F.9) with the marginal posterior distributions of the ENRON dataset (Figures F.3 and F.4) the main difference that can be noticed is that the marginal distributions of the words related to the category relevant have a more flat marginal posterior distribution in the case of ENRON. This might indicate that the classification results of the

ENRON dataset will be worse than the classification results of the confidential dataset. As is stated earlier, it was expected that the parameters trained on the ENRON dataset are less fixed, due to the way the dataset is labeled (it is unknown if all relevant e-mails are labeled as relevant or if there are still relevant e-mails in the category unlabeled).

Discriminative model - Bayesian estimation

For the discriminative Bayesian model an analysis will be done based on the samples that are used to determine the values of the parameters. In Figure 6.5 the samples for various parameters are shown. As can be noticed, the difference in sample value are in all 4 cases very big. Most of the samples are near the values of 0, and therefore the parameter values are expected to be near the true value. Which means that the classification results are expected to be good.



(a) Word: 4

(b) Word: 8

(c) Word: 6

(d) Word: 9

Figure 6.5: Training process discriminative model with Bayesian estimation (confidential dataset). The blue line indicates the sample values on which the parameter value is based. As can be noted the sample values show much difference.

6.2.2. Classification analysis

In the previous section it has been explained that the parameters based on the data of the confidential dataset are promising. The classification results of the model based on word frequencies is presented in this section. In Section 1.3 it has already been noted that the feature of word frequencies is successfully used in the spam classification, and therefore the results are worth looking at. When looking at the classification results of Naive Bayes (generative model with maximum likelihood estimation) for the TREC dataset a precision of the spam messages is noted of 93.9%, and a recall of 99.6% (as can be seen in Table F.21). This corresponds to the results presented by Sakkis et al. [50] (maximum recall of 82.35% and a precision of 99.02%). It should be noted that the results of all models perform average to good in the classification of the e-mails in the TREC dataset. The worst performing model is by far the generative Bayesian model with a recall of 14.5% (followed by the generative model with maximum likelihood estimation).

In Table 6.1 the various statistics that have been explained in Section 2.3.2 are shown. These statistics are averages of the 5-fold cross validation that has been applied to the confidential dataset. The statistics of the classification of each training and test set are presented in Appendix F. Based on the Tables F.1 - F.5 and on Table 6.1 it can be concluded that the model based on word frequencies applied on the confidential dataset performs a lot better than it does on the ENRON dataset. An average recall of the category relevant of the Bayesian generative model of 52.1% can be noted. Moreover, when using the same generative Bayesian model, but only with the top 500 words identifiable with relevant and top 500 words identifiable with not relevant e-mails the recall percentage is a constant 100%. This is very useful in practice, and is exactly what a model should be doing.

When comparing the results based on the confidential dataset and the ENRON dataset with the TREC dataset, it can be noted that the classification results are lower. As has been remarked during the analysis of the parameters no very good results were expected (due to the flat posterior distributions, and the spiky behaviour). Furthermore it is also stated earlier that the TREC dataset is much bigger in size than the ENRON/confidential dataset, and therefore better performance of TREC was already expected.

Based on the results the discriminative models are not performing as well as the generative models. The recall of the category relevant of the discriminative models is a lot lower. The same holds for the generative model with maximum likelihood estimation, although this model does perform a little better in terms of recall of the category relevant when compared to the discriminative models.

	MLE (GM)	BE (GM)	MLE (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
Accuracy	85.7%	76.3%	93.18%	90.1%	79.0%	9.1%
'relevant' recall	27.5%	52.1%	0.0%	26.2%	44.8%	100%
'relevant' precision	16.9%	15.9%	n.a.	26.2%	14.9%	6.0%
'not relevant' recall	90.0%	78.2%	100%	94.9%	81.5%	2.4%
'not relevant' precision	94.4%	95.7%	93.18%	94.5%	96.1%	100%

Table 6.1: Average performance results word frequencies (5-fold cross validation), based on confidential dataset. Striking is the average recall of 100% of the Bayesian generative model based on 1000 words as parameters, this is the best performing model present in the table. (GM = Generative model, DM = Discriminative Model, MLE = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category 'relevant' and the top 500 words identified as best corresponding to the category 'not relevant'.

For both Bayesian generative models (with and without parameters based on only 1000 words) additional research has been done. In order to question whether it is luck that the models are performing very well, Bayes Factors have been calculated. Tables F.6-F.15 (two representative tables are also shown below, Table 6.2 and Table 6.3) show that for the generative Bayesian model based on all words the classification of the True positives is so called 'strong' to 'decisive' according to Bayes factor. Whilst for the generative Bayesian model based on the top 1000 words the classification of the true positives is most of the times 'decisive'. This means that the models are certain about their classification (i.e. no probabilities near 0.5), and thus suggests that the models might be useful in practice. The Bayes factors also indicate that the generative Bayesian model based on the top 1000 words is more useful, since its classification is more certain in general (also for the True negatives).

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	7	0	0
$10^2 > B > 10^{\frac{3}{2}}$	0	0	0	0
$10^{\frac{3}{2}} > B > 10$	0	0	0	0
$10 > B > 10^{\frac{1}{2}}$	0	0	0	0
$10^{\frac{1}{2}} > B > 1$	0	0	0	0
$B = 1$	0	0	8	0
$1 > B > 10^{-\frac{1}{2}}$	0	0	0	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	0	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	0	0	1	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	0	0	0	0
$10^{-2} > B$	20	0	363	0

Table 6.2: Bayes Factor results word frequencies for the generative model with Bayesian estimation, based on 1000 words as parameters (set 2). This table shows that the model is certain about its classification.

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	240	0	6
$10^2 > B > 10^{\frac{3}{2}}$	0	11	0	0
$10^{\frac{3}{2}} > B > 10$	0	5	0	0
$10 > B > 10^{\frac{1}{2}}$	0	11	0	0
$10^{\frac{1}{2}} > B > 1$	0	11	0	0
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	0	0	16	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	1	0	12	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	0	0	9	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	1	0	4	0
$10^{-2} > B$	12	0	60	0

Table 6.3: Bayes Factor results word frequencies for the generative model with Bayesian estimation (set 2). This table shows that the model is certain about its classification, in strength of evidence the classification of the true positives would be called 'strong' to 'certain'.

6.2.3. Conclusion

Based on the the results presented in this section it can be concluded that the generative models based on word frequencies perform well enough in order to make the distinction between relevant and not relevant e-mails. Especially the results based on the confidential dataset are promising, with a recall of the generative Bayesian model (based on 1000 words) of 100% on average over 5-fold cross validation. The classification on ENRON does not have similar performances, but also shows that the generative Bayesian models perform best. Furthermore, the classification based on the TREC dataset given results that are comparable to results stated in other research reports (as is also stated in the State of The Art, Section 1.3).

The discriminative models are in neither case able to compete with the results of the generative models. Moreover, as is shown in the performance tables the generative models are in general computationally faster than the discriminative models.

6.3. Word occurrences

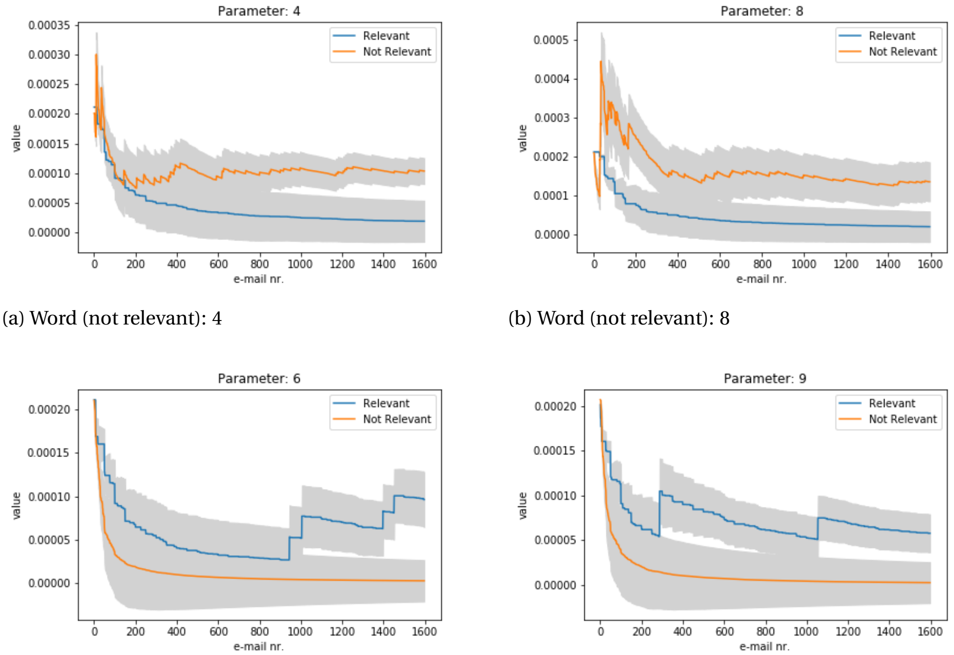
Besides the results of the feature word frequencies, described in the previous section, also results based on the feature word occurrences will be presented. The most important results will be explained in this section, more detailed results can also be found in Appendix F.1, especially results corresponding to the TREC and ENRON dataset, as well as additional results of the 5-fold cross validation.

6.3.1. Parameter analysis

In this section an analysis on the behaviour of the parameters in the various models will be done. By doing this analysis any possible problems of the model can be identified before even looking at the results.

Generative model - MLE

For various words it can be checked how the corresponding parameter is trained. Figure 6.6 show this training process for two words that are identifiable with the two categories (in Section 4.2.1 it has been explained how the words are identifiable with each category). In the Appendix the same kind of figures can also be found for several other words. In the figures two lines can be seen, one line is for the training process of the parameter related to the category relevant, the other to the parameter related to the category not relevant. Furthermore, the grey area around these lines represent one standard deviation. From these figures it can be concluded that especially the parameters of the words related to the category relevant show spiky behaviour (e.g. the words numbered 6 and 9), which might be an indication that the model is not trained well enough on this part. After all, two gradually stabilizing lines would be best. This behaviour is present with words that are identifiable as not relevant e-mails (e.g. the words numbered 4 and 8), but generally speaking not with the category relevant. During the analysis of the results it should be kept in mind that this spiky behaviour is present, the results might increase in performance if more information is available about these words (meaning if more e-mails in the category relevant used).



(a) Word (not relevant): 4

(b) Word (not relevant): 8

(c) Word (relevant): 6

(d) Word: 9

Figure 6.6: Training process generative model for feature word occurrences with MLE (confidential dataset). The two lines indicate how the value of the parameter corresponding to the words are trained. In these figures 'Legit' is the category of the unlabeled e-mails.

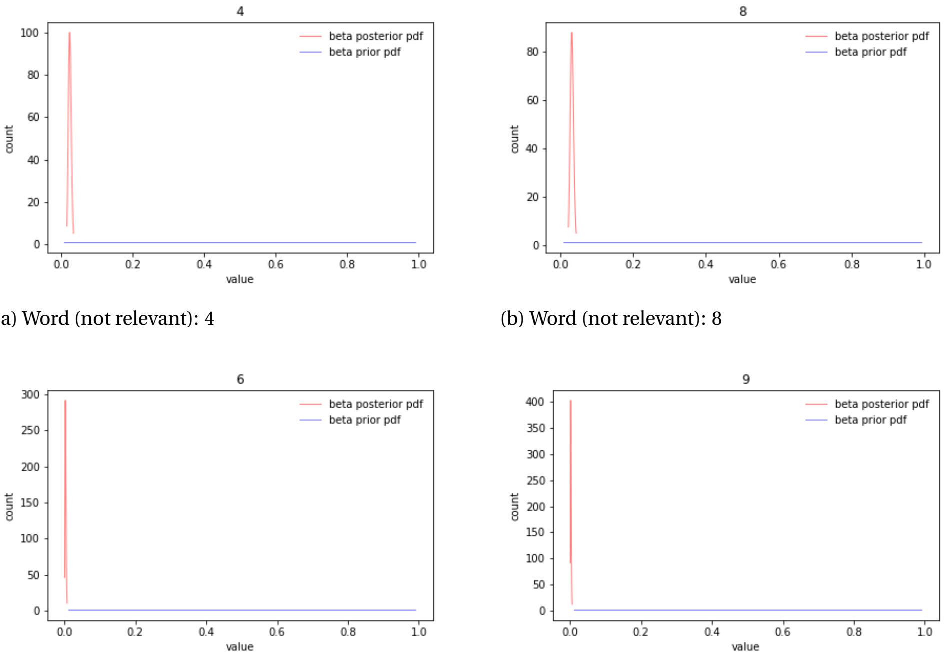
When looking at the differences of the training process between the various fraud related datasets no major differences between the ENRON dataset (Figures F.11 and F.12) and the confidential dataset (Figures F.16 and F.17) can be seen. They both show stabilizing lines for parameters related to the category not relevant, whilst the parameters related to the category relevant show in both cases spiky behaviour. This is also the behaviour seen during the parameter analysis of the models based on word frequencies.

It should be remarked that these figures are unique for the way the model is trained, if the e-mails are shuffled the figures will be slightly different.

Furthermore, as can be seen in Figures F.11 and F.12 there are various non-words present (e.g. developmentenron), these words can occur due to the preparation of the e-mails (removing various characters). Another reason can of course be that the non-word is just used in this way.

Generative model - Bayesian estimation

For various words the difference in prior and posterior marginal distributions will be checked. Figure 6.7 show this difference for two words that are identifiable with the two categories. In the figures two distribution lines can be seen, one line is for the prior marginal distribution, the other for the posterior distribution. From these figures it can be concluded that the marginal prior distributions are very flat, and therefore as far from subjective as possible. It can also be noted that the marginal posterior distributions are fixed distributions. This is the case since the marginal posterior distributions are centered around specific values (in this case near zero), this might be an indication that the model is very certain about their posterior distributions, and therefore trained well enough on this part.



(a) Word (not relevant): 4

(b) Word (not relevant): 8

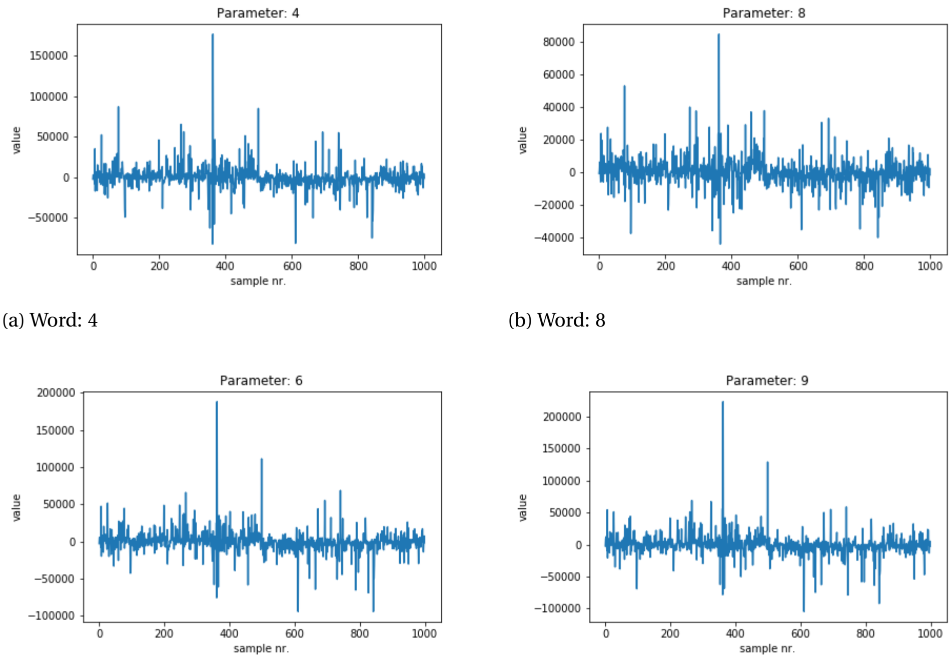
(c) Word (relevant): 6

(d) Word (relevant): 9

Figure 6.7: Training process generative model for feature word occurrences with Bayesian estimation (confidential dataset). The two lines indicate the prior and posterior marginal distribution of the corresponding word.

Discriminative model - Bayesian estimation

For the discriminative Bayesian model an analysis will be done based on the samples that are used to determine the values of the parameters. In Figure 6.8 the samples for various parameters are shown. As can be noticed, the difference in sample value are in all 4 cases very big. This is caused by the very flat normally distributed prior. Most of the samples are near the values of 0, and therefore the parameter values are expected to be near the true value. Which means that the classification results are expected to be good. A similar conclusion has been drawn when the analysis of the samples of the discriminative Bayesian model based on word frequencies was analysed. In that case it has also been concluded that the classification results showed a good recall for the category relevant.



(a) Word: 4

(b) Word: 8

(c) Word: 6

(d) Word: 9

Figure 6.8: Training process discriminative model for feature word occurrences with Bayesian estimation (confidential dataset). The blue line indicates the sample values on which the parameter value is based. As can be noted the sample values show much difference.

6.3.2. Classification analysis

In the previous section it has been shown that especially the parameters based on the data of the confidential dataset are as promising as the parameters for the feature word frequencies. The classification results of the model based on word occurrences is presented in this section. In Section 1.3 it has already been noted that the feature of word occurrences is used in the spam classification, and therefore the results are worth looking at. The classification results based on the TREC for word occurrences are not as good as they are for the word frequencies. Based on word occurrences the best performing model is Naive Bayes (generative model with maximum likelihood estimation) based on 1000 words, with a precision of the spam messages of 99.7%, and a recall of 67.1% (as can be seen in Table F.37). Especially the recall is worse than the recall statistics seen during the analysis of the feature word frequencies. Another good performing model is the discriminative model based on maximum likelihood estimation, with a recall of 56.6% for the category spam. With these performances it is not expected that the classification of fraud related e-mails will have a high performance.

In Table 6.4 the various statistics that have been explained in Section 2.3.2 are shown. These statistics are averages of the 5-fold cross validation that has been applied to the confidential dataset. The statistics of the classification of each training and test set are presented in Appendix F. Based on the tables F.22 - F.26 and on Table 6.4 it can be concluded that the models based on word occurrences applied on the confidential dataset do not perform very well. Although it has already been concluded that the parameters looked similarly trained as in the model of word frequencies, in this case the results are not the same. Only the generative model with maximum likelihood estimation (based on 1000 words as parameters) has a decent recall for the category relevant. The discriminative model with Bayesian estimation is the other model that performs reasonable with a 'relevant' recall of 24.1% on average. It can be noted that the other models do not actually classify any e-mails, but only assign one or the other category to all of the to be predicted e-mails. The reason for the worse performance might be that each word (regardless of whether the word has a high or low influence on the e-mail being fraud related) has a similar influence on the outcome that the e-mail belongs to either category. This results in that the classification might be heavily influenced by words that are of less importance. This reasoning is partially justified by the fact that the Bayesian discriminative model and the generative model with maximum likelihood estimation based on the top 1000 words do perform reasonable.

	MLE (GM)	BE (GM)	MLE (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
Accuracy	92.1%	6.8%	93.2%	91.3%	60.1%	6.8%
’relevant’ recall	0.0%	100%	0.0%	24.1%	58.2%	100%
’relevant’ precision	0.0%	6.8%	n.a.	33.1%	26.7%	6.8%
’not relevant’ recall	98.9%	0.0%	100%	96.3%	60.2%	0.0%
’not relevant’ precision	92.0%	n.a.	93.18%	94.5%	95.1%	n.a.

Table 6.4: Average performance results word occurrences (5-fold cross validation), based on confidential dataset. Based on these averages the Bayesian generative model based on 1000 words as parameters is the best performing model present in the table. However, since this model does not have any practical advantage (it classifies every e-mail as ’relevant’) the best model for practical usage would be the generative model with maximum likelihood estimation based on 1000 words as parameters (GM = Generative model, DM = Discriminative Model, MLE = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’.

On average the best performing model in Table 6.4 is the generative model based on maximum likelihood estimation with parameters based on the top 1000 words (top 500 words identifiable with the category relevant and top 500 words identifiable with the category not relevant), it has been noticed that the generative model based on maximum likelihood estimation with all words has an average recall of 0.0%. This might show that the classification is highly dependent on the parameters (i.e. words on which the model is trained) chosen. Table 6.5 (and Tables F.27-F.27 in the Appendix) show that the generative model with with maximum likelihood estimation based on the top 1000 words is not very certain about its classification. As can be noticed there are quite a number of True negative classifications that have so called only ’slight evidence’ against the other class. This means that the model has its doubts about the classification of these e-mails. On the other hand there are also a number of false negatives that only have ’slight evidence’ against the other class, meaning that if the decision boundary would be shifted a few more true positives can easily be added (i.e. changing the probability of 0.5 that has been used to make the predictive classification of belonging to one class or the other). The observation that there is in various cases only ’slight evidence’ for the classification of the other class can also be the reason that the generative model based on maximum likelihood and all words does not perform very well, which gives additional reasons that the classification of this model is based on the parameters (i.e. words) used.

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	171	0	13
$10^2 > B > 10^{\frac{3}{2}}$	0	14	0	2
$10^{\frac{3}{2}} > B > 10$	0	14	0	1
$10 > B > 10^{\frac{1}{2}}$	0	20	0	1
$10^{\frac{1}{2}} > B > 1$	0	8	0	1
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	0	0	25	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	10	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	3	0	30	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	1	0	7	0
$10^{-2} > B$	10	0	65	0

Table 6.5: Bayes Factor results word occurrences for the generative model with maximum likelihood estimation based on the top 1000 words (set 1). This table shows that the model is not very certain about its classification. Many classifications are near $B = 1$, meaning that the probabilities are almost equal. The true positives are rather ’certain’.

6.3.3. Conclusion

Generally speaking the feature word occurrences does not perform very well in order to give e-mails a good classification. Based on the confidential dataset the best performing model is the generative model with maximum likelihood estimation, with an average recall of the category relevant of 58.2%. Looking at the Bayes factor indicates that the model has in several cases only ’slight evidence’ against the other class. Other models (except for the discriminative Bayesian model) do not classify other than applying one category to all to be predicted e-mail messages, in the case of the confidential dataset.

Moreover, as is shown in the performance tables the generative models are in general computationally faster than the discriminative models.

6.4. E-mail length

Results based on the feature of the length of e-mail will be presented in this chapter . The most important results will be explained in this section, more detailed results can also be found in Appendix F.1, especially results corresponding to the TREC and ENRON dataset, as well as additional results of the 5-fold cross validation.

6.4.1. parameter analysis

In this section an analysis on the behaviour of the parameters in the various models will be done. By doing this analysis any possible problems of the model can be identified before even looking at the results.

Generative model - MLE

It has been checked how the parameter of the model based on the feature of lengths is trained. Figure 6.9 shows this training process. In this figure two lines can be seen, one line is for the training process of the parameter related to the category relevant, the other to the parameter related to the category not relevant. Furthermore, the grey area around these lines represent one standard deviation. From this figure it can be concluded that the parameters seem stabilized after the training process. It can be remarked that the values of both parameters are very much the same (i.e. the lines are near each other at the right side of the plot), which means that little distinction can be made between the e-mails when regarding the length of e-mails in each category. This can have impact on the classification performance, since it is better to have bigger differences between the models. In Chapter 4 it has already been concluded that not much difference between lengths is present, but especially the outlying values might be distinguished with this model.

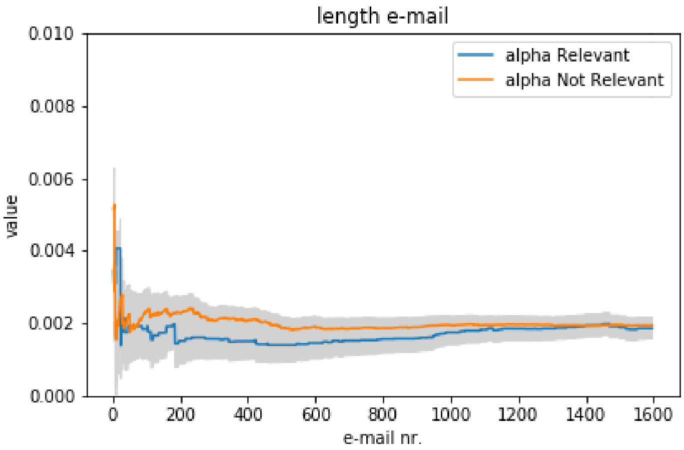


Figure 6.9: Training process generative model for feature length e-mail with MLE (confidential dataset). The two lines indicate how the value of the parameter corresponding to the words are trained. In these figures 'Legit' is the category of the unlabeled e-mails.

It should be remarked that these figures are unique for the way the model is trained, if the e-mails are shuffled the figures will be slightly different.

Furthermore, the parameters of the model trained with the ENRON dataset again shows similar characteristics. This can be found in Figure F.21. Therefore, similar performance statistics will be expected based on this parameter analysis.

Generative model - Bayesian estimation

The difference in prior and posterior marginal distributions has also been checked. Figure 6.10 shows this difference. In the figure two distribution lines can be seen, one line is for the prior marginal distribution, the other for the posterior distribution. From this figure it can be concluded that the marginal posterior distributions are very flat, and therefore it is expected that the classification results are not very good. In general a model with more fixed marginal posterior distributions has better classification performances, as has already been noted with the models based on word frequencies.

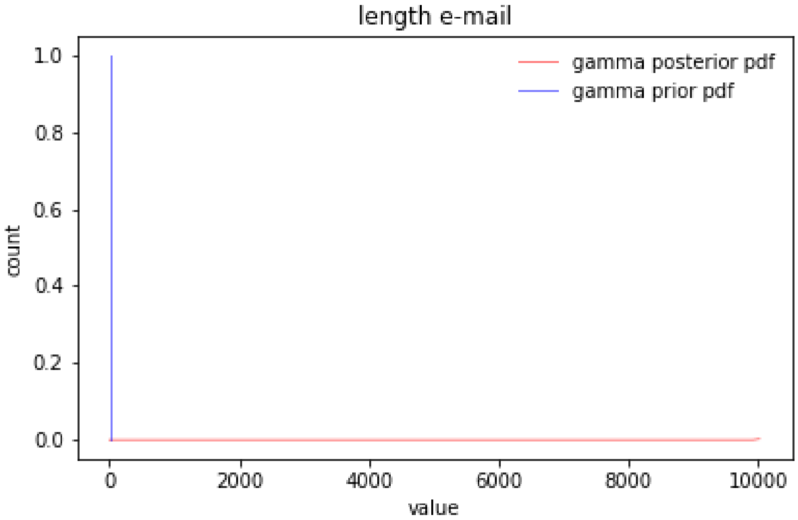


Figure 6.10: Training process generative model for feature length e-mail with Bayesian estimation (confidential dataset). The two lines indicate how the value of the parameter corresponding to the words are trained. In these figures 'Legit' is the category of the unlabeled e-mails.

Discriminative model - Bayesian estimation

For the discriminative Bayesian model an analysis has been done based on the samples that are used to determine the value of the parameter. In Figure 6.11 the samples for various parameters are shown. As can be noticed, the difference in sample value are not very big (in contrast with the samples analysed for the features word frequencies and word occurrences). Most of the samples are near the values of 0, and therefore the parameter values are expected to be near the true value. Which means that the classification results are expected to be good. The model trained by the ENRON dataset shows similar characteristics, as can be seen in Figure F.23.

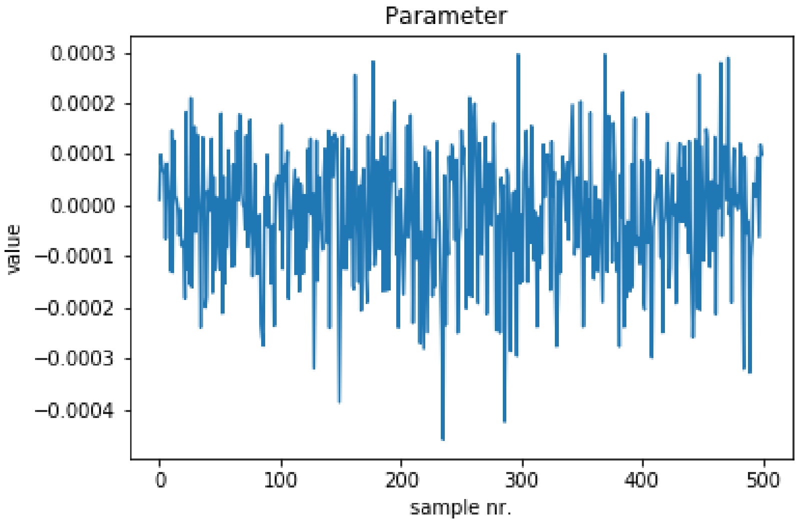


Figure 6.11: Training process discriminative model for feature length e-mail with Bayesian estimation (confidential dataset). The blue line indicates the sample values on which the parameter value is based. As can be seen the sample values are quite close to each other.

6.4.2. Classification analysis

In the previous section it has been shown that especially the parameters based on the data of the confidential dataset are not as promising as the parameters for the feature word frequencies and word occurrences. The classification results of the model based on word occurrences is presented in this section. The classification results based on the TREC for word occurrences are not as good as they are for the word frequencies but better than for word occurrences. The best performing model is the generative model based on Bayesian estimation, with a precision of the spam messages of 67.3%, and a recall of 98.8% (as can be seen in Table F.53). Another good performing model is the generative maximum likelihood model, with a recall of 93.8% for the category spam. It is also noticed that the discriminative model based on maximum likelihood estimation does not actually perform any classification other than assigning the same label to all e-mails. With these performances it is not expected that the classification of fraud related e-mails will have a reasonable performance for some models, but not for all.

In Table 6.6 the various statistics that have been explained in Section 2.3.2 are shown. These statistics are averages of the 5-fold cross validation that has been applied to the confidential dataset. The statistics of the classification of each training and test set are presented in Appendix F. Based on the tables F.38 - F.42 and on Table 6.6 it can be concluded that the model based on e-mail length applied on the confidential dataset does not perform very well. It has already been concluded that the parameters (except for the parameter of the discriminative Bayesian model) looked as if good classification results would become a problem. Only the generative model with maximum likelihood estimation has a decent recall for the category relevant. It can be noted that the other models do not actually classify any e-mails, but only assign one or the other category to all of the to be predicted e-mails. The reason for the worse performance might be that there is not enough distinction between the lengths of the e-mail of the two categories. The same conclusion has already been made in Chapter 4, although it has also been concluded there that especially the outlying lengths are different between the categories. In the classification such results are not noticed.

In Table 6.7 one of the tables with the Bayes factor outcomes is shown (this table is based on cross validation set 1, but shows similar outcomes as with the other sets, the outcomes of the other sets can be found in Table F.43-F.47). As expected the model is not sure about its classifications, this can be seen with all the e-mails being in the 'strength of evidence' class named 'slight evidence'. The behaviour is, as already stated, most likely the cause of the worse performance.

	MLE (GM)	BE (GM)	MLE (DM)	BE (DM)
Accuracy	84.7%	6.8%	93.2%	93.2%
'relevant' recall	12.7%	100%	0.0%	0.0%
'relevant' precision	7.0%	6.8%	n.a.	n.a.
'not relevant' recall	87.8%	0.0%	100%	100%
'not relevant' precision	93.2%	n.a.	93.2%	93.2%

Table 6.6: Average performance results length e-mail (5-fold cross validation), based on confidential dataset. Based on these averages the only practically usable model is the generative model based with maximum likelihood estimation. All other models are not performing any classification other than assigning one class to all e-mail messages. (GM = Generative model, DM = Discriminative Model, MLE = Maximum Likelihood Estimation, BE = Bayesian Estimation).

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	0	0	0
$10^2 > B > 10^{\frac{3}{2}}$	0	0	0	0
$10^{\frac{3}{2}} > B > 10$	0	0	0	0
$10 > B > 10^{\frac{1}{2}}$	0	0	0	0
$10^{\frac{1}{2}} > B > 1$	0	328	0	27
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	5	0	39	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	0	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	0	0	0	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	0	0	0	0
$10^{-2} > B$	0	0	0	0

Table 6.7: Bayes Factor results of e-mail lengths for the generative model with maximum likelihood estimation (set 1). This table shows that the model is not certain about its classification.

6.4.3. Conclusion

Generally speaking the feature of e-mail length does not perform very well in order to give e-mails a good classification, this is due to the limited difference of e-mail lengths between the categories. Based on the confidential dataset the best performing model is the generative model with maximum likelihood estimation, with an average recall of the category relevant of 12.7%. Other models do not classify other than applying one category to all to be predicted e-mail messages, in the case of the confidential dataset. Furthermore, the classification based on the ENRON dataset does not show much better or different results.

7

Additional classification model: AdaBoost

Based on the results presented in the previous Chapter an additional model will be introduced and only limitedly analysed. The additional model is the machine learning model AdaBoost that shows promising results in difficult classification problems. As has been noted the models performed best on the confidential dataset. The reason that has been given is that the confidential dataset is most likely the best representative dataset for fraud classification, and the others probably lack data quality. Therefore, to save computing time, in this chapter the AdaBoost model has only be applied to the confidential dataset.

7.1. Classification model

AdaBoost is an iterative algorithm that uses different weights in each simulation during the training process. The weights are increased or decreased based on the misclassification at each iteration. In this way AdaBoost might be able to give better classifications. As is stated by Weiss [60] "rare classes are more error-prone than common classes, and therefore it is reasonable to belief that boosting may improve their classification performance."

Besides the reason that AdaBoost might improve performance of the imbalanced datasets, especially in the classification of rare classes (i.e. the relevant e-mail messages), AdaBoost has also been chosen as additional classification model to compare the classification performance of the (full) Bayesian models described in Chapter 5 with a already implemented machine learning model.

For the classification of the data with AdaBoost the function AdaBoost of the package sk.learn will be used. A detailed description of AdaBoost is given by for example Freund and Schapire [23] or Bishop [7]. AdaBoost from the package sk.learn uses the algorithm described by Zhu et al. [63].

AdaBoost uses many 'weak learners' (learning algorithms that perform just above random classification) in order to create a predictive model that has a high classification performace. The same features are used (namely word frequencies, word occurrences and length of an e-mail) as have been described in Chapter 1 and as have been used in the previous chapters. The output of the AdaBoost algorithm is the class label of the to be predicted e-mail. The package sk.learn uses decision trees as weak learners.

7.2. Results

The various results based on the confidential dataset are given in Tables 7.1 - 7.3 (statistics based on word frequencies, word occurrences and e-mail lengths respectively). It should be remarked that due to computation limitations only 1000 parameters (top 500 words identifying relevant and top 500 words identifying not relevant e-mail messages) are used. It can in general be noticed that AdaBoost does not perform well in classifying relevant e-mail messages. This was not expected since AdaBoost is known for its good classification of rare classes.

The maximum recall of the category relevant over all results 9.7%, whilst in the previous chapter multiple times a recall of 20% till even 50% on average has been seen. This gives additional reason to believe that the best performing models found in the previous chapter are indeed actually good models to use in practice.

	set 1	set 2	set 3	set 4	set 5
# Predicted = given	362	379	368	377	358
# False Positives	5	0	0	0	11
# False Negatives	32	20	31	22	30
# True Positives	0	0	0	0	1
# True Negatives	362	379	368	377	357
Accuracy	90.7%	95.0%	92.2%	94.5 %	89.7%
Error rate	9.3%	5.0%	7.8%	5.5%	10.3%
’relevant’ recall	0.0%	0.0%	0.0%	0.0%	3.2%
’relevant’ precision	0.0%	n.a.	n.a.	n.a.	8.3%
’not relevant’ recall	98.6%	100%	100%	100%	97.0%
’not relevant’ precision	91.9%	95.0%	92.2%	94.5%	92.2%
’relevant’ F-score	n.a.	n.a.	n.a.	n.a.	4.6%
’not relevant’ F-score	95.1%	95.1%	95.1%	97.0%	94.5%
’relevant’ percentage	8.0%	5.0%	7.8%	5.5%	7.8%
Time (test)	1049 sec	1350 sec	802 sec	1224 sec	1023 sec
Time (training)	9440 sec	9275 sec	9035 sec	9420 sec	9081 sec

Table 7.1: Performance results, based on the feature word frequencies of e-mail messages in the confidential dataset. AdaBoost is used to perform the classification, and the top 1000 words are used as parameters in the model.

	set 1	set 2	set 3	set 4	set 5
# Predicted = given	352	379	362	376	352
# False Positives	15	0	7	1	19
# False Negatives	32	20	30	22	28
# True Positives	0	0	1	0	3
# True Negatives	352	379	361	376	349
Accuracy	88.2%	95.0%	90.7%	94.2%	88.2%
Error rate	11.8%	5.0%	9.3%	5.8%	11.8%
’relevant’ recall	0.0%	0.0%	3.2%	0.0%	9.7%
’relevant’ precision	0.0%	n.a.	12.5%	0.0%	13.6%
’not relevant’ recall	95.9%	100%	98.1%	99.7%	94.8%
’not relevant’ precision	91.7%	95.0%	92.3%	94.5%	92.6%
’relevant’ F-score	n.a.	n.a.	5.1%	n.a.	11.3%
’not relevant’ F-score	93.8%	95.1%	95.1%	97.0%	93.7%
’relevant’ percentage	8.0%	5.0%	7.8%	5.5%	7.8%
Time (test)	766 sec	966 sec	607 sec	918 sec	719 sec
Time (training)	9229 sec	8907 sec	9122 sec	9067 sec	9117 sec

Table 7.2: Performance results, based on the feature word occurrences of e-mail messages in the confidential dataset. AdaBoost is used to perform the classification, and the top 1000 words are used as parameters in the model.

	set 1	set 2	set 3	set 4	set 5
# Predicted = given	367	378	368	376	367
# False Positives	0	1	0	1	1
# False Negatives	32	20	31	22	31
# True Positives	0	0	0	0	0
# True Negatives	367	378	368	376	367
Accuracy	92.0%	94.7%	92.2%	94.2%	92.0%
Error rate	8.0%	5.3%	7.8%	5.8%	8.0%
’relevant’ recall	0.0%	0.0%	0.0%	0.0%	0.0%
’relevant’ precision	n.a.	0.0%	n.a.	0.0%	0.0%
’not relevant’ recall	100%	99.7%	100%	99.7%	99.7%
’not relevant’ precision	92.0%	95.0%	92.2%	94.5%	92.2%
’relevant’ F-score	n.a.	n.a.	n.a.	n.a.	n.a.
’not relevant’ F-score	95.8%	97.3%	95.9%	97.0%	95.8%
’relevant’ percentage	8.0%	5.0%	7.8%	5.5%	7.8%
Time (test)	5 sec	5 sec	4 sec	5 sec	5 sec
Time (training)	19 sec	19 sec	19 sec	19 sec	19 sec

Table 7.3: Performance results, based on the feature of e-mail lengths of e-mail messages in the confidential dataset. AdaBoost is used to perform the classification, and the top 1000 words are used as parameters in the model.

7.3. Conclusion

Based on the short analysis AdaBoost does not show any promising results. It was expected to increase the performance found in the previous chapter, but the classification results show that a decrease in performance is found. It might therefore indicate that the best performing models in the previous chapter are indeed good models to use in practice.

8 Conclusion, discussion and recommendations

*"A complex system that works is invariably found to have evolved from a simple system that worked.
A complex system designed from scratch never works and cannot be patched up to make it work.
You have to start over with a working simple system."*

John Gall (1975)

The quote from John Gall perfectly summarises the premise of the research presented in this thesis. First a simple model that works needs to be found, in order to try anything more advanced. In the Introduction, and in the summary of the state of the art research it was emphasized that currently only limited to no research is available that is similar as the research done in this thesis. The goal of this thesis was to find techniques that are able to be trained on historical datasets in order to give a classification of new e-mails in terms of whether they are relevant or not relevant to a new e-Discovery case. In the Introduction the following research question was formulated: *Can spam filtering techniques be used as viable techniques for detecting fraud related (i.e. relevant) e-mails?*

In the following sections first the findings are concluded and the research question is answered. After that the results are discussed, and any difficulties or consequences of assumptions are pointed out. Finally recommendations for future research are given based on the findings.

8.1. Conclusion

Based on the results presented in this thesis it has to be concluded that the best performing feature is word frequencies. Although no model has been found of which with certainty can be said that the results are good enough to be put into practice directly, the generative model with Bayesian estimation showed an average recall of 51.1% and when taking into account only the top 1000 words even an average recall of 100%. It is therefore concluded that this model taken from the spam classification might be a viable way for the detection of relevant e-mail messages. It has been remarked that the parameters of this model are most likely not trained well enough (due to the spiky behaviour during the training process), and therefore the classification performance is expected to increase if better and more data is available. Furthermore, by the means of Bayes factor analysis it has been concluded that the classes assigned by the generative Bayesian model are so called 'strong' to 'decisive' in terms of strength of evidence. The other models did not perform as good as the generative Bayesian model. The generative model with Bayesian estimation is also one of the models for which it holds that the classification can be computed relatively fast, which is useful for using the model in practice.

The classification based on word occurrences or e-mail lengths did not perform as expected and did not get near the results that were found with the word frequencies.

8.2. Discussion

As has already been remarked in the various chapters that described the implementation of the features, one of the major drawbacks is the nature of the data of an e-Discovery case. This drawback consists of three parts, namely the unbalancedness of the data, the data quality as well as the availability of the data. As has been noted in the report data used in e-Discovery cases are very unbalanced, it is not uncommon to have between 1% and 10% of relevant e-mail in the dataset. This can have influence on the results shown in this thesis, since the parameters are estimated based on the available data. Therefore if from one class a lot of data is available, and from the other class only limited data, then various models are biased towards the larger class. Furthermore, it has been remarked that the data quality of especially the ENRON dataset is not high. The results showed low classification performance based on the ENRON dataset, it is expected that the data quality is the major cause. It has been decided to include the ENRON dataset (regardless of its quality) in order for future research to be able to compare results. Moreover, and closely related to the previous two discussion parts on the aspect of data, there is not much data available. As might already have been noted, the current models have been applied in a cross validation setting, whilst the original idea was to train on one dataset and classify on a new dataset. However, no additional datasets were available at the time of doing research, and therefore the analysis of cross validation is done.

Secondly, it has not been studied which influence the priors have on the classification performances. The prior values used are tried to be kept as objective as possible, by using flat priors and values for the hyper parameters that are widely accepted, but the influence of the prior on the posterior distribution is not measured. In order to keep the research focused on the goal of the thesis, it is decided to not include an analyse on the influence of the priors. It is possible that the priors used have a big influence on the classification performance, and that an other prior of hyper parameter value would have resulted in better results.

Furthermore, the models are based on various assumptions. It is known that two of the used assumptions (independence of words and e-mails) are in practice not valid. Despite not being valid these assumptions are used in order to make the models more useful in practice and computationally more efficient. Many other research report use the same assumptions, but it might be the case that these assumptions influence our results more than expected.

Moreover, the pre-processing of the e-mails can be questionable. It is chosen to remove any characters other than the small and capital letters. In this way the original text is adapted and therefore unique characteristics that can be of importance in the classification of relevant e-mails might be removed. At the same time no stemming or natural language processing is applied in order to keep the original text, and to be able to classify based on the original words used. Keeping the original text results in many different unique parameters (i.e. words) whilst stemming or natural language processing results in more counts per feature, but less unique features. In general both possibilities have their own advantages and disadvantages.

Finally, the consequences of using classification model should be kept in mind. A model that is able to detect relevant e-mails is useful in terms of money and time. However, the model is only trained on historical data, and therefore a natural bias will be present. If historically certain messages are labeled as relevant that were not actually relevant, then in the future anything similar will of course also be labeled relevant. Furthermore, if the actual fraud in a new case is not in the trainingsset, but some messages in the case are pointed out by the model as being relevant a tunnel vision might be created. It is always advisable to keep in mind that a model only gives an indication, but never knows with certainty that the messages labeled as relevant are indeed relevant to the case. The models discussed are not able to detect the meaning of words, only their usage.

8.3. Recommendations

Although in this thesis different models and datasets are looked into, research is never finished. It has especially been noted that this thesis was focused on exploratory research. Therefore, various recommendations for further research will be given in this section.

First of all, it is recommended for any organisation, but especially KPMG since the thesis was written in cooperation with Forensic Technology, to create a database of datasets that can be used for training these kind of models. As has been noted throughout the report, the quality of the (publicly) available data is currently not very high. Public data is simply not available, and in order to be able to develop a model that is able to be put into practice, quite a number of datasets are needed in order to train a model that is able to perform well enough.

Secondly, this thesis has only looked at a certain selection of models. The models have been chosen based on the available information, but this does not mean that other models cannot perform better. At the end of the thesis an additional model, AdaBoost, has been looked into. Although AdaBoost did not perform well, it might be that for example Support Vector Machines perform a lot better.

Furthermore, the application of stemming or language processing in general might improve results significantly. In this thesis it has been decided not to apply natural language processing since information is removed. However, it has been concluded that the words identifiable with fraud are not present in high quantities. Natural language processing might result in higher counts of certain features, and therefore better overall performance.

Besides the development of a model that is able to select fraudulent messages. It is also useful for an e-Discovery case to look into models that remove non fraudulent messages (i.e. news articles, spam, newsletters etc.). In this way the size of the dataset that still needs to be review is decreased, and therefore the review process is faster.

Moreover, a straightforward recommendation is looking into new features. This thesis has only looked into word frequencies, word occurrences and length of e-mails, however many more features are available. Such as the number of recipients, the day of the week the e-mail is sent, the presence of an attachment, etc. It is even possible to create a overall model that combines different features. In this case the most computational efficient way is to assume that features are independent, but taking dependence of features (or even words/e-mails) into account might improve the classification results.

Furthermore, it has been noted that the e-mail datasets related to e-Discovery cases are in almost all cases unbalanced datasets. Only limited information is available on the way unbalanced unstructured datasets need to be handled, and therefore additional research can be added.

looking into the threshold of $p = 0.5$ might improve performance. By the means of Bayes factor it has been concluded that no large performance increase was possible, but by determining the most optimal threshold value might just give the additional increase in performance in order to make a model useful in practice.
Last but not least, based on the conclusion it is recommended to do additional research on the generative Bayesian model based on word frequencies (i.e. Extended Naive Bayes) in order to improve the performance. There are multiple ways to improve the performance, such as by taking different prior distributions, other classification rules or by looking at the word frequencies used. Each of these are different assumptions of the model, and by changing these assumptions better classification might be achieved.

Bibliography

[1] ANDROUTSOPOULOS, I., KOUTSIAS, J., CHANDRINOS, K., AND SPYROPOULOS, C. An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages.

[2] ANDROUTSOPOULOS, I., PALIOURAS, G., AND MICHELAKIS, E. Learning to filter unsolicited commercial e-mail. Tech. rep., NCSR "Demokritos", 2006.

[3] ASPOSE. Python apis for email processing. `https://products.aspose.com/email/python-net`.

[4] BASAVARAJUL, M., AND PRABHAKAR, D. A novel method of spam mail detection using text based clustering approach. *International Journal of Computer Applications 5*, 4 (Aug. 2010), 15–25.

[5] BHALERAO, D., AND INGLE, D. Novel survey on email spam filtering methods. *International Journal on Recent and Innovation Trends in Computing and Communication 6*, 4 (Apr. 2018), 303– 306.

[6] BHOWMICK, A., AND HAZARIKA, S. E-mail spam filtering: A review of techniques and trends. In *Advances in Electronics, Communication and Computing* (2018), A. K. S. D. K. Sharma, Ed., vol. 443.

[7] BISHOP, C. *Pattern Recognition and Machine Learning.* Springer, 2006.

[8] BLANZIERI, E., AND BRYL, A. A survey of learning-based techniques of email spam filtering. *Artif Intell Rev 29*, 1 (Mar. 2008), 63–92.

[9] CHEN, Y., WU, C., CHEN, Y., LI, H., AND CHEN, H. Enhancement of fraud detection for narratives in annual reports. *International Journal of Accounting Information Systems 26* (2017), 32–45.

[10] CONSULTANCY.NL. Pwc: Fraude kost nederlander 600 euro per jaar. `https://www.consultancy.nl/nieuws/7513/pwc-fraude-kost-nederlander-600-euro-per-jaar`, Dec. 2013.

[11] CORMACK, G. Trec 2007 spam track overview.

[12] DE BRUIJN, A. Naar een fraudebeeld nederland: Inzicht in fraude draagt bij aan bewustwording en effectieve prioriteitsstelling in de aanpak. `https://kennisopenbaarbestuur.nl/media/53886/pwc-naar-een-fraudebeeld-nederland.pdf`, Dec. 2013.

[13] DEFAZIO, A., BACH, F., AND LACOSTE-JULIEN, S. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives.

[14] EDRM. New edrm enron email data set. `https://www.edrm.net/resources/data-sets/edrm-enron-email-data-set/`.

[15] EDRM. Technology assisted review. `https://www.edrm.net/frameworks-and-standards/technology-assisted-review/`.

[16] EDRM. Quick reference edrm workflow poster — a training tool for legal professionals and e-discovery practitioners. `https://www.edrm.net/frameworks-and-standards/edrm-model/edrm-wall-poster/`, July 2018.

[17] FINANCIEEL DAGBLAD. Affaire rentederivaten. `https://fd.nl/dossier/rentederivaten`.

[18] FINANCIEEL DAGBLAD. Belastingfraude in curaçao. `https://fd.nl/dossier/curacao`.

[19] FINANCIEEL DAGBLAD. Steinhoff stelt publicatie jaarcijfers verder uit. `https://fd.nl/ondernemen/1296043/steinhoff-stelt-publicatie-jaarcijfers-verder-uit`, Apr. 2019.

[20] FISSETTE, M. *Text mining to detect indications of fraud in annual reports worldwide.* phdthesis, University of Twente, 2017.

[21] FRANK, E., AND BOUCKAERT, R. Naive bayes for text classification with unbalanced classes. *PKDD* (2006), 503–510.

[22] FREED, N., AND BORENSTEIN, N. *Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples.* Network Working Group, Nov. 1996.

[23] FREUND, Y., AND SCHAPIRE, R. A decision-theoretic generalization of on-line learning and an application to boosting. *Computer and system sciences 55* (1997), 119–139.

[24] GALL, J. `https://en.wikiquote.org/wiki/John_Gall`, 1975.

[25] GELMAN, A., CARLIN, J., STERN, H., DUNSON, D., VEHTARI, A., AND RUBIN, D. *Bayesian Data Analysis*, third ed. Taylor & Francis Group, 2014.

[26] GLANCY, F., AND YADAV, S. A computational model for financial reporting fraud detection. *Decision Support Systems 50* (Aug. 2011), 595–601.

[27] GUZELLA, T., AND CAMINHAS, W. A review of machine learning approaches to spam filtering. *Expert Systems with Applications 36* (2009), 10206–10222.

[28] HAGGERY, J., KARRAN, A., LAMB, D., AND TAYLOR, M. A framework for the forensic investigation of unstructured email relationship data. *International Journal of Digital Crime and Forensics 3*, 3 (July 2011), 1–18.

[29] HAWKINS, D. Identification of outliers, 1980.

[30] HOFFMAN, M., AND GELMAN, A. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo.

[31] HUMPHERYS, S., MOFFITT, K., BURNS, M., BURGOON, J., AND FELIX, W. Identification of fraudulent financial statements using linguistic credibility analysis. *Decision Support Systems 50* (Aug. 2011), 585–594.

[32] JEFFREYS, H. *Theory of Probability*, third edition ed. Oxford University Press, 1961.

[33] KEILA, P., AND SKILLICORN, D. Detecting unusual and deceptive communication in email, 2005.

[34] KUPERUS, W. Classifying deceptive emails with first and third person pronouns: Applying support vector machine on the enron email dataset. mathesis, Tilburg University, 2017.

[35] LAI, C., AND TSAI, M. An empirical performance comparison of machine learning methods for spam e-mail categorization. In *Proceedings of the Fourth International Conference on Hybrid Interlligent Systems (HIS'04)* (2004).

[36] LAUNCHPAD. Beautiful soup. `https://www.crummy.com/software/BeautifulSoup//#Download`.

[37] LAWTON, D., STACEY, R., AND DODD, G. ediscovery in digital forensic investigations, Sept. 2014.

[38] LUCIAN, C., AND CRISTINA, D. Fraud case analysis: ENRON corporation.

[39] MAHAPATRA, A., SRIVASTAVA, N., AND SRIVASTAVA, J. Contextual anomaly detection in text data. *Algorithms* (2012), 469–489.

[40] METSIS, V., ANDROUTSOPOULOS, I., AND PALIOURAS, G. Spam filtering with naive bayes – which naive bayes? In *CEAS 2006 Third Conference on Email and AntiSpam* (July 2006).

[41] MITCHELL, T. *Machine Learning.* 2017, ch. Generative and Discriminative Classifiers: Naive Bayes and Logistic Regression.

[42] MOON, J., SHON, T., SEO, J., KIM, J., AND SEO, J. An approach for spam e-mail detection with support vector machine and n-gram indexing. In *International Symposium on Computer and Information Sciences* (2004), C. et al, Ed., pp. 351–362.

[43] MUJTABA, G., SHUIB, L., RAJ, R., MAJEED, N., AND AL-GARADI, M. Email classification research trends: Review and open issues. *IEEE Access 5* (May 2017), 9044–9064.

[44] NEWMAN, M. Power laws, pareto distributions and zipf's law. *Contemporary Physics 46*, 5 (2005), 323–351.

[45] NORTHRUP, J., AND GERBER, B. A comment on priors for bayesian occupancy models. *PLoS One 13*, 2 (2018).

[46] NUIX, AND EDRM. Removing pii from the edrm enron data set.

[47] OPENBAAR MINISTERIE. Fraude. `https://www.om.nl/onderwerpen/fraude/`.

[48] PYTHON SOFTWARE FOUNDATION. email — an email and mime handling package. `https://docs.python.org/2.7/library/email.html`.

[49] RESNICK, P. *Internet Message Format.* Network Working Group, Apr. 2001.

[50] SAKKIS, G., ANDROUTSOPOULOS, I., PALIOURAS, G., KAKALETSIS, V., AND SPYROPOULOS, C. A memory-based approach to anti-spam filtering for mailing lists. *Information Retrieval 6* (2003), 49–73.

[51] SAPPELLI, M., VERBERNE, S., AND KRAAIJ, W. Combining textual and non-textual features for e-mail importance estimation.

[52] SCIKIT-LEARN. Generalized linear models. `https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression`.

[53] SEAMAN, J. *Topics in Bayesian Inference: Induced Priors, Proof Loading for Combination Drugs, and Distribution of Archaeological Skeletal Assemblages.* PhD thesis, Baylor University, 2010.

[54] SHARAFF, A., NAGWANI, N., AND DHADSE, A. Comparative study of classification algorithms for spam email detection. *Emerging Research in Computing, Information, Communication and Applications* (2016), 237–244.

[55] SHETTY, J., AND ADIBI, J. The enron email dataset: Database schema and brief statistical report.

[56] THE UNITED STATES DEPARTMENT OF JUSTICE. ENRON trial exhibits and releases. `https://www.justice.gov/archive/index-enron.html`, Mar. 2014.

[57] TREC. Spam track. `https://trec.nist.gov/data/spam.html`, Feb. 2017.

[58] TUTEJA, S. A survey on classification algorithms for email spam filtering. *International Journal of Engineering Science and Computing 6*, 5 (2016), 5937–5940.

[59] U.S. SECURITIES AND EXCHANGE COMMISSION. Spotlight on enron. `https://www.sec.gov/spotlight/enron.htm`, May 2010.

[60] WEISS, G. Mining with rarity: A unifying framework. *Sigkdd Explorations 6*, 1 (unknown).

[61] YOUNG, G., AND SMITH, R. *Essentials of Statistical Inference.* Cambridge University Press, 2005.

[62] ZHANG, L., ZHU, J., AND YAO, T. An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing 3*, 4 (Dec. 2004), 243–269.

[63] ZHU, J., ROSSET, S., ZOU, H., AND HASTIE, T. Multi-class adaboost.

A

Mathematical background - distributions

Throughout the report various distributions will be used. For completeness these distributions will be stated in the following subsections.

A.1. Multinomial distribution

The multinomial distribution is a generalization of the binomial distribution. The distribution has two parameters namely n, the number of trials, and $p_1,\ldots,p_k$, the event probabilities (with $\sum_{i=1}^{k} p_i = 1$). Furthermore, it uses the support $x_i \in \{0,\ldots,n\}$ with $i \in \{1,\ldots,k\}$ and $\sum_{i=1}^{k} x_i = n$.

The probability density function is given by:

$$f(x_1,\ldots,x_k;p_1,\ldots,p_k) = \frac{n!}{\prod_{i=1}^{k} x_i!} \cdot \prod_{i=1}^{k} p_i^{x_i}.$$

The conjugate prior of the Multinomial distribution is the Dirichlet distribution.

A.2. Dirichlet distribution

The Dirichlet distribution has two parameters, namely the number of categories (K) and the concentration parameters $\boldsymbol{\alpha} = \alpha_1,\ldots,\alpha_K$, where $\alpha_i > 0$. Furthermore, it uses the support $\tau_1,\ldots,\tau_K$ with $\tau_i \in (0,1)$ and $\sum_{i=1}^{K} \tau_i = 1$.

The probability density function is given by:

$$f(\tau_1,\ldots,\tau_K;\alpha_1,\ldots,\alpha_K) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^{K} \tau_i^{\alpha_i - 1},$$

in which the Beta function is given by

$$B(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha_i)}.$$

The gamma function is known by (for complex numbers with a positive real part):

$$\Gamma(\alpha_i) = \int_{0}^{\infty} x^{\alpha_i - 1} e^{-x} dx.$$

The mean of a random variable $\boldsymbol{X} = (X_1,\ldots,X_K) \sim Dir(\boldsymbol{\alpha})$ is given by $\mathbb{E}[X_i] = \frac{\alpha_i}{\sum_{k=1}^{K}\alpha_k}$. The mode is given by $\frac{\alpha_i - 1}{\sum_{k=1}^{K}\alpha_k - K}$.

It should also be noted that the Dirichlet distribution is a conjugate prior for the multinomial distribution. Which means that if the prior distribution of multinomially distributed parameters is Dirichlet then the posterior distribution is also a Dirichlet distribution.

A.3. Pareto distribution

The Pareto distribution is a distribution that is a power law probability distribution. The distribution has two parameters, namely η, the scale, and α, the shape. It must hold that $\eta > 0$ and $\alpha > 0$. Furthermore it uses the support $x \in [\eta, \infty)$.

The probability density function is given by

$$f(x;\alpha,\eta) = \begin{cases} \frac{\alpha\eta^\alpha}{x^{\alpha+1}} & if\ x \geq \eta \\ 0 & if\ x < \eta. \end{cases}$$

The conjugate prior of the Pareto distribution is the Gamma distribution.

A.4. Gamma distribution

The Gamma distribution is te conjugate prior for the Pareto distribution. It uses two parameters, namely $\alpha > 0$ as its shape and $\beta > 0$ as its rate parameter. The support is $x \in (0, \infty)$.

The probability density function is given by (for $\alpha, \beta, x > 0$)

$$f(x;\alpha,\beta) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta \cdot x}}{\Gamma(\alpha)}.$$

A.5. Bernoulli distribution

The Bernoulli distribution is the distribution in which a random variable takes value 1 with probability p and value 0 with probability $(1-p)$. The parameters of the distribution are $0 \leq p \leq 1$. and it uses the support $k \in \{0, 1\}$.

The probability density function is given by

$$f(k;p) = p^k (1-p)^{1-k}.$$

Note that the Bernoulli distribution is a special case of the Binomial distribution (with $n = 1$).

A.6. Beta distribution

The Beta distribution uses the parameters $\alpha > 0$ and $\beta > 0$, with support $x \in [0, 1]$.

The probability density function is given by

$$f(x;\alpha,\beta) = \frac{x^{\alpha-1} \cdot (1-x)^{\beta-1}}{B(\alpha,\beta)},$$

in which $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$.

The beta distribution is the conjugate prior for the Bernoulli distribution.

B

Mathematical details - model word frequencies

B.1. Generative model

A simple generative model for e-mails will be described. Let $\left(t_1, \ldots, t_J\right)$ be a given dictionary of words. Let $\boldsymbol{\theta}^y = \left(\theta_j^y\right)_{j=1}^J$ be relative frequencies for these words as they occur in e-mails, in which y indicates whether the e-mail is relevant ($y = 1$) or not relevant ($y = 0$).

Let $\boldsymbol{z} = (z_1, \ldots, z_n)$ be an ordered sequence of words. The probability distribution over e-mails $\boldsymbol{z}$ can be written as:

$$\begin{aligned} p(\boldsymbol{z}) &= \sum_{y=0,1} p(y)p(\boldsymbol{z}|y) \\ &= \sum_{y=0,1} p(y)p(n \mid y)p(\boldsymbol{z} \mid n, y) \\ &= \sum_{y=0,1} p(y)p(n \mid y)\prod_{i=1}^{n} p\left(z_i \mid n, y\right). \end{aligned}$$

For simplicity, assume that the distribution of n does not depend on y. With $p\left(z_i = t_j \mid n, y\right) = p\left(z_i \mid y\right) = \theta_j^y$, this fully specifies our generative model. The notation $x_j(\boldsymbol{z}) = \#\left\{i : z_i = t_j\right\}$ will be used, i.e. x_j is the word count of the j-th word in the dictionary. The following holds

$$p(\boldsymbol{z} \mid y) = p(n)\prod_{j=1}^{J}\left(\theta_j^y\right)^{x_j(\boldsymbol{z})}.$$

The posterior distribution over $\boldsymbol{x}$ will then become (taking into account multiple ways in which word counts may occur):

$$p(\boldsymbol{x} \mid y) = p(n)\binom{n}{x_1 \cdots x_J}\prod_{j=1}^{J}\left(\theta_j^y\right)^{x_j}. \tag{B.1}$$

B.1.1. Maximum likelihood estimation of θ^y

Let $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(N)}$ denote a sequence of e-mails (or for the purpose of this model, equivalently: word counts $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}$). Furthermore, let $y^{(1)}, \ldots, y^{(N)}$ denote whether an e-mail is relevant or not relevant. It is assumed that all e-mails are independently generated according to the model. Our available parameters are $\boldsymbol{\theta}^0$ and $\boldsymbol{\theta}^1$, and it is assumed that the probability distributions $p(y)$ and $p(n)$ to be fixed. The likelihood of this observation is

$$\begin{aligned} L\left(\boldsymbol{\theta}^0,\boldsymbol{\theta}^1\right) &= \prod_{m=1}^{N} p\left(\boldsymbol{x}^{(m)},y^{(m)}\right) \\ &= \prod_{m=1}^{N} p\left(y^{(m)}\right)p\left(\boldsymbol{x}^{(m)} \mid y^{(m)}\right) \\ &= \prod_{m=1}^{N} p\left(y^{(m)}\right)p\left(n^{(m)}\right)\binom{n^{(m)}}{x_1^{(m)}\cdots x_J^{(m)}}\prod_{j=1}^{J}\left(\theta_j^{y^{(m)}}\right)^{x_j^{(m)}}. \end{aligned}$$

The log likelihood will be, up to irrelevant additional constants, equal to

$$l\left(\boldsymbol{\theta}^0,\boldsymbol{\theta}^1\right) = \sum_{m=1}^{N}\sum_{j=1}^{J} x_j^{(m)}\log\left(\theta_j^{y^{(m)}}\right).$$

This log likelihood will be maximized subject to the constraint that $\sum_{j=1}^{J}\theta_j^y = 1$, and $\theta_j^y \geq 0$ for all j and y. This yields

$$\theta_j^y = \frac{\sum\limits_{m=1,\ldots,N:y^{(m)}=y} x_j^{(m)}}{\sum\limits_{m=1,\ldots,N:y^{(m)}=y} n^{(m)}}.$$

Classification based on MLE

It is known that y gets either the class 'relevant' or 'not relevant', therefore it can be assumed that the distribution of y is the Bernoulli distribution with parameter p (the probability of an e-mail being spam), i.e.,

$$p(y) = p^y\cdot(1-p)^{1-y}.$$

In this equation p is a fixed valued hyperparameter.

For the classification of a new e-mail, Bayes' law will be used and the obtained MLE estimators for $\boldsymbol{\theta}^y$ are plugged in:

$$\begin{aligned} p(y \mid \boldsymbol{z}) &= \frac{p(\boldsymbol{z} \mid y)p(y)}{p(\boldsymbol{z} \mid y=0)p(y=0)+p(\boldsymbol{z} \mid y=1)p(y=1)} && \text{(B.2)} \\ &\propto p(\boldsymbol{z} \mid y)p(y) && \text{(B.3)} \\ &\propto \prod_{j=1}^{J}(\theta_j^y)^{x_j(\boldsymbol{z})}\cdot p^y(1-p)^{1-y}, && \text{(B.4)} \end{aligned}$$

in which

$$\theta_j^y = \frac{\sum\limits_{m=1,\ldots,N:y^{(m)}=y} x_j^{(m)}}{\sum\limits_{m=1,\ldots,N:y^{(m)}=y} n^{(m)}}.$$

It is likely that with this formula a probability of 0 of both situations $y=0$ and $y=1$ will be given. This is the case since our dataset corresponding to the training of $\boldsymbol{\theta}$ does not have to have the situation in which every word occurs in an e-mail for both cases. Therefore Laplace smoothing will be applied to the MLE estimations of θ_j^y:

$$\tilde{\theta}_j^y = \frac{\sum\limits_{m=1,\ldots,N:y^{(m)}=y}\left(x_j^{(m)}\right)+1}{\sum\limits_{m=1,\ldots,N:y^{(m)}=y}\left(n^{(m)}\right)+J}. \tag{B.5}$$

B.1.2. Bayesian estimation

The posterior distribution of $\boldsymbol{\theta} = \left(\boldsymbol{\theta}^0, \boldsymbol{\theta}^1\right)$ will become

$$\begin{aligned} p\left(\boldsymbol{\theta} \mid y, \boldsymbol{x}\right) &= \frac{p(\boldsymbol{x} \mid \boldsymbol{\theta}, y) p(y \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(y \mid \boldsymbol{x}) p(\boldsymbol{x})} \\ &= \frac{p(\boldsymbol{x} \mid \boldsymbol{\theta}, y) p(y) p(\boldsymbol{\theta})}{p(y \mid \boldsymbol{x}) p(\boldsymbol{x})} \\ &\propto p(\boldsymbol{x} \mid \boldsymbol{\theta}, y) p(\boldsymbol{\theta}), \end{aligned}$$

in which it is used that

$$p(\boldsymbol{x} \mid \boldsymbol{\theta}, y) = p(n) \binom{n}{x_1 \cdots x_J} \prod_{j=1}^{J} \left(\theta_j^y\right)^{x_j},$$

and $p\left(\boldsymbol{\theta}^y\right) \sim Dir(\boldsymbol{\alpha})$, i.e.

$$p\left(\boldsymbol{\theta}^y\right) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{j=1}^{J} \left(\theta_j^y\right)^{\alpha_j - 1}.$$

It is generally known that

$$B(\boldsymbol{\alpha}) = \frac{\prod_{j=1}^{J} \Gamma\left(\alpha_j\right)}{\Gamma\left(\sum_{j=1}^{J}, \alpha_j\right)} \tag{B.6}$$

and for some $\beta > 0$

$$\Gamma(\beta) = (\beta - 1)! \tag{B.7}$$

This gives us that

$$p\left(\boldsymbol{\theta}^0, \boldsymbol{\theta}^1 \mid y, \boldsymbol{x}\right) \propto \prod_{j=1}^{J} \left(\theta_j^y\right)^{x_j + \alpha_j - 1} \cdot \prod_{j=1}^{J} \left(\theta_j^{1-y}\right)^{x_j + \alpha_j - 1}.$$

Let $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(N)}$ denote a sequence of e-mails (or for the purpose of this model, equivalently: word counts $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}$). Furthermore, let $y^{(1)}, \ldots, y^{(N)}$ denote whether an e-mail is relevant or not relevant. It is assumed that all e-mails are independently generated according to the model. Our available parameters are $\boldsymbol{\theta}^0$ and $\boldsymbol{\theta}^1$, and it is assumed that the probability distributions $p(y)$ and $p(n)$ to be fixed. This gives us the following posterior distribution for $\boldsymbol{\theta}^y$:

$$\begin{aligned} p\left(\boldsymbol{\theta}^\xi \mid \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}, y^{(1)}, \ldots, y^{(N)}\right) &\propto \prod_{m=1,\ldots,N: y^{(m)} = \xi} p\left(\boldsymbol{x}^{(m)} \mid \boldsymbol{\theta}, y^{(m)}\right) p(\boldsymbol{\theta}) \\ &\propto \prod_{m=1,\ldots,N: y^{(m)} = \xi} \left(\prod_{j=1}^{J} \left(\theta_j^{y^{(m)}}\right)^{x_j^{(m)}}\right) \cdot \frac{1}{B(\boldsymbol{\alpha})} \prod_{j=1}^{J} \left(\theta_j^\xi\right)^{\alpha_j - 1} \\ &= \frac{1}{B(\boldsymbol{\alpha})} \prod_{j=1}^{J} \left(\theta_j^\xi\right)^{\sum_{m=1,\ldots,N: y^{(m)} = \xi} \left(x_j^{(m)}\right) + \alpha_j - 1} \\ &\sim Dir\left(\boldsymbol{\alpha}^\xi\right), \end{aligned}$$

with $\alpha_j^\xi = \sum_{m=1,\ldots,N: y^{(m)} = \xi} \left(x_j^{(m)}\right) + \alpha_j$ and $\xi = 0, 1$.

It is known that the maximum a posteriori (MAP) estimate equals the mode of the posterior distribution. Using the conclusion that the posterior distribution of $\boldsymbol{\theta}^\xi$ equals the Dirichlet distribution with parameter $\boldsymbol{\alpha}^\xi$, it follows that the MAP estimate for $\boldsymbol{\theta}^\xi$ is given by

$$\theta_j^\xi = \frac{\alpha_j^\xi - 1}{\sum_{j=1}^{J} \left(\alpha_j^\xi\right) - J} = \frac{\sum_{m=1,\ldots,N: y^{(m)} = \xi} \left(x_j^{(m)}\right) + \alpha_j - 1}{\sum_{j=1}^{J} \left(\sum_{m=1,\ldots,N: y^{(m)} = \xi} \left(x_j^{(m)}\right) + \alpha_j\right) - J},$$

with the restriction that $\alpha_j^\xi > 1$. If $\alpha_j^\xi \leq 1$ then the maximum will be on the boundary.

The posterior mean is given by

$$\mathbb{E}\theta_j^\xi = \frac{\alpha_j^\xi}{\sum\limits_{j=1}^{J} \alpha_j^\xi} = \frac{\sum\limits_{m=1,\ldots,N:y^{(m)}=\xi} \left(x_j^{(m)}\right) + \alpha_j}{\sum\limits_{j=1}^{J} \left(\sum\limits_{m=1,\ldots,N:y^{(m)}=\xi} \left(x_j^{(m)}\right) + \alpha_j \right)}.$$

Classification based on Bayesian estimate

It is known that y gets either the class 'relevant' or 'not relevant', therefore it can be assumed that the distribution of y is the Bernoulli distribution with parameter p (the probability of an e-mail being fraud), i.e.,

$$p(y) = p^y \cdot (1-p)^{1-y}.$$

In this equation p is a fixed valued hyperparameter.

Using Bayes' law and the formula (with conditioning on $\boldsymbol{\theta}$) $p(y \mid \boldsymbol{z}) = \frac{p(\boldsymbol{z}|y)p(y)}{p(\boldsymbol{z}|y=0)p(y=0)+p(\boldsymbol{z}|y=1)p(y=1)}$ of Section B.1.1:

$$\begin{aligned} p(y \mid \boldsymbol{x}, \boldsymbol{\theta}) &\propto p(\boldsymbol{x} \mid y, \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid y) p(y) \\ &\propto p(\boldsymbol{x} \mid y, \boldsymbol{\theta}) p(y) \\ &\propto \prod_{j=1}^{J} \left(\theta_j^y\right)^{x_j} \cdot p^y (1-p)^{1-y}. \end{aligned}$$

This gives us that the formula for classification equals:

$$p(y \mid \boldsymbol{x}) = \int_{S^J} \prod_{j=1}^{J} \left(\theta_j^y\right)^{x_j} \cdot p^y (1-p)^{1-y} d\boldsymbol{\theta},$$

in which S^J is the corresponding simplex over the J dimensions of $\boldsymbol{\theta}$.

Therefore, when taking into account the sequence of e-mails $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(N)}$ (or for our purposes, equivalently: word counts $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}$) together with information $y^{(1)}, \ldots, y^{(N)}$, the conditional distribution of $y^{(N+1)}$ for our new e-mail can be written as:

$$\begin{aligned} p\left(y^{(N+1)} \mid \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}, y^{(1)}, \ldots, y^{(N)}, \boldsymbol{x}^{(N+1)}\right) &= \int_{S^J} p\left(y^{(N+1)} \mid \boldsymbol{x}^{(N+1)}, \boldsymbol{\theta}\right) p\left(\boldsymbol{\theta} \mid \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}, y^{(1)}, \ldots, y^{(N)}\right) d\boldsymbol{\theta} \\ &\propto \int_{S^J} \prod_{j=1}^{J} \left(\theta_j^{y^{(N+1)}}\right)^{x_j^{(N+1)}} \cdot p^{y^{(N+1)}} (1-p)^{1-y^{(N+1)}} \cdot \\ &\quad \frac{1}{B\left(\boldsymbol{\alpha}^{y^{(N+1)}}\right)} \prod_{j=1}^{J} \left(\theta_j^{y^{(N+1)}}\right)^{\sum_{m=1,\ldots,N:y^{(m)}=y^{(N+1)}} \left(x_j^{(m)}\right)+\alpha_j-1} d\boldsymbol{\theta} \\ &= p^{y^{(N+1)}} (1-p)^{1-y^{(N+1)}} \cdot \frac{1}{B\left(\boldsymbol{\alpha}^{y^{(N+1)}}\right)} \cdot \\ &\quad \int_{S^J} \prod_{j=1}^{J} \left(\theta_j^{y^{(N+1)}}\right)^{x_j^{(N+1)} + \sum_{m=1,\ldots,N:y^{(m)}=y^{(N+1)}} \left(x_j^{(m)}\right)+\alpha_j-1} d\boldsymbol{\theta} \\ &= p^{y^{(N+1)}} (1-p)^{1-y^{(N+1)}} \cdot \frac{1}{B\left(\boldsymbol{\alpha}^{y^{(N+1)}}\right)} \cdot B\left(\tilde{\boldsymbol{\alpha}}^{y^{(N+1)}}\right), \end{aligned}$$

using $\int_{S^J} \frac{1}{B\left(\tilde{\boldsymbol{\alpha}}^{y^{(N+1)}}\right)} \prod_{j=1}^{J} \left(\theta_j^{y^{(N+1)}}\right)^{\tilde{\boldsymbol{\alpha}}^{y^{(N+1)}}-1} d\boldsymbol{\theta} = 1$ (inside the integral is a typical Dirichlet distribution with parameters J and $\tilde{\boldsymbol{\alpha}}^{y^{(N+1)}}$, so the probability density function integrates to 1) and with

$$\tilde{\alpha}_j^{y^{(N+1)}} = x_j^{(N+1)} + \sum_{m=1,\ldots,N:y^{(m)}=y^{(N+1)}} \left(x_j^{(m)}\right) + \alpha_j.$$

Which needs to be calculated for $y^{(N+1)} = 0$ and $y^{(N+1)} = 1$ since this expression holds up to the proportionality constant.

B.2. Discriminative model

A simple discriminative model for e-mails will be described. As in the generative model, let $(t_1, \dots, t_J)$ be a given dictionary of words. Let y indicate whether the e-mail is relevant $(y = 1)$ or not relevant $(y = 0)$.

Let $\boldsymbol{z} = (z_1, \dots, z_n)$ be an ordered sequence of words. Also the notation $x_j(z) = \#\{i : z_i = t_j\}$ will be used, i.e. x_j is the word count of the j-th word in the dictionary.

Logistic Regression will be used as discriminative model. Therefore it is assumed that

$$p(y=1|\boldsymbol{x}) = \frac{1}{1+\exp\left(w_0 + \sum_{j=1}^{J} w_j x_j\right)}; \tag{B.8}$$

$$p(y=0|\boldsymbol{x}) = \frac{\exp\left(w_0 + \sum_{j=1}^{J} w_j x_j\right)}{1+\exp\left(w_0 + \sum_{j=1}^{J} w_j x_j\right)}. \tag{B.9}$$

B.2.1. Maximum likelihood estimation of w

Let $\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(N)}$ denote a sequence of e-mails (or for the purpose of this model, equivalently: word counts $\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(N)}$). Furthermore, let $y^{(1)}, \dots, y^{(N)}$ denote whether an e-mail is relevant or not relevant. All e-mails are assumed to be independently generated according to the model. Let $\boldsymbol{w} = (w_1, \dots, w_J)$ be the weights needed in Formula (5.8) and (5.7). With the assumption that the outcomes are Bernoulli distributed, this will get us the following likelihood (log likelihood denoted with l, and likelihood with L):

$$\begin{aligned} L(\boldsymbol{w}) &= \prod_{m=1}^{N} p\left(y^{(m)} \mid \boldsymbol{x}^{(m)}\right) & \text{(B.10)} \\ &= \prod_{m=1}^{N} p\left(y=1 \mid \boldsymbol{x}^{(m)}\right)^{y^{(m)}} \cdot p\left(y=0 \mid \boldsymbol{x}^{(m)}\right)^{1-y^{(m)}}. & \text{(B.11)} \end{aligned}$$

$$l(\boldsymbol{w}) = \sum_{m=1}^{N} \left(y^{(m)} \ln\left(p(y=1 \mid \boldsymbol{x}^{(m)})\right) + \left(1-y^{(m)}\right) \ln\left(p\left(y=0 \mid \boldsymbol{x}^{(m)}\right)\right)\right). \tag{B.12}$$

The only steps left to get the best parameters $\boldsymbol{w}$ is to maximize Equation (B.12) (in other words setting the derivative w.r.t. $\boldsymbol{w}$ to zero and solving this equation). Note that no closed form of the maximization of the log likelihood is available. Various algorithms are available to approximate the maximization, in the next section the used approximation method will be stated.

Approximation method

For the MLE of $\boldsymbol{w}$ the available function LogisticRegression from the sklearn.linear_model package has been used. The approximation method selected in this function is SAGA, other available approximation methods in LogisticRegression are:

- liblinear
- lbfgs
- newton-cg
- SAG

Of these approximation methods the first three have the major disadvantage that they are not faster for larger datasets. SAGA is a incremental gradient algorithm with fast linear convergence rates, and based on SAG. It has been shown that SAGA is one of the methods that is most efficient for high dimensional data, as is the case in our application. The reason to choose SAGA over SAG is because SAGA does not require a predefined number of iterations and because SAGA is the solver of choice for sparse multinomial logistic regression according to the description of the package. A detailed description of SAGA is given by Defazio et al. [13].

Classification based on MLE

For the classification of a new e-mail, the Equations (5.8) and (5.7) are used. The classification property of logistic regression comes down to giving e-mail $\boldsymbol{z}^{(N+1)}$ (and corresponding data $\boldsymbol{x}^{(N+1)}$) label $y^{(N+1)} = 0$ if

$$1 < \frac{p\left(y=0|x^{(N+1)}\right)}{p\left(y=1|x^{(N+1)}\right)} = \exp\left(w_0 + \sum_{j=1}^{J} w_j x_j^{(N+1)}\right).$$

Rewriting gives that $\boldsymbol{z}^{(N+1)}$ (with corresponding word counts $\boldsymbol{x}^{(N+1)}$) is classified with label $y = 0$ if $0 < w_0 + \sum_{j=1}^{J} w_j x_j^{(N+1)}$, and with label $y = 1$ otherwise.

B.2.2. Bayesian estimation

The posterior distribution of $\boldsymbol{w}$ will become:

$$p(\boldsymbol{w} \mid y, \boldsymbol{x}) \propto p(y \mid \boldsymbol{x}, \boldsymbol{w}) p(\boldsymbol{w})$$

in which Equations (5.8) and (5.7) are used. For clarification purposes, these will be stated again:

$$p(y=1|\boldsymbol{x},\boldsymbol{w}) = \frac{1}{1+\exp\left(w_0 + \sum_{j=1}^{J} w_j x_j\right)},$$

$$p(y=0|\boldsymbol{x},\boldsymbol{w}) = \frac{\exp\left(w_0 + \sum_{j=1}^{J} w_j x_j\right)}{1+\exp\left(w_0 + \sum_{j=1}^{J} w_j x_j\right)}.$$

Furthermore, use that $p(\boldsymbol{w}) \sim N(0, 10^6)$, with each w_i independent of the other weights [45, 53].

This gives us that

$$p(\boldsymbol{w} \mid y, \boldsymbol{x}) \propto \left(\frac{1}{1+\exp\left(w_0 + \sum_{j=1}^{J} w_j x_j\right)} \cdot y + \frac{\exp\left(w_0 + \sum_{j=1}^{J} w_j x_j\right)}{1+\exp\left(w_0 + \sum_{j=1}^{J} w_j x_j\right)} \cdot (1-y)\right) \cdot \prod_{i=0}^{J} \frac{1}{\sqrt{2\pi \cdot 10^{12}}} \exp\left(-\frac{w_i^2}{2 \cdot 10^{12}}\right).$$

Let $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(N)}$ denote a sequence of e-mails (or for the purpose of this model, equivalently: word counts $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}$). Furthermore, let $y^{(1)}, \ldots, y^{(N)}$ denote whether an e-mail is relevant or not relevant. All e-mails are assumed to be independently generated according to the model. This gives us the following posterior distribution for $\boldsymbol{w}$:

$$p\left(\boldsymbol{w} \mid \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}, y^{(1)}, \ldots, y^{(N)}\right)$$

$$\propto \prod_{m=1}^{N} \left(\frac{1}{1+\exp\left(w_0 + \sum_{j=1}^{J} w_j x_j^{(m)}\right)} \cdot y^{(m)} + \frac{\exp\left(w_0 + \sum_{j=1}^{J} w_j x_j^{(m)}\right)}{1+\exp\left(w_0 + \sum_{j=1}^{J} w_j x_j^{(m)}\right)} \cdot \left(1 - y^{(m)}\right)\right) \cdot \prod_{i=0}^{J} \frac{1}{\sqrt{2\pi \cdot 10^{12}}} \exp\left(-\frac{w_i^2}{2 \cdot 10^{12}}\right).$$

Classification based on Bayesian estimate

Using Bayes' law it follows that the formula for classification equals:

$$p(y \mid \boldsymbol{x}) \propto \int_{S^J} p(\boldsymbol{w} \mid y, \boldsymbol{x}) p(y \mid \boldsymbol{x}, \boldsymbol{w}) d\boldsymbol{w}$$

Therefore, when taking into account the sequence of e-mails $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(N)}$ (or for our purposes, equivalently: word counts $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}$) together with information $y^{(1)}, \ldots, y^{(N)}$, the conditional distribution of $y^{(N+1)}$ for our new e-mail can be written as:

$$p\left(y^{(N+1)} \mid \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N+1)}, y^{(1)}, \ldots, y^{(N)}\right)$$

$$\begin{aligned}
\propto \quad & \int_{S^J} p\left(y^{(N+1)}, \boldsymbol{w} \mid \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}, y^{(1)}, \ldots, y^{(N)}, \boldsymbol{x}^{(N+1)}\right) d\boldsymbol{w} \\
= \quad & \int_{S^J} p\left(y^{(N+1)} \mid \boldsymbol{x}^{(N+1)}, \boldsymbol{w}\right) p\left(\boldsymbol{w} \mid \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}, y^{(1)}, \ldots, y^{(N)}\right) d\boldsymbol{w} \\
= \quad & \int_{S^J} \left(\frac{1}{1 + \exp\left(w_0 + \sum_{j=1}^{J} w_j x_j^{(N+1)}\right)} \cdot y^{(N+1)} + \frac{\exp\left(w_0 + \sum_{j=1}^{J} w_j x_j^{(N+1)}\right)}{1 + \exp\left(w_0 + \sum_{j=1}^{J} w_j x_j^{(N+1)}\right)} \cdot \left(1 - y^{(N+1)}\right) \right) \\
& \cdot \prod_{m=1}^{N} \left(\frac{1}{1 + \exp\left(w_0 + \sum_{j=1}^{J} w_j x_j^{(m)}\right)} \cdot y^{(m)} + \frac{\exp\left(w_0 + \sum_{j=1}^{J} w_j x_j^{(m)}\right)}{1 + \exp\left(w_0 + \sum_{j=1}^{J} w_j x_j^{(m)}\right)} \cdot \left(1 - y^{(m)}\right) \right) \cdot \prod_{i=0}^{J} \frac{1}{\sqrt{2\pi \cdot 10^{12}}} \exp\left(-\frac{w_i^2}{2 \cdot 10^{12}}\right) d\boldsymbol{w}
\end{aligned}$$

Which needs to be calculated for $y^{(N+1)} = 0$ and $y^{(N+1)} = 1$ since this expression holds up to the proportionality constant.

For the Bayesian estimation of $\boldsymbol{w}$ the available package Pystan has been used. The approximation method available and used in this this package is NUTS. A detailed description of NUTS is given by Hoffman and Gelman [30].

B.3. Relation generative and discriminative model

When considering the following assumptions it will be shown that the generative model (in our case Multinomial Naive Bayes) implies the form of the discriminative model (in our case Logistic Regression) [41]:

- $\boldsymbol{X} = (X_1, \ldots, X_n)$, with X_i a discrete random variable, in our case the counts of words in an e-mail
- It holds that $p(\boldsymbol{x}|y)$ follows the distribution in Equation (5.1), described in Section B.1
- X_i and X_j are conditionally independent given Y, for each $i \neq j$

Using these assumptions, and Bayes rule, it holds that:

$$\begin{aligned}
\mathbb{P}(y = 1|\boldsymbol{x}) &= \frac{p(y=1)p(\boldsymbol{x}|y=1)}{p(y=1)p(\boldsymbol{x}|y=1) + p(y=0)p(\boldsymbol{x}|y=0)} \\
&= \frac{1}{1 + \frac{p(y=0)p(\boldsymbol{x}|y=0)}{p(y=1)p(\boldsymbol{x}|y=1)}} \\
&= \frac{1}{1 + \exp(\ln \frac{p(y=0)p(\boldsymbol{x}|y=0)}{p(y=1)p(\boldsymbol{x}|y=1)})} \\
&= \frac{1}{1 + \exp(\ln \frac{p(y=0)}{p(y=1)} + \ln \frac{p(\boldsymbol{x}|y=0)}{p(\boldsymbol{x}|y=1)})}
\end{aligned}$$

With the assumption that $p(\boldsymbol{x}|y)$ follows the distribution in Equation (5.1), it follows that:

$$\begin{aligned}
\ln\frac{p(\boldsymbol{x}|y=0)}{p(\boldsymbol{x}|y=1)} &= \ln\frac{p(n)\binom{n}{x_1\cdots x_J}\prod_{j=1}^{J}(\theta_j^0)^{x_j}}{p(n)\binom{n}{x_1\cdots x_J}\prod_{j=1}^{J}(\theta_j^1)^{x_j}} \\
&= \ln\frac{\prod_{j=1}^{J}(\theta_j^0)^{x_j}}{\prod_{j=1}^{J}(\theta_j^1)^{x_j}} \\
&= \ln\left(\prod_{j=1}^{J}(\theta_j^0)^{x_j}\right)-\ln\left(\prod_{j=1}^{J}(\theta_j^1)^{x_j}\right) \\
&= x_j\ln\left(\prod_{j=1}^{J}(\theta_j^0)\right)-x_j\ln\left(\prod_{j=1}^{J}(\theta_j^1)\right) \\
&= x_j\left(\ln\left(\prod_{j=1}^{J}(\theta_j^0)\right)-\ln\left(\prod_{j=1}^{J}(\theta_j^1)\right)\right) \\
&= x_j\left(\sum_{j=1}^{J}\ln\left((\theta_j^0)\right)-\ln\left((\theta_j^1)\right)\right) \\
&= x_j\left(\sum_{j=1}^{J}\ln\left(\frac{\theta_j^0}{\theta_j^1}\right)\right) \\
&= \sum_{j=1}^{J}x_jP_j
\end{aligned}$$

In which $P_j = \ln\left(\frac{\theta_j^0}{\theta_j^1}\right)$.

Taking these two expressions together gives us:

$$p(y=1|\boldsymbol{x}) = \frac{1}{1+\exp(\ln\frac{p(y=0)}{p(y=1)}+\sum_{j=1}^{J}x_jP_j)} \tag{B.13}$$

Which is of the form corresponding to Logistic Regression, i.e.

$$p(y=1|\boldsymbol{x}) = \frac{1}{1+\exp(w_0+\sum_{j=1}^{J}w_jx_j)}$$

and

$$p(y=1|\boldsymbol{x}) = 1-p(y=0|\boldsymbol{x}) = \frac{\exp(w_0+\sum_{j=1}^{J}w_jx_j)}{1+\exp(w_0+\sum_{j=1}^{J}w_jx_j)}$$

with $w_0 = \ln\frac{p(y=0)}{p(y=1)}$ and $w_j = P_j$ for $j = 1,\ldots,J$.

Based on this proof it can be concluded that the implemented generative model is a special case of the discriminative model. The discriminative model is therefore more general. However, as will be seen in for example Section 6.2 that the generative model is computationally less expensive.

C

Mathematical details - model word occurrences

C.1. Generative model

A simple generative model for e-mails will be described. Let $\boldsymbol{z} = (z_1, \ldots, z_n)$ be a sequence of words of an e-mail. Let $(t_1, \ldots, t_J)$ be a given dictionary of words. Let $\boldsymbol{q}^y = (q_j^y)_{j=1}^J$, in which q_j^y represents the probability that word t_j corresponds to relevant $(y = 1)$ or not relevant $(y = 0)$, i.e. $p(t_i \mid y) = q_j^y$. Furthermore, it holds that $q_j^1 = 1 - q_j^0$.

Let $\boldsymbol{x} = (x_1, \ldots, x_J)$ be the corresponding features of the given dictionary, in which $x_i = 1$ if word t_i is present in the e-mail (i.e. $t_i \in \boldsymbol{z}$) and $x_i = 0$ if word t_i is not present (i.e. $t_i \notin \boldsymbol{z}$). The probability distribution over e-mail features $\boldsymbol{x}$ can be written as:

$$\begin{aligned} p(\boldsymbol{x}) &= \sum_{y=0,1} p(y) p(\boldsymbol{x}|y) \\ &= \sum_{y=0,1} p(y) p(n \mid y) p(\boldsymbol{x} \mid n, y) \\ &= \sum_{y=0,1} p(y) p(n \mid y) \prod_{i=1}^{J} p(x_i \mid n, y). \end{aligned}$$

For simplicity, it will be assumed that the distribution of n does not depend on y. Once it is noted that $p(x_i \mid n, y) = p(x_i \mid y) = p(t_i \mid y)^{x_i} \cdot (1 - p(t_i \mid y))^{1-x_i}$, and recall that $p(t_i \mid y) = q_i^y$ this fully specifies our generative model.

The posterior distribution over $\boldsymbol{x}$ will then become:

$$p(\boldsymbol{x} \mid y) = p(n) \prod_{j=1}^{J} (q_j^y)^{x_j} \cdot (1 - q_j^y)^{1-x_j} \tag{C.1}$$

C.1.1. Maximum likelihood estimation for θ^y

Let $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}$ denote word occurrences of e-mails. Furthermore, let $y^{(1)}, \ldots, y^{(N)}$ denote whether an e-mail is relevant or not relevant. All e-mails are assumed to be independently generated according to the model. Our available parameters are $\boldsymbol{q}^0$ and $\boldsymbol{q}^1$, and the probability distributions $p(y)$ and $p(n)$ are assumed to be fixed. The likelihood of this observation is

$$\begin{aligned} L(\boldsymbol{q}^0,\boldsymbol{q}^1) &= \prod_{m=1}^{N} p(\boldsymbol{x}^{(m)}, y^{(m)}) \\ &= \prod_{m=1}^{N} p(y^{(m)}) p(\boldsymbol{x}^{(m)} \mid y^{(m)}) \\ &= \prod_{m=1}^{N} p(y^{(m)}) p(n^{(m)}) \prod_{j=1}^{J} (q_j^{y^{(m)}})^{x_j^{(m)}} \cdot (1-q_j^{y^{(m)}})^{1-x_j^{(m)}}. \end{aligned}$$

The log likelihood will be, up to irrelevant additional constants, equal to

$$l(\boldsymbol{q}^0,\boldsymbol{q}^1) = \sum_{m=1}^{N}\sum_{j=1}^{J} x_j^{(m)} \log(q_j^{y^{(m)}}) + (1-x_j^{(m)})\log(1-q_j^{y^{(m)}}).$$

This log likelihood will be maximized subject to the constraint that $q_j^y \geq 0$ for all j and y. This yields

$$q_j^y = \frac{\sum\limits_{m=1,\dots,N:y^{(m)}=y} x_j^{(m)}}{\sum\limits_{m=1,\dots,N:y^{(m)}=y} 1}.$$

Classification based on MLE

It is known that y gets either the class 'relevant' or 'not relevant', therefore it can be assumed that the distribution of y is the Bernoulli distribution with parameter p (the probability of an e-mail being relevant), i.e.,

$$p(y) = p^y \cdot (1-p)^{1-y}.$$

In this equation p is a fixed valued hyperparameter.

For the classification of a new e-mail, Bayes' law is used and the obtained MLE estimators for $\boldsymbol{\theta}^y$ are plugged in:

$$\begin{aligned} p(y \mid \boldsymbol{x}) &= \frac{p(\boldsymbol{x} \mid y)p(y)}{p(\boldsymbol{x} \mid y=0)p(y=0) + p(\boldsymbol{x} \mid y=1)p(y=1)} && \text{(C.2)} \\ &\propto p(\boldsymbol{x} \mid y)p(y) && \text{(C.3)} \\ &\propto \prod_{j=1}^{J} (q_j^y)^{x_j} \cdot (1-q_j^y)^{1-x_j} \cdot p^y(1-p)^{1-y}, && \text{(C.4)} \end{aligned}$$

in which

$$q_j^y = \frac{\sum\limits_{m=1,\dots,N:y^{(m)}=y} x_j^{(m)}}{\sum\limits_{m=1,\dots,N:y^{(m)}=y} 1}.$$

It is likely that with this formula a probability of 0 of both situations $y=0$ and $y=1$ will be given. This is the case since our dataset corresponding to the training of $\boldsymbol{q}$ does not have to have the situation in which every word occurs in an e-mail for both cases. Therefore Laplace smoothing will be applied to the MLE estimations of q_j^y:

$$\tilde{\theta}_j^y = \frac{\sum\limits_{m=1,\dots,N:y^{(m)}=y} x_j^{(m)} + 1}{\sum\limits_{m=1,\dots,N:y^{(m)}=y} 1 + J}. \tag{C.5}$$

C.1.2. Bayesian estimation

The posterior distribution of $\boldsymbol{q} = (\boldsymbol{q}^0, \boldsymbol{q}^1)$ will become

$$\begin{aligned} p(\boldsymbol{q} \mid y, \boldsymbol{x}) &= \frac{p(\boldsymbol{x} \mid \boldsymbol{q}, y)p(y \mid \boldsymbol{q})p(\boldsymbol{q})}{p(y \mid \boldsymbol{x})p(\boldsymbol{x})} \\ &= \frac{p(\boldsymbol{x} \mid \boldsymbol{q}, y)p(y)p(\boldsymbol{q})}{p(y \mid \boldsymbol{x})p(\boldsymbol{x})} \\ &\propto p(\boldsymbol{x} \mid \boldsymbol{q}, y)p(\boldsymbol{q}), \end{aligned}$$

in which the following equations are used

$$p(\boldsymbol{x} \mid \boldsymbol{q}, y) = p(n) \prod_{j=1}^{J} (q_j^y)^{x_j} \cdot (1 - q_j^y)^{1-x_j}$$

and $p(q_j^y) \sim Beta(\alpha_j, \beta_j)$, i.e.

$$p(q_j^y) = \frac{(q_j^y)^{\alpha_j - 1} \cdot (1 - q_j^y)^{\beta_j - 1}}{B(\alpha_j, \beta_j)}. \tag{C.6}$$

It is generally known that

$$B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha + \beta)} \tag{C.7}$$

and for some $\zeta > 0$

$$\Gamma(\zeta) = (\zeta - 1)! \tag{C.8}$$

This gives us that

$$p(\boldsymbol{q}^y) = \prod_{j=1}^{J} \frac{(q_j^y)^{\alpha_j - 1} \cdot (1 - q_j^y)^{\beta_j - 1}}{B(\alpha_j, \beta_j)}. \tag{C.9}$$

Combining Equation (C.1) and (C.6), gives us

$$p(\boldsymbol{q}^0, \boldsymbol{q}^1 \mid y, \boldsymbol{x}) \propto \prod_{j=1}^{J} \frac{(q_j^y)^{\alpha_j + x_j - 1} \cdot (1 - q_j^y)^{\beta_j - x_j}}{B(\alpha_j, \beta_j)} \cdot \prod_{j=1}^{J} \frac{(q_j^{1-y})^{\alpha_j + x_j - 1} \cdot (1 - q_j^{1-y})^{\beta_j - x_j}}{B(\alpha_j, \beta_j)}$$

Let $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)})$ denote word occurrences of e-mails. Furthermore, let $y^{(1)}, \ldots, y^{(N)}$ denote whether an e-mail is relevant or not. All e-mails are assumed to be independently generated according to the model. Our available parameters are $\boldsymbol{q}^0$ and $\boldsymbol{q}^1$, and the probability distributions $p(y)$ and $p(n)$ are assumed to be fixed. This gives us the following posterior distribution for $\boldsymbol{q}^y$:

$$\begin{aligned}
p(\boldsymbol{q}^\xi \mid \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}, y^{(1)}, \ldots, y^{(N)}) \quad &\propto \quad \prod_{m=1,\ldots,N: y^{(m)} = \xi} p(\boldsymbol{x}^{(m)} \mid \boldsymbol{q}, y^{(m)}) p(\boldsymbol{q}) \\
&\propto \quad \prod_{m=1,\ldots,N: y^{(m)} = \xi} \left(\prod_{j=1}^{J} (q_j^{y^{(m)}})^{x_j^{(m)}} \cdot (1 - q_j^{y^{(m)}})^{1 - x_j^{(m)}} \right) \cdot \prod_{j=1}^{J} \frac{(q_j^{y^{(m)}})^{\alpha_j - 1} \cdot (1 - q_j^{y^{(m)}})^{\beta_j - 1}}{B(\alpha_j, \beta_j)} \\
&= \quad \prod_{j=1}^{J} \frac{(q_j^\xi)^{\alpha_j - 1 + \sum_{m=1,\ldots,N: y^{(m)} = \xi} x_j^{(m)}} \cdot (1 - q_j^\xi)^{\beta_j - 1 + \sum_{m=1,\ldots,N: y^{(m)} = \xi} (1 - x_j^{(m)})}}{B(\alpha_j, \beta_j)},
\end{aligned}$$

with $\xi = 0, 1$.

Classification based on Bayesian estimate

It is known that y gets either the class 'relevant' or 'not relevant', therefore it will be assumed that the distribution of y is the Bernoulli distribution with parameter p (the probability of an e-mail being spam), i.e.,

$$p(y) = p^y \cdot (1 - p)^{1-y}.$$

In this equation p is a fixed valued hyperparameter.

Using Bayes' law and the formula (with conditioning on $\boldsymbol{q}$) $p(y \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x}|y)p(y)}{p(\boldsymbol{x}|y=0)p(y=0) + p(\boldsymbol{x}|y=1)p(y=1)}$ of Section B.1.1:

$$\begin{aligned}
p(y \mid \boldsymbol{x}, \boldsymbol{q}) \quad &\propto \quad p(\boldsymbol{x} \mid y, \boldsymbol{q}) p(\boldsymbol{q} \mid y) p(y) \\
&\propto \quad p(\boldsymbol{x} \mid y, \boldsymbol{q}) p(y) \\
&\propto \quad \prod_{j=1}^{J} (q_j^y)^{x_j} \cdot (1 - q_j^y)^{1 - x_j} \cdot p^y (1 - p)^{1-y}.
\end{aligned}$$

This gives us that the formula for classification equals:

$$p(y \mid \boldsymbol{x}) = \int_{S^J} \prod_{j=1}^{J} (q_j^y)^{x_j} \cdot (1-q_j^y)^{1-x_j} \cdot p^y (1-p)^{1-y} d\boldsymbol{q},$$

in which S^J is the corresponding simplex over the J dimensions of $\boldsymbol{q}$.

Therefore, when taking into account the sequence of word occurrences $\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(N)})$ together with information $y^{(1)},\dots,y^{(N)}$, the conditional distribution of $y^{(N+1)}$ for our new e-mail will be written as:

$$p(y^{(N+1)} \mid \boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(N)}, y^{(1)},\dots,y^{(N)}, \boldsymbol{x}^{(N+1)})$$

$$\begin{aligned}
&\propto \int_{S^J} p(y^{(N+1)} \mid \boldsymbol{x}^{(N+1)}, \boldsymbol{q}) p(\boldsymbol{q} \mid \boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(N)}, y^{(1)},\dots,y^{(N)}) d\boldsymbol{q} \\
&\propto \int_{S^J} \prod_{j=1}^{J} (q_j^{y^{(N+1)}})^{x_j^{(N+1)}} \cdot (1-q_j^{y^{(N+1)}})^{1-x_j^{(N+1)}} \cdot p^{y^{(N+1)}} (1-p)^{1-y^{(N+1)}} \cdot \\
&\quad \prod_{j=1}^{J} \frac{(q_j^{y^{(N+1)}})^{\alpha_j - 1 + \sum_{m=1,\dots,N: y^{(m)}=y^{(N+1)}} x_j^{(m)}} (1-q_j^{y^{(N+1)}})^{\beta_j - 1 + \sum_{m=1,\dots,N: y^{(m)}=y^{(N+1)}} (1-x_j^{(m)})}}{B(\alpha_j, \beta_j)} d\boldsymbol{q} \\
&= \prod_{j=1}^{J} \frac{p^{y^{(N+1)}} (1-p)^{1-y^{(N+1)}}}{B(\alpha_j, \beta_j)} \cdot \int_0^1 \left(q_j^{y^{(N+1)}}\right)^{\tilde{\alpha}_j - 1} \cdot \left(1-q_j^{y^{(N+1)}}\right)^{\tilde{\beta}_j - 1} dq_j^{y^{(N+1)}} \\
&= \prod_{j=1}^{J} \frac{p^{y^{(N+1)}} (1-p)^{1-y^{(N+1)}}}{B(\alpha_j, \beta_j)} \cdot B(\tilde{\alpha}_j, \tilde{\beta}_j)
\end{aligned}$$

in which $\tilde{\alpha}_j = \alpha_j + \sum_{m=1,\dots,N+1: y^{(m)}=y^{(N+1)}} x_j^{(m)}$ and $\tilde{\beta}_j = \beta_j + \sum_{m=1,\dots,N+1: y^{(m)}=y^{(N+1)}} (1-x_j^{(m)})$.
Which needs to be calculated for $y^{(N+1)} = 0$ and $y^{(N+1)} = 1$ since this expression holds up to the proportionality constant.

D

Mathematical details - model e-mail length

D.1. Generative model

A simple generative model for e-mails will be described. Let $\boldsymbol{z} = (z_1, \ldots, z_n)$ be a sequence of words of an e-mail. Let $x = n$ denote the length of an e-mail. The length of an e-mail will be defined by the number of words present in the e-mail. As has been noted in Chapter 4, the lengths of e-mail messages are Pareto distributed. Therefore, the probability distribution over the e-mail length x can be written as:

$$\begin{aligned} p(x) &= \sum_{y=0,1} p(y)p(x|y) \\ &= \sum_{y=0,1} p(y)\frac{\alpha^y \cdot \eta^{\alpha^y}}{x^{\alpha^y+1}}. \end{aligned}$$

The posterior distribution over x equals:

$$p(x \mid y) = \frac{\alpha^y \cdot \eta^{\alpha^y}}{x^{\alpha^y+1}}. \tag{D.1}$$

D.1.1. Maximum likelihood estimation for α^y

Let $x^{(1)}, \ldots, x^{(N)}$ denote lengths of e-mails. Furthermore, let $y^{(1)}, \ldots, y^{(N)}$ denote whether an e-mail is relevant or not relevant. All e-mails are assumed to be independently generated according to the model. Our available parameters are α^0, α^1 and η, and the probability distribution $p(y)$ is assumed to be fixed. The likelihood of this observation is

$$\begin{aligned} L(\alpha^0, \alpha^1, \eta) &= \prod_{m=1}^{N} p(x^{(m)}, y^{(m)}) \\ &= \prod_{m=1}^{N} p(y^{(m)})p(x^{(m)} \mid y^{(m)}) \\ &= \prod_{m=1}^{N} p(y^{(m)})\frac{\alpha^{y^{(m)}} \cdot \eta^{\alpha^{y^{(m)}}}}{(x^{(m)})^{\alpha^{y^{(m)}}+1}}. \end{aligned}$$

The log likelihood will be, up to irrelevant additional constants, equal to

$$l(\alpha^0, \alpha^1, \eta) = \sum_{m=1}^{N} \ln\left(\alpha^{y^{(m)}}\right) + \alpha^{y^{(m)}} \ln(\eta) - \left(\alpha^{y^{(m)}} + 1\right)\ln\left(x^{(m)}\right).$$

This log likelihood will be maximized subject to the constraint that $q_j^y \geq 0$ for all j and y. This yields

$$\hat{\alpha}^y = \frac{\sum\limits_{m=1,\ldots,N:y^{(m)}=y} 1}{\sum\limits_{m=1,\ldots,N:y^{(m)}=y} \ln\left(\frac{x^{(m)}}{\hat{\eta}}\right)}.$$

Furthermore,

$$\hat{\eta} = \min_{m=1,\dots,N:y^{(m)}=y} \left(x^{(m)}\right).$$

Classification based on MLE

It is known that y gets either the class 'relevant' or 'not relevant', therefore the distribution of y will be assumed to be Bernoulli distributed with parameter p (the probability of an e-mail being relevant), i.e.,

$$p(y) = p^y \cdot (1-p)^{1-y}.$$

In this equation p is a fixed valued hyperparameter.

For the classification of a new e-mail, Bayes' law will be used and the obtained MLE estimators for α^y and η will be plugged in:

$$\begin{aligned} p(y \mid x) &= \frac{p(x \mid y)p(y)}{p(x \mid y=0)p(y=0) + p(x \mid y=1)p(y=1)} && \text{(D.2)} \\ &\propto p(x \mid y)p(y) && \text{(D.3)} \\ &\propto \frac{\hat{\alpha}^y \cdot \eta^{\hat{\alpha}^y}}{x^{\hat{\alpha}^y+1}} \cdot p^y \cdot (1-p)^{1-y}, && \text{(D.4)} \end{aligned}$$

in which

$$\hat{\alpha}^y = \frac{\sum\limits_{m=1,\dots,N:y^{(m)}=y} 1}{\sum\limits_{m=1,\dots,N:y^{(m)}=y} \ln\left(\frac{x^{(m)}}{\hat{\eta}}\right)},$$

and

$$\hat{\eta} = \min_{m=1,\dots,N:y^{(m)}=y} \left(x^{(m)}\right).$$

D.1.2. Bayesian estimation

η will be assumed to be a fixed valued hyperparameter. The value will be taken equal to 1, since that is the smallest size an e-mail can be.

The posterior distribution of $\boldsymbol{\alpha} = (\alpha^0, \alpha^1)$ will become

$$\begin{aligned} p(\boldsymbol{\alpha} \mid y, x) &= \frac{p(x \mid \boldsymbol{\alpha}, y)p(y \mid \boldsymbol{\alpha})p(\boldsymbol{\alpha})}{p(y \mid x)p(x)} \\ &= \frac{p(x \mid \boldsymbol{\alpha}, y)p(y)p(\boldsymbol{\alpha})}{p(y \mid x)p(x)} \\ &\propto p(x \mid \boldsymbol{\alpha}, y)p(\boldsymbol{\alpha}), \end{aligned}$$

in which it is used that

$$p(x \mid \boldsymbol{\alpha}, y) = \frac{\alpha^y \cdot \eta^{\alpha^y}}{x^{\alpha^y+1}}$$

and $p(\alpha^y) \sim Gamma(a, b)$, i.e.

$$p(\alpha^y) = \frac{b^a}{\Gamma(a)}(\alpha^y)^{a-1} \cdot e^{-b \cdot \alpha^y}. \tag{D.5}$$

Combining Equation (D.1) and (D.5), gives us

$$p(\alpha^0, \alpha^1 \mid y, x) \propto \left(\frac{b^a}{\Gamma(a)}(\alpha^y)^{a-1} \cdot e^{-b \cdot \alpha^y} \cdot \frac{\alpha^y \cdot \eta^{\alpha^y}}{x^{\alpha^y+1}}\right) \cdot \left(\frac{b^a}{\Gamma(a)}(\alpha^{1-y})^{a-1} \cdot e^{-b \cdot \alpha^{1-y}} \cdot \frac{\alpha^{1-y} \cdot \eta^{\alpha^{1-y}}}{x^{\alpha^{1-y}+1}}\right).$$

Let $x^{(1)}, \dots, x^{(N)}$ denote the lengths of e-mails. Furthermore, let $y^{(1)}, \dots, y^{(N)}$ denote whether an e-mail is relevant or not. All e-mails are assumed to be independently generated according to the model. Our available parameters are α^0 and α^1, and the probability distribution $p(y)$ is assumed to be fixed. This gives us the following posterior distribution for α^y:

$$\begin{aligned}
p(\alpha^\xi \mid x^{(1)},\dots,x^{(N)},y^{(1)},\dots,y^{(N)}) &\propto \prod_{m=1,\dots,N:y^{(m)}=\xi} p(x^{(m)} \mid \boldsymbol{\alpha}, y^{(m)})p(\boldsymbol{\alpha}) \\
&= \prod_{m=1,\dots,N:y^{(m)}=\xi} \left(\frac{\alpha^\xi \eta^{\alpha^\xi}}{\left(x^{(m)}\right)^{\alpha^\xi+1}}\right) \cdot \frac{b^a}{\Gamma(a)} \left(\alpha^\xi\right)^{a-1} e^{-b\cdot\alpha^\xi} \\
&= \left(\alpha^\xi\right)^{a-1+\sum_{m=1,\dots,N:y^{(m)}=\xi} 1} \cdot \frac{b^a}{\Gamma(a)} \cdot e^{\sum_{m=1,\dots,N:y^{(m)}=\xi} \ln(\eta)\alpha^\xi - \sum_{m=1,\dots,N:y^{(m)}=\xi} \ln(x^{(m)})(\alpha^\xi+1) - b\alpha^\xi} \\
&= \left(\alpha^\xi\right)^{a-1+\sum_{m=1,\dots,N:y^{(m)}=\xi} 1} \cdot e^{\alpha^\xi\left(\sum_{m=1,\dots,N:y^{(m)}=\xi} \left(\ln(\eta)-\ln(x^{(m)})\right)-b\right)} \\
&\sim Gamma(\hat{a}^\xi, \hat{b}^\xi),
\end{aligned}$$

with $\xi = 0,1$. Furthermore, $\hat{a}^\xi = a + \sum_{m=1,\dots,N:y^{(m)}=y} 1$ and $\hat{b}^\xi = -\sum_{m=1,\dots,N:y^{(m)}=y} \left(\ln(\eta) - \ln(x^{(m)})\right) + b$. It is common to set the hyperparameters a and b equal to 1.

Classification based on Bayesian estimate

It is known that y gets either the class 'relevant' or 'not relevant', therefore the distribution of y will be assumed to be Bernoulli distributed with parameter p (the probability of an e-mail being spam), i.e.,

$$p(y) = p^y \cdot (1-p)^{1-y}.$$

In this equation p is a fixed valued hyperparameter.

Using Bayes' law and the formula (with conditioning on $\boldsymbol{q}$) $p(y \mid x) = \frac{p(x|y)p(y)}{p(x|y=0)p(y=0)+p(x|y=1)p(y=1)}$ of Section B.1.1:

$$\begin{aligned}
p(y \mid x, \boldsymbol{\alpha}) &\propto p(x \mid y, \boldsymbol{\alpha})p(\boldsymbol{\alpha} \mid y)p(y) \\
&\propto p(x \mid y, \boldsymbol{\alpha})p(y) \\
&\propto \frac{\alpha^y \cdot \eta^{\alpha^y}}{x^{\alpha^y+1}} \cdot p^y(1-p)^{1-y}.
\end{aligned}$$

This gives us that the formula for classification equals:

$$p(y \mid \boldsymbol{x}) = \int_0^\infty \frac{\alpha^y \cdot \eta^{\alpha^y}}{x^{\alpha^y+1}} \cdot p^y(1-p)^{1-y} d\boldsymbol{\alpha}.$$

Therefore, when taking into account the e-mail length $x^{(1)},\dots,x^{(N)}$ together with information $y^{(1)},\dots,y^{(N)}$, the conditional distribution of $y^{(N+1)}$ for our new e-mail can be written as:

$$\begin{aligned}
p(y^{(N+1)} \mid x^{(1)},\dots,x^{(N)},y^{(1)},\dots,y^{(N)},x^{(N+1)}) &\propto \int_0^\infty p(y^{(N+1)} \mid x^{(N+1)}, \boldsymbol{\alpha})p(\boldsymbol{\alpha} \mid x^{(1)},\dots,x^{(N)},y^{(1)},\dots,y^{(N)}) d\alpha^{y^{(N+1)}} \\
&= \int_0^\infty \left(\alpha^{y^{(N+1)}}\right)^{\hat{a}^{y^{(N+1)}}} \cdot \frac{\eta^{\alpha^{y^{(N+1)}}}}{\left(x^{(N+1)}\right)^{\alpha^{y^{(N+1)}}}} \cdot e^{-\hat{b}^{y^{(N+1)}}\alpha^{y^{(N+1)}}} \cdot p^{y^{(N+1)}}(1-p)^{y^{(N+1)}} d\alpha^{y^{(N+1)}} \\
&= p^{y^{(N+1)}}(1-p)^{y^{(N+1)}} \cdot \int_0^\infty \left(\alpha^{y^{(N+1)}}\right)^{\hat{a}^{y^{(N+1)}}} \cdot e^{-\left(\hat{b}^{y^{(N+1)}} - \ln\left(\frac{\eta}{x^{N+1}}\right)\right)\alpha^{y^{(N+1)}}} d\alpha^{y^{(N+1)}} \\
&= p^{y^{(N+1)}}(1-p)^{y^{(N+1)}} \cdot \frac{\Gamma(\hat{a}^{y^{(N+1)}}+1)}{\left(\hat{b}^{y^{(N+1)}} - \ln\left(\frac{\eta}{x^{(N+1)}}\right)\right)^{\hat{a}^{y^{(N+1)}}+1}}
\end{aligned}$$

Which needs to be calculated for $y^{(N+1)} = 0$ and $y^{(N+1)} = 1$ since this expression holds up to the proportionality constant.

E

Additional data analysis

E.1. Length of small e-mails ENRON dataset

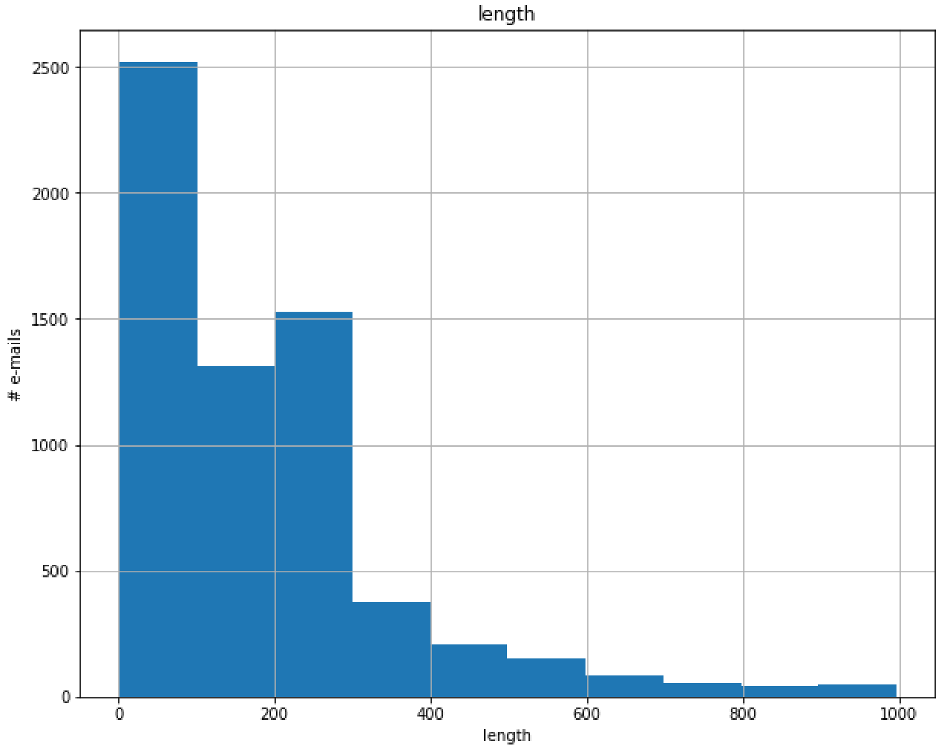


Figure E.1: Histogram of the length of e-mails which contain less than 1000 words in the ENRON dataset (in which length is defined as the number of words)

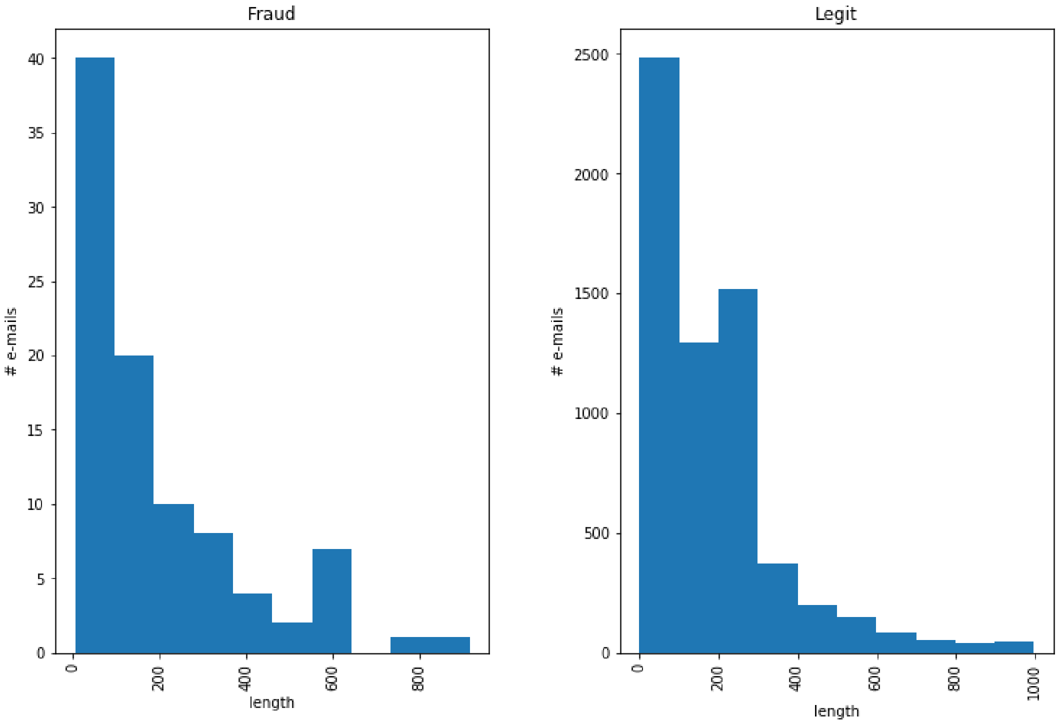


Figure E.2: Histogram of the length of e-mails which contain less than 1000 words in the ENRON dataset, shown for both categories (in which length is defined as the number of words)

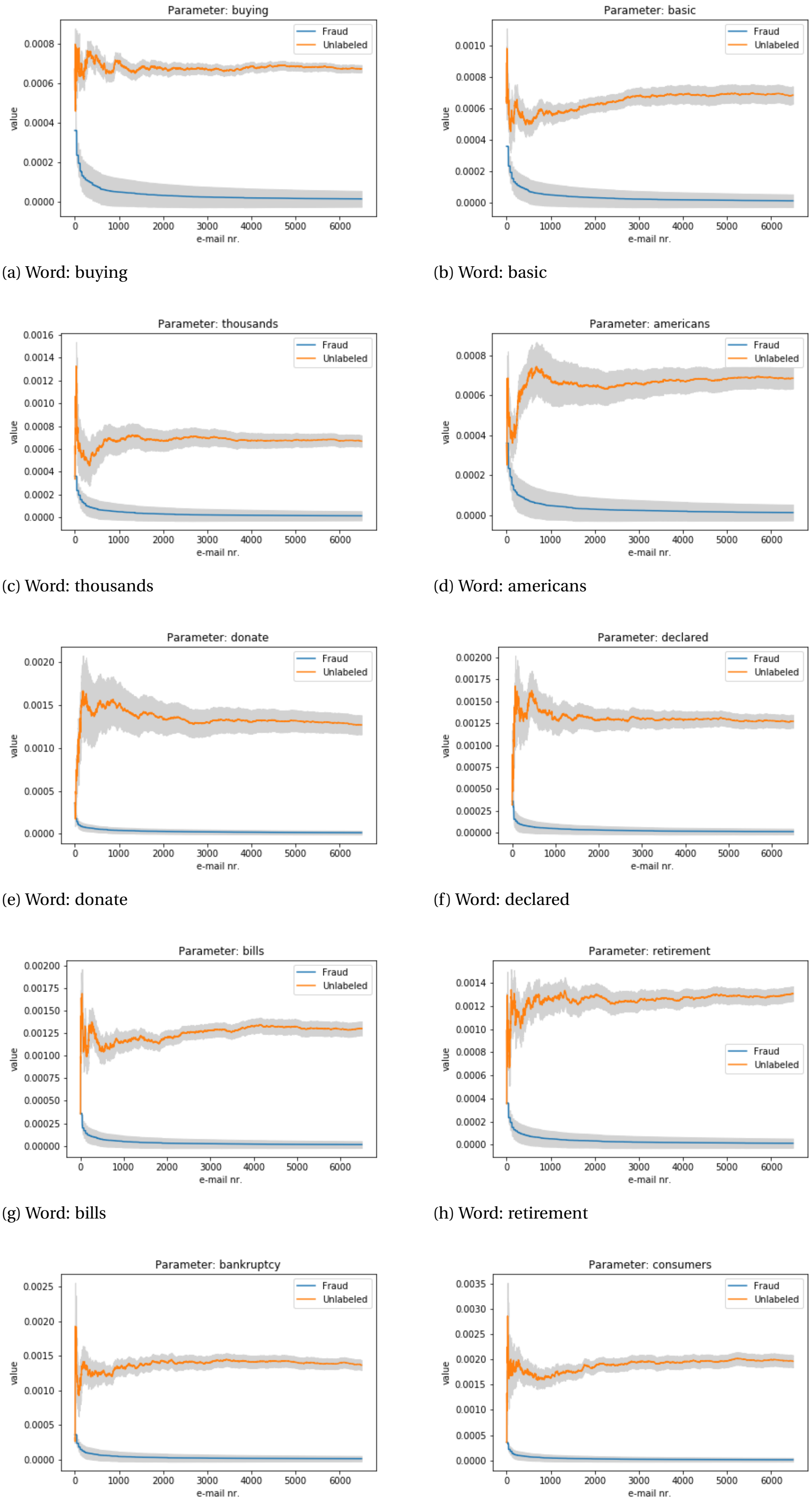
F

Additional Results

F.1. Model word frequencies

F.1.1. Parameter analysis

ENRON



(a) Word: buying

(b) Word: basic

(c) Word: thousands

(d) Word: americans

(e) Word: donate

(f) Word: declared

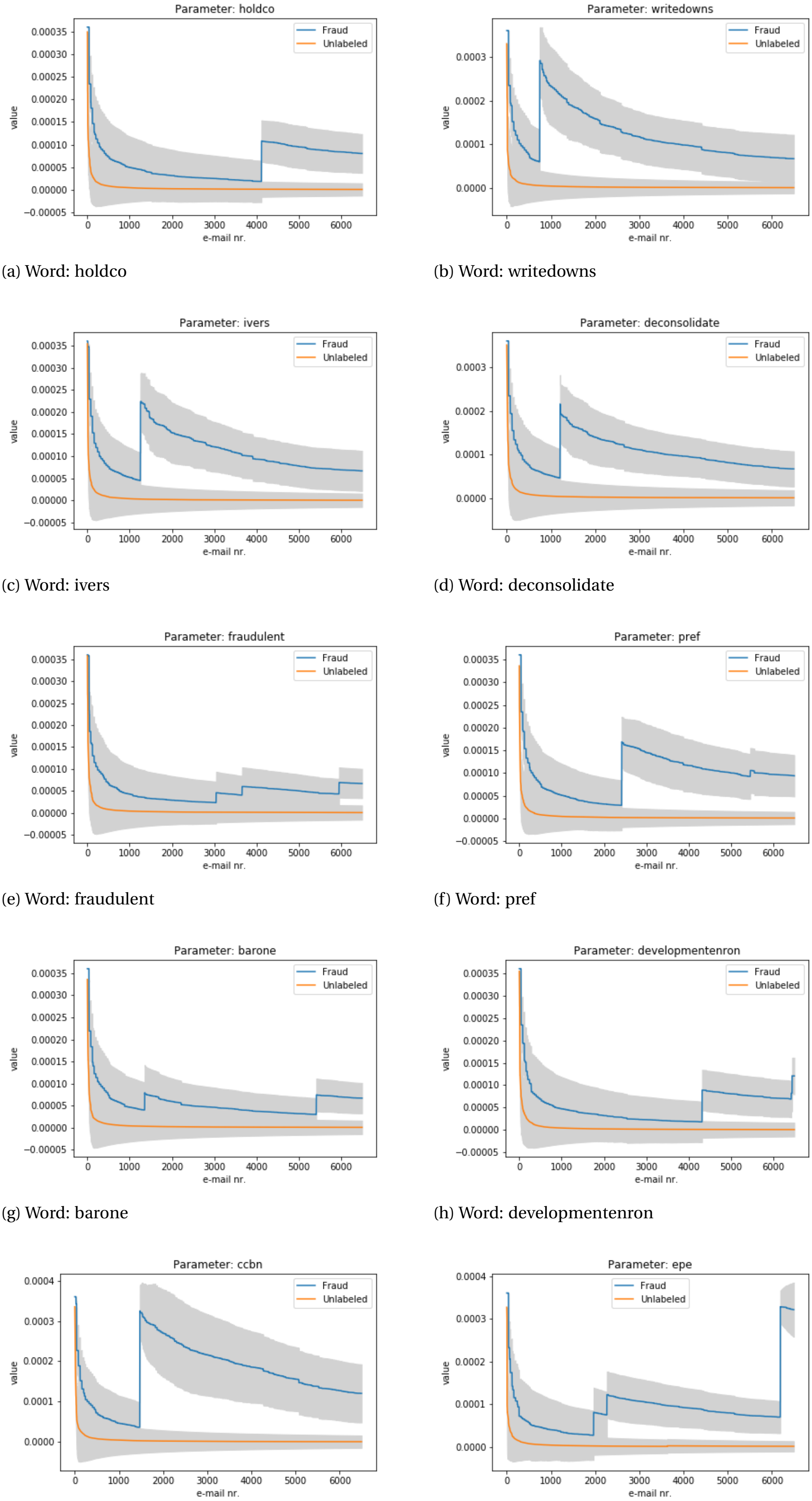
(g) Word: bills

(h) Word: retirement

(i) Word: bankruptcy

(j) Word: consumers

Figure F.1: Training process generative model with MLE for words identifiable as unlabeled (ENRON)



(a) Word: holdco

(b) Word: writedowns

(c) Word: ivers

(d) Word: deconsolidate

(e) Word: fraudulent

(f) Word: pref

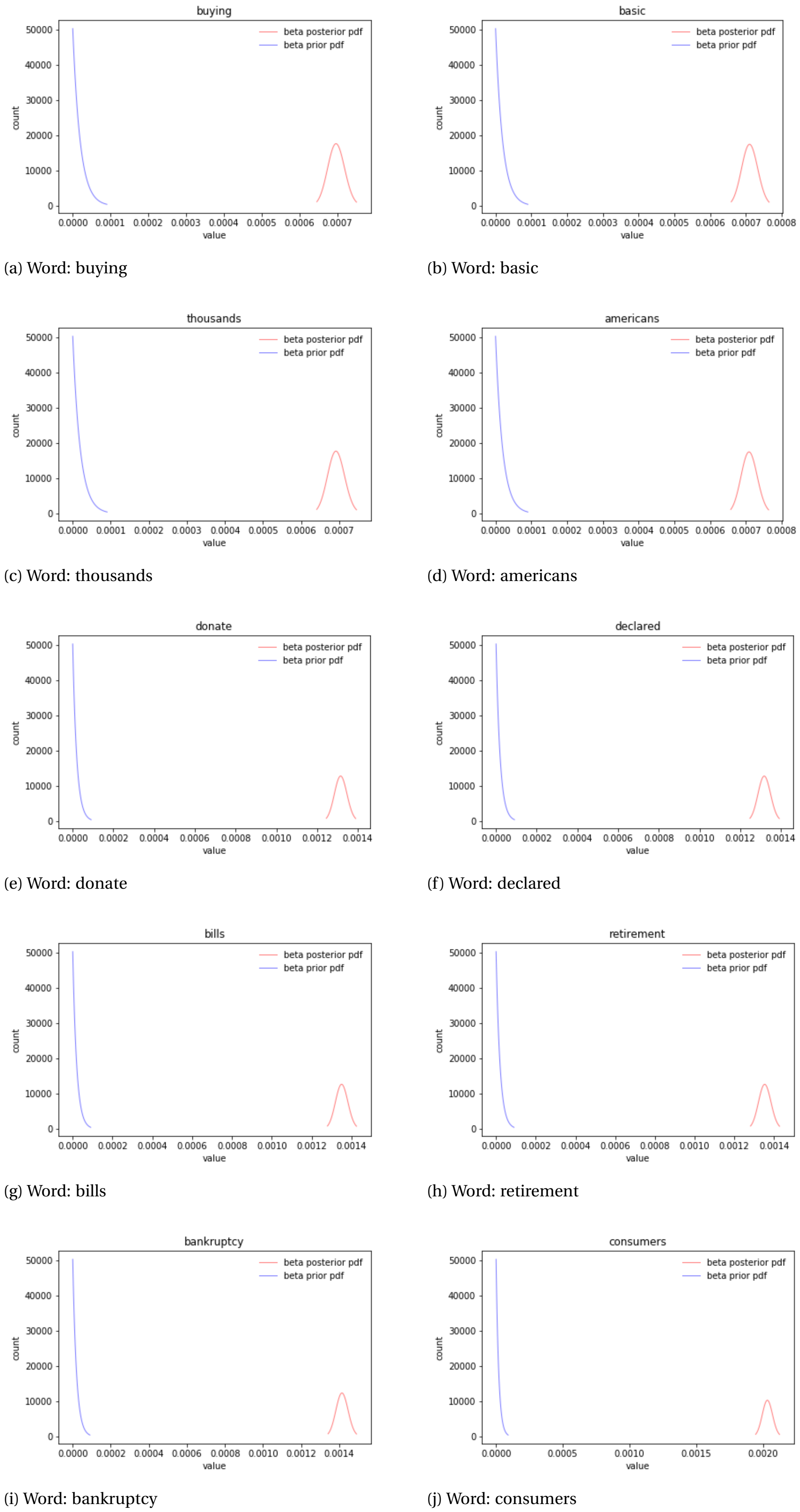
(g) Word: barone

(h) Word: developmentenron

(i) Word: ccbn

(j) Word: epe

Figure F.2: Training process generative model with MLE for words identifiable as relevant (ENRON)



(a) Word: buying

(b) Word: basic

(c) Word: thousands

(d) Word: americans

(e) Word: donate

(f) Word: declared

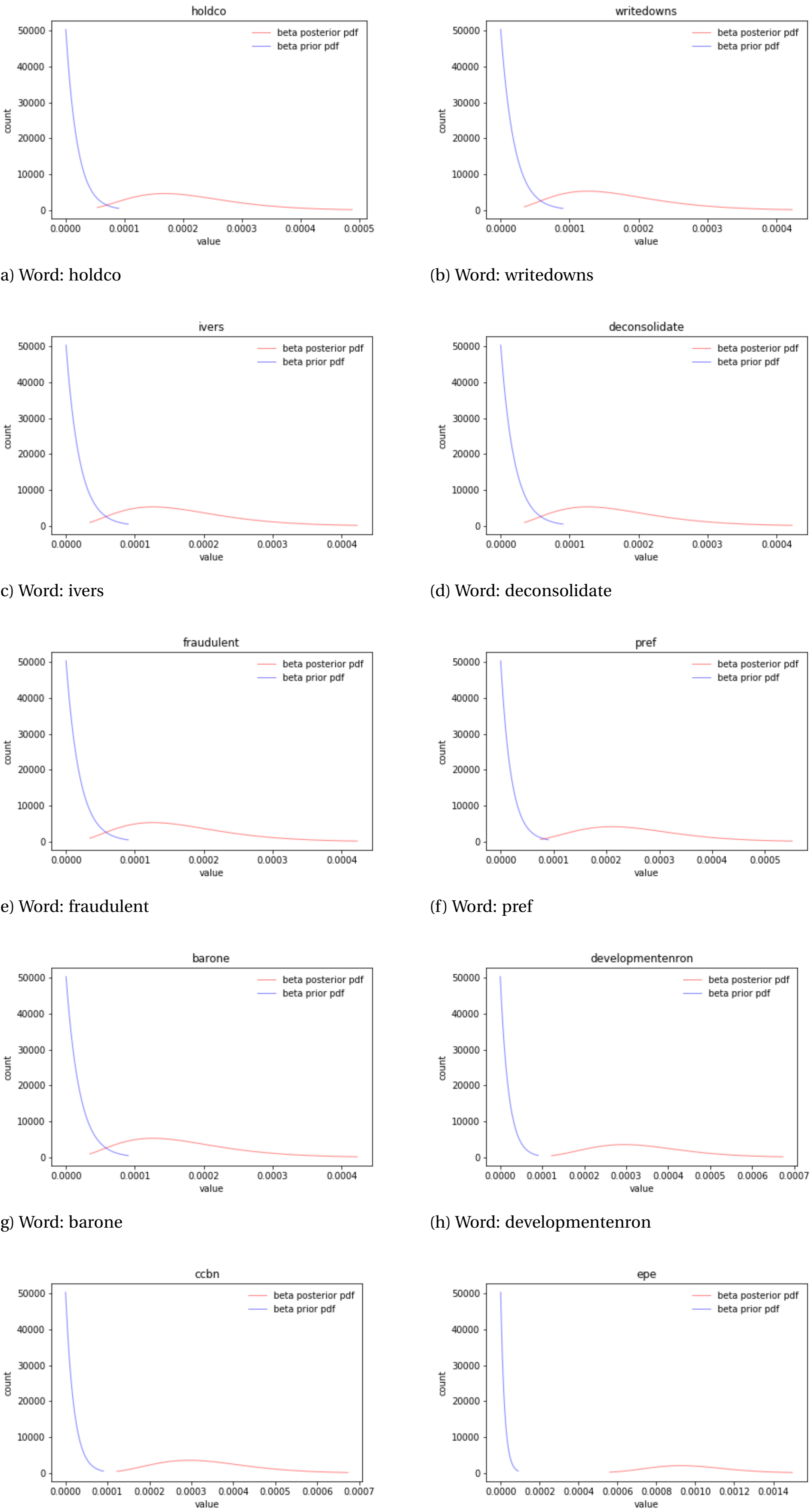
(g) Word: bills

(h) Word: retirement

(i) Word: bankruptcy

(j) Word: consumers

Figure F.3: Training process generative model with Bayesian estimation words identifiable as unlabeled (ENRON). The two lines indicate the prior and posterior marginal distribution of the corresponding word.



(a) Word: holdco

(b) Word: writedowns

(c) Word: ivers

(d) Word: deconsolidate

(e) Word: fraudulent

(f) Word: pref

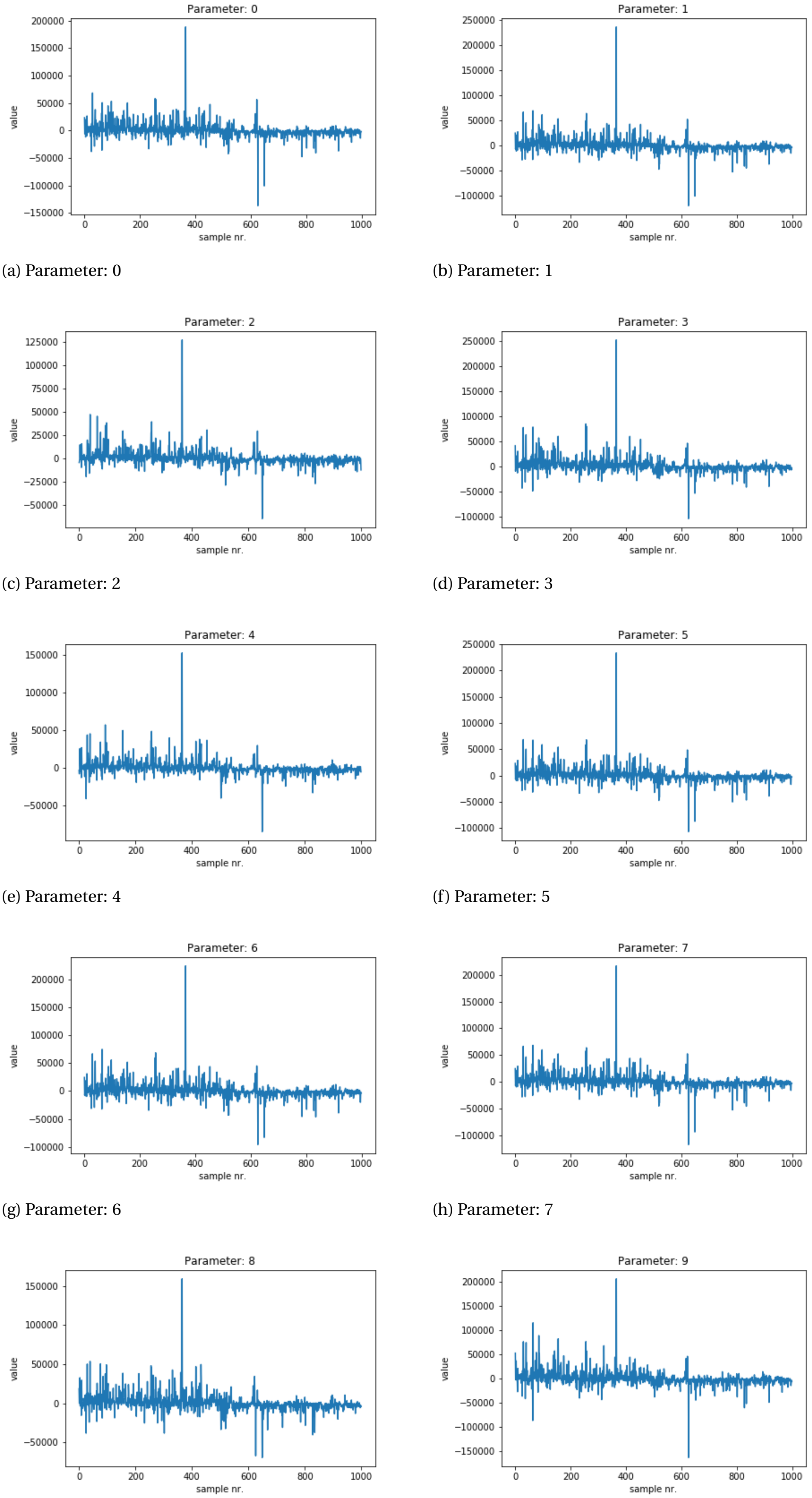
(g) Word: barone

(h) Word: developmentenron

(i) Word: ccbn

(j) Word: epe

Figure F.4: Training process generative model with Bayesian estimation for words identifiable as relevant (ENRON). The two lines indicate the prior and posterior marginal distribution of the corresponding word.



(a) Parameter: 0

(b) Parameter: 1

(c) Parameter: 2

(d) Parameter: 3

(e) Parameter: 4

(f) Parameter: 5

(g) Parameter: 6

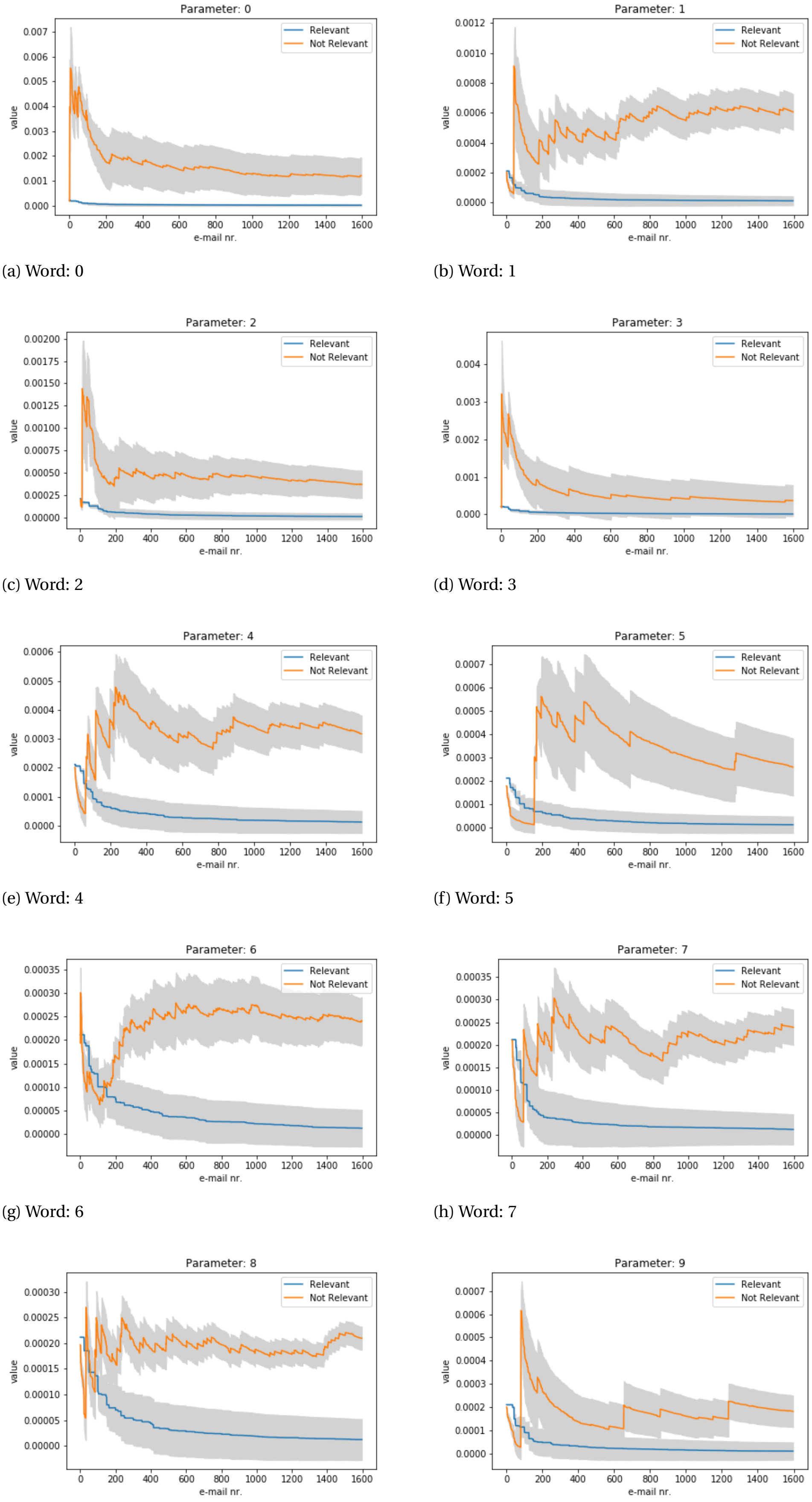
(h) Parameter: 7

(i) Parameter: 8

(j) Parameter: 9

Figure F.5: Training process discriminative model for the feature of word frequencies with Bayesian estimation (ENRON dataset).The blue line indicates the sample values on which the parameter value is based. The titles of each plot are numbered because it is now 100% known to which word the parameter used by the package is related.

confidential dataset



(a) Word: 0

(b) Word: 1

(c) Word: 2

(d) Word: 3

(e) Word: 4

(f) Word: 5

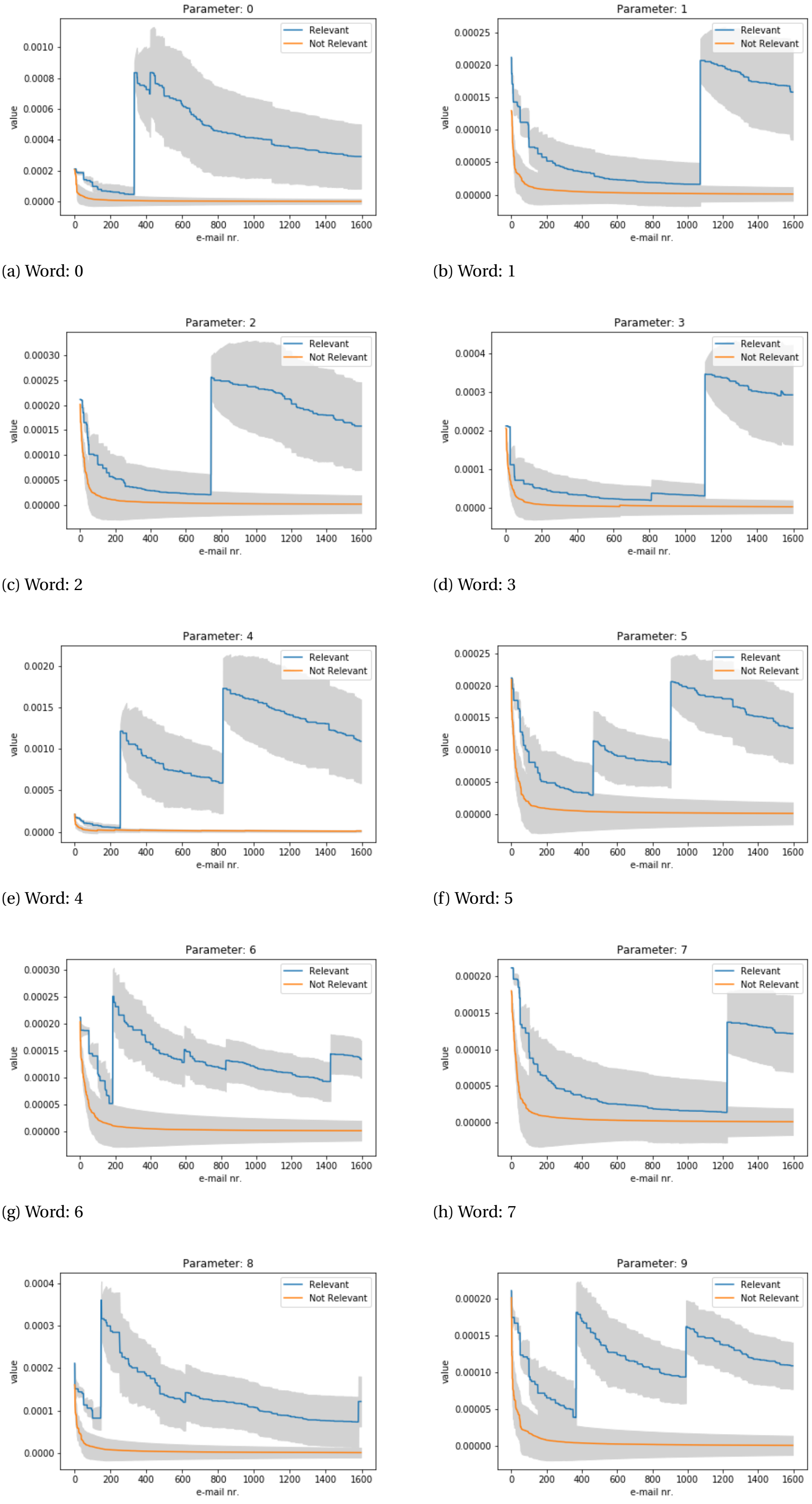
(g) Word: 6

(h) Word: 7

(i) Word: 8

(j) Word: 9

Figure F.6: Training process generative model with MLE for words identifiable as not relevant (confidential dataset). The words are numbered due to the confidentiality of the dataset.



(a) Word: 0

(b) Word: 1

(c) Word: 2

(d) Word: 3

(e) Word: 4

(f) Word: 5

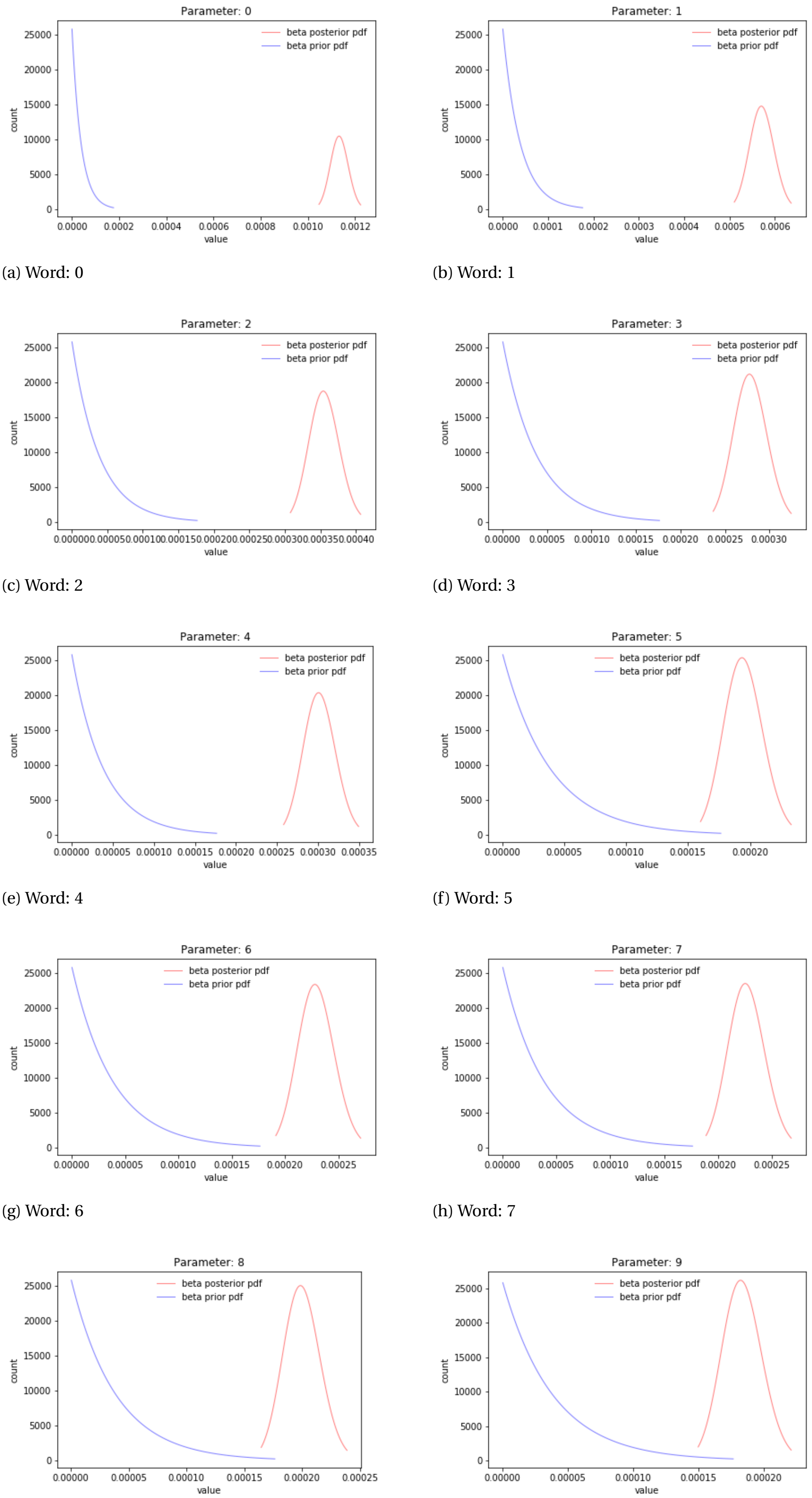
(g) Word: 6

(h) Word: 7

(i) Word: 8

(j) Word: 9

Figure F.7: Training process generative model with MLE for words identifiable as relevant (confidential dataset). The words are numbered due to the confidentiality of the dataset.



(a) Word: 0

(b) Word: 1

(c) Word: 2

(d) Word: 3

(e) Word: 4

(f) Word: 5

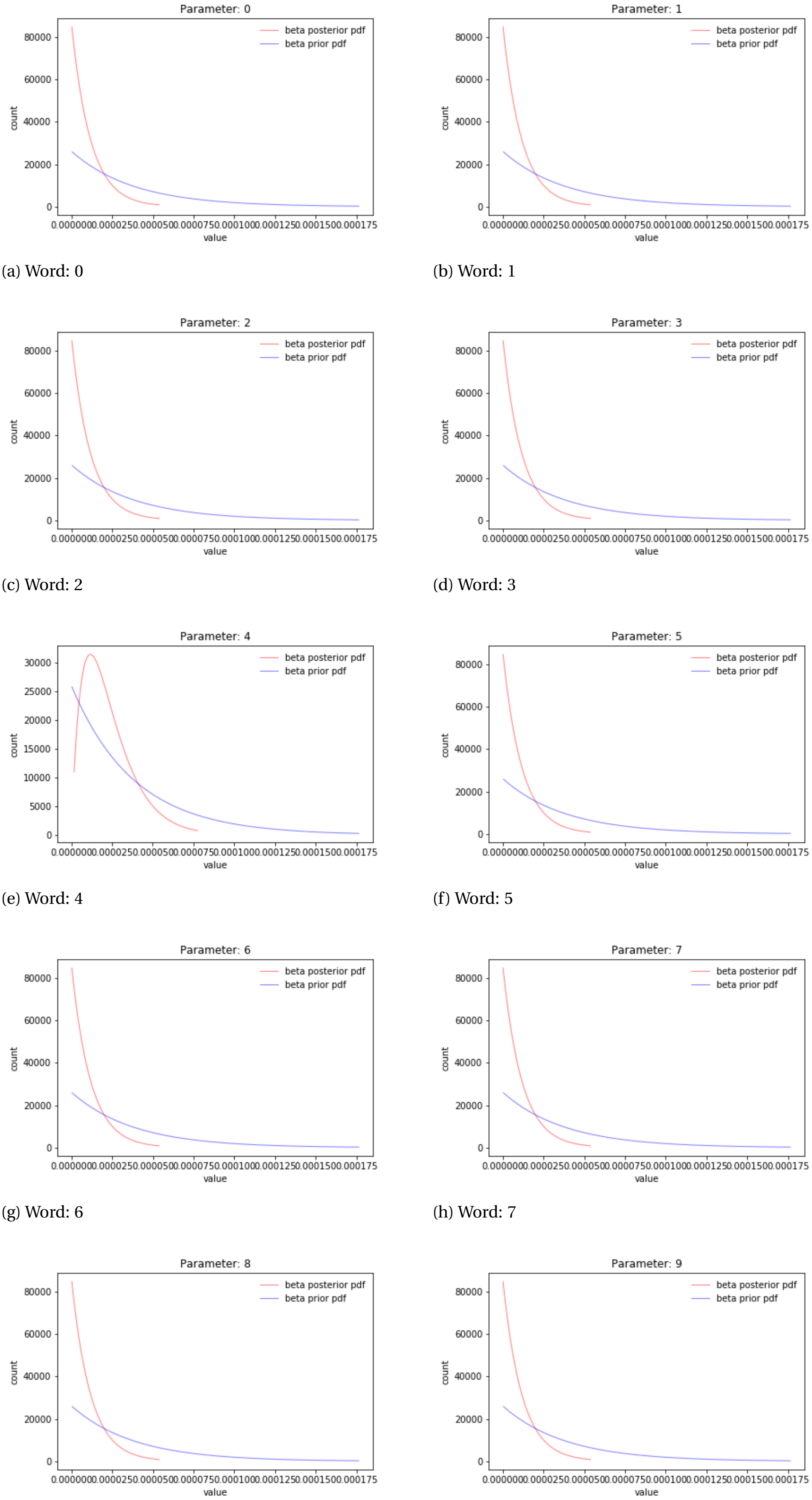
(g) Word: 6

(h) Word: 7

(i) Word: 8

(j) Word: 9

Figure F.8: Training process generative model with Bayesian estimation words identifiable as not relevant (confidential dataset). The two lines indicate the prior and posterior marginal distribution of the corresponding word. The words are numbered due to the confidentiality of the dataset.



(a) Word: 0

(b) Word: 1

(c) Word: 2

(d) Word: 3

(e) Word: 4

(f) Word: 5

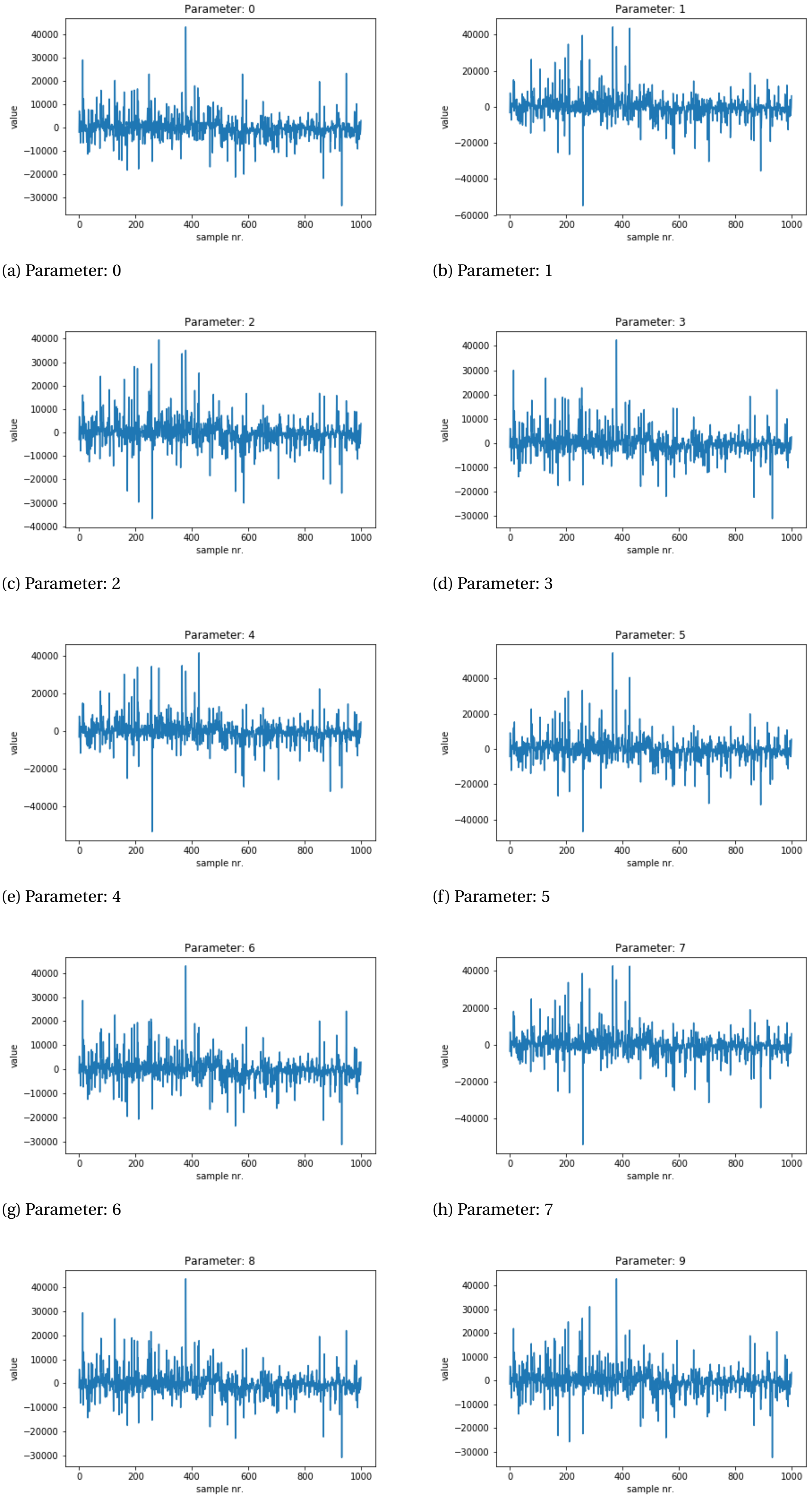
(g) Word: 6

(h) Word: 7

(i) Word: 8

(j) Word: 9

Figure F.9: Training process generative model with Bayesian estimation words identifiable as relevant (confidential dataset). The two lines indicate the prior and posterior marginal distribution of the corresponding word. The words are numbered due to the confidentiality of the dataset.



(a) Parameter: 0

(b) Parameter: 1

(c) Parameter: 2

(d) Parameter: 3

(e) Parameter: 4

(f) Parameter: 5

(g) Parameter: 6

(h) Parameter: 7

(i) Parameter: 8

(j) Parameter: 9

Figure F.10: Training process discriminative model with Bayesian estimation for feature word frequencies (confidential dataset). The blue line indicates the sample values on which the parameter value is based. As can be noted the sample values show much difference.

F.1.2. Results classification

confidential dataset

	ML (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	339	306	367	353	304	43
# False Positives	35	75	0	21	75	356
# False Negatives	25	18	32	25	17	0
# True Positives	7	14	0	7	15	32
# True Negatives	332	292	367	346	289	11
Accuracy	85.0%	76.6%	92.0%	88.5%	76.8%	10.8%
Error rate	15.0%	23.4%	8.0%	11.5%	23.2%	89.2%
'relevant' recall	21.9%	43.8%	0.0%	21.9%	46.9%	100%
'relevant' precision	16.7%	15.7%	n.a.	25.0%	16.7%	8.2%
'not relevant' recall	90.5%	79.6%	100%	94,3%	79.4%	3.0%
'not relevant' precision	93.0%	94.2%	92.0%	93.3%	99.4%	100%
'relevant' F-score	18.9%	23.1%	n.a.	23.3%	24.6%	15.2%
'not relevant' F-score	91.7%	86.3%	95.8%	93.8%	88.3%	5.8%
'relevant' percentage	8.0%	8.0%	8.0%	8.0%	8.0%	8.0%
Time (test)	67 sec	349 sec	47 sec	10 sec	8 sec	117 sec
Time (training)	294 sec	286 sec	33 sec	26434 sec	16 sec	19 sec

Table F.1: Performance results (set 1), based on e-mail messages in a confidential dataset and using the feature word frequencies. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category 'relevant' and the top 500 words identified as best corresponding to the category 'not relevant'.

	MLE (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	349	292	379	363	316	27
# False Positives	38	101	0	24	67	372
# False Negatives	12	6	20	23	8	0
# True Positives	8	14	0	8	12	20
# True Negatives	341	278	379	355	304	7
Accuracy	87.5%	73.2%	95.0%	91.0%	80.8%	6.8%
Error rate	12.5%	26.8%	5.0%	9.0%	19.2%	93.2%
'relevant' recall	40.0%	70.0%	0.0%	40.0%	60.0%	100%
'relevant' precision	17.4%	12.2%	n.a.	25.0%	15.2%	5.1%
'not relevant' recall	90.0%	73.4%	100%	93.7%	81.9%	1.8%
'not relevant' precision	96.6%	97.9%	95.0%	96.7%	97.4%	100%
'relevant' F-score	24.3%	20.8%	n.a.	30.8%	24.3%	9.7%
'not relevant' F-score	93.2%	83.9%	97.4%	95.2%	89.0%	3.5%
'relevant' percentage	5.0%	5.0%	5.0%	5.0%	5.0%	5.0%
Time (test)	78 sec	353 sec	51 sec	24 sec	8 sec	131 sec
Time (training)	300 sec	295 sec	33 sec	26554 sec	16 sec	19 sec

Table F.2: Performance results (set 2), based on e-mail messages in a confidential dataset and using the feature word frequencies. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category 'relevant' and the top 500 words identified as best corresponding to the category 'not relevant'.

	MLE (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	346	321	368	360	309	37
# False Positives	29	65	0	16	72	362
# False Negatives	24	13	31	3	15	0
# True Positives	7	18	0	8	16	31
# True Negatives	339	303	368	352	293	6
Accuracy	86.7%	80.5%	92.2%	90.2%	78.0%	9.3%
Error rate	13.3%	19.5%	7.8%	9.8%	22.0%	90.7%
’relevant’ recall	22.6%	58.1%	0.0%	25.8%	51.6%	100%
’relevant’ precision	19.4%	21.7%	n.a.	33.3%	18.2%	7.9%
’not relevant’ recall	92.1%	82.3%	100%	95.7%	80.3%	1.6%
’not relevant’ precision	93.4%	95.9%	92.2%	93.9%	95.1%	100%
’relevant’ F-score	20.9%	31.6%	n.a.	29.1%	26.9%	14.6%
’not relevant’ F-score	92.8%	88.6%	95.9%	94.8%	87.1%	3.1%
’relevant’ percentage	7.8%	7.8%	7.8%	7.8%	7.8%	7.8%
Time (test)	62 sec	357 sec	40 sec	24 sec	7 sec	131 sec
Time (training)	313 sec	308 sec	34 sec	26563 sec	17 sec	19 sec

Table F.3: Performance results (set 3), based on e-mail messages in a confidential dataset and using the feature word frequencies. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’.

	MLE (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	329	266	377	365	312	34
# False Positives	54	122	0	18	65	365
# False Negatives	16	11	22	16	14	0
# True Positives	6	11	0	6	8	22
# True Negatives	323	255	377	359	304	12
Accuracy	82.5%	66.7%	94.5%	91.5%	79.8%	8.5%
Error rate	17.5%	33.3%	5.5%	8.5%	20.2%	91.5%
’relevant’ recall	27.3%	50.0%	0.0%	27.3%	36.4%	100%
’relevant’ precision	10.0%	8.3%	n.a.	25.0%	11.0%	5.7%
’not relevant’ recall	85.7%	67.6%	100%	95.2%	82.4%	3.2%
’not relevant’ precision	95.3%	95.9%	94.5%	95.7%	95.6%	100%
’relevant’ F-score	14.6%	14.2%	n.a.	26.1%	16.9%	10.8%
’not relevant’ F-score	90.2%	79.3%	97.2%	95.4%	88.5%	6.2%
’relevant’ percentage	5.5%	5.5%	5.5%	5.5%	5.5%	5.5%
Time (test)	71 sec	354 sec	49 sec	24 sec	8 sec	130 sec
Time (training)	301 sec	301 sec	34 sec	26653 sec	16 sec	19 sec

Table F.4: Performance results (set 4), based on e-mail messages in a confidential dataset and using the feature word frequencies. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’.

	MLE (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	346	337	368	356	311	40
# False Positives	30	43	0	17	59	359
# False Negatives	23	19	31	26	22	0
# True Positives	8	12	0	5	9	31
# True Negatives	338	325	368	351	303	9
Accuracy	86.7%	84.5%	92.2%	89.2%	79.4%	10.0%
Error rate	13.3%	15.5%	7.8%	10.8%	20.6%	90.0%
’relevant’ recall	25.8%	38.7%	0.0%	16.1%	29.0%	100%
’relevant’ precision	21.1%	21.8%	n.a.	22.7%	13.2%	7.9%
’not relevant’ recall	91.8%	88.3%	100%	95.4%	83.7%	2.4%
’not relevant’ precision	93.6%	94.5%	92.2%	93.1%	93.2%	100%
’relevant’ F-score	23.2%	27.9%	n.a.	18.8%	18.1%	14.6%
’not relevant’ F-score	92.7%	91.3%	95.9%	94.2%	88.2%	4.7%
’relevant’ percentage	7.8%	7.8%	7.8%	7.8%	7.8%	7.8%
Time (test)	70 sec	359 sec	42 sec	23 sec	9 sec	129 sec
Time (training)	314 sec	309 sec	33 sec	27423 sec	17 sec	19 sec

Table F.5: Performance results (set 5), based on e-mail messages in a confidential dataset and using the feature word frequencies. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’.

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	10	0	0
$10^2 > B > 10^{\frac{3}{2}}$	0	0	0	0
$10^{\frac{3}{2}} > B > 10$	0	0	0	0
$10 > B > 10^{\frac{1}{2}}$	0	1	0	0
$10^{\frac{1}{2}} > B > 1$	0	0	0	0
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	0	0	3	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	0	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	0	0	0	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	0	0	0	0
$10^{-2} > B$	32	0	353	0

Table F.6: Bayes Factor results word frequencies for the generative model with Bayesian estimation, based on 1000 words as parameters (set 1)

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	7	0	0
$10^2 > B > 10^{\frac{3}{2}}$	0	0	0	0
$10^{\frac{3}{2}} > B > 10$	0	0	0	0
$10 > B > 10^{\frac{1}{2}}$	0	0	0	0
$10^{\frac{1}{2}} > B > 1$	0	0	0	0
$B = 1$	0	0	8	0
$1 > B > 10^{-\frac{1}{2}}$	0	0	0	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	0	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	0	0	1	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	0	0	0	0
$10^{-2} > B$	20	0	363	0

Table F.7: Bayes Factor results word frequencies for the generative model with Bayesian estimation, based on 1000 words as parameters (set 2)

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	6	0	0
$10^2 > B > 10^{\frac{3}{2}}$	0	0	0	0
$10^{\frac{3}{2}} > B > 10$	0	0	0	0
$10 > B > 10^{\frac{1}{2}}$	0	0	0	0
$10^{\frac{1}{2}} > B > 1$	0	0	0	0
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	0	0	3	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	0	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	0	0	0	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	0	0	0	0
$10^{-2} > B$	31	0	359	0

Table F.8: Bayes Factor results word frequencies for the generative model with Bayesian estimation, based on 1000 words as parameters (set 3)

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	12	0	0
$10^2 > B > 10^{\frac{3}{2}}$	0	0	0	0
$10^{\frac{3}{2}} > B > 10$	0	0	0	0
$10 > B > 10^{\frac{1}{2}}$	0	0	0	0
$10^{\frac{1}{2}} > B > 1$	0	0	0	0
$B = 1$	0	0	8	0
$1 > B > 10^{-\frac{1}{2}}$	0	0	0	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	0	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	0	0	0	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	0	0	0	0
$10^{-2} > B$	22	0	357	0

Table F.9: Bayes Factor results word frequencies for the generative model with Bayesian estimation, based on 1000 words as parameters (set 4)

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	8	0	0
$10^2 > B > 10^{\frac{3}{2}}$	0	1	0	0
$10^{\frac{3}{2}} > B > 10$	0	0	0	0
$10 > B > 10^{\frac{1}{2}}$	0	0	0	0
$10^{\frac{1}{2}} > B > 1$	0	0	0	0
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	0	0	6	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	0	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	0	0	0	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	0	0	0	0
$10^{-2} > B$	31	0	353	0

Table F.10: Bayes Factor results word frequencies for the generative model with Bayesian estimation, based on 1000 words as parameters (set 5)

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	260	0	12
$10^2 > B > 10^{\frac{3}{2}}$	0	17	0	2
$10^{\frac{3}{2}} > B > 10$	0	3	0	2
$10 > B > 10^{\frac{1}{2}}$	0	8	0	2
$10^{\frac{1}{2}} > B > 1$	0	4	0	0
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	0	0	7	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	1	0	7	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	0	0	8	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	0	0	8	0
$10^{-2} > B$	13	0	45	0

Table F.11: Bayes Factor results word frequencies for the generative model with Bayesian estimation (set 1)

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	240	0	6
$10^2 > B > 10^{\frac{3}{2}}$	0	11	0	0
$10^{\frac{3}{2}} > B > 10$	0	5	0	0
$10 > B > 10^{\frac{1}{2}}$	0	11	0	0
$10^{\frac{1}{2}} > B > 1$	0	11	0	0
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	0	0	16	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	1	0	12	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	0	0	9	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	1	0	4	0
$10^{-2} > B$	12	0	60	0

Table F.12: Bayes Factor results word frequencies for the generative model with Bayesian estimation (set 2)

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	282	0	13
$10^2 > B > 10^{\frac{3}{2}}$	0	7	0	0
$10^{\frac{3}{2}} > B > 10$	0	5	0	0
$10 > B > 10^{\frac{1}{2}}$	0	7	0	0
$10^{\frac{1}{2}} > B > 1$	0	2	0	0
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	0	0	8	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	2	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	1	0	4	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	0	0	3	0
$10^{-2} > B$	17	0	48	0

Table F.13: Bayes Factor results word frequencies for the generative model with Bayesian estimation (set 3)

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	228	0	9
$10^2 > B > 10^{\frac{3}{2}}$	0	6	0	0
$10^{\frac{3}{2}} > B > 10$	0	3	0	0
$10 > B > 10^{\frac{1}{2}}$	0	3	0	0
$10^{\frac{1}{2}} > B > 1$	0	15	0	2
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	0	0	11	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	11	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	0	0	10	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	1	0	11	0
$10^{-2} > B$	10	0	79	0

Table F.14: Bayes Factor results word frequencies for the generative model with Bayesian estimation (set 4)

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	299	0	17
$10^2 > B > 10^{\frac{3}{2}}$	0	4	0	0
$10^{\frac{3}{2}} > B > 10$	0	11	0	1
$10 > B > 10^{\frac{1}{2}}$	0	3	0	1
$10^{\frac{1}{2}} > B > 1$	0	8	0	0
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	1	0	1	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	1	0	4	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	0	0	1	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	1	0	0	0
$10^{-2} > B$	9	0	37	0

Table F.15: Bayes Factor results word frequencies for the generative model with Bayesian estimation (set 5)

ENRON dataset

	ML (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	1574	1575	1596	1582	27	27
# False Positives	26	25	0	16	1596	1596
# False Negatives	21	21	27	25	0	0
# True Positives	1	1	0	2	27	27
# True Negatives	1573	1574	1596	1580	0	0
Accuracy	97.1%	97.2%	98.3%	97.5%	1.7%	1.7%
Error rate	2.9%	2.8%	1.7%	2.5%	98.3%	98.3%
'relevant' recall	4.5%	4.5%	0.0%	7.4%	100%	100%
'relevant' precision	3.7%	3.8%	n.a.	11.1%	1.7%	1.7%
'not relevant' recall	98.4%	98.4%	100%	99.0%	0.0%	0.0%
'not relevant' precision	98.7%	98.7%	98.3%	98.5%	n.a.	n.a.
'relevant' F-score	4.1%	4.1%	n.a.	8.9%	3.3%	3.3%
'not relevant' F-score	98.5%	98.5%	99.1%	98.7%	n.a.	n.a.
'relevant' percentage	1.7%	1.7%	1.7%	1.7%	1.7%	1.7%
Time (test)	119 sec	1740 sec	96 sec	80 sec	19 sec	210 sec
Time (training)	1114 sec	1104 sec	437 sec	92269 sec	32 sec	109 sec

Table F.16: Performance results (set 1), based on e-mail messages in the ENRON dataset and using the feature word frequencies. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category 'relevant' and the top 500 words identified as best corresponding to the category 'not relevant'.

	MLE (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	1571	1570	1599	1583	23	23
# False Positives	28	29	0	13	1599	1599
# False Negatives	22	22	25	25	0	0
# True Positives	0	0	0	0	23	23
# True Negatives	1571	1570	1599	1585	0	0
Accuracy	96.9%	96.9%	98.5%	97.7%	1.4%	1.4%
Error rate	3.1%	3.1%	1.5%	2.3%	98.6%	98.6%
’relevant’ recall	0.0%	0.0%	0.0%	0.0%	100%	100%
’relevant’ precision	0.0%	0.0%	n.a.	0.0%	1.4%	1.4%
’not relevant’ recall	98.2%	98.2%	100%	99.2%	0.0%	0.0%
’not relevant’ precision	98.6%	98.6%	98.5%	98.4%	n.a.	n.a.
’relevant’ F-score	n.a.	n.a.	n.a.	n.a.	2.8%	2.8%
’not relevant’ F-score	98.4%	98.4%	99.2%	98.8%	n.a.	n.a.
’relevant’ percentage	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%
Time (test)	119 sec	1860 sec	90 sec	65 sec	19 sec	205 sec
Time (training)	1077 sec	1056 sec	461 sec	93568 sec	38 sec	92 sec

Table F.17: Performance results (set 2), based on e-mail messages in the ENRON dataset and using the feature word frequencies. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’.

	MLE (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	1571	1562	1594	1578	30	30
# False Positives	29	38	0	15	1594	1594
# False Negatives	21	21	30	29	0	0
# True Positives	1	1	0	1	30	30
# True Negatives	1570	1561	1594	1577	0	0
Accuracy	96.9%	96.4%	98.2%	97.3%	1.8%	1.8%
Error rate	3.1%	3.6%	1.8%	2.7%	98.2%	98.2%
’relevant’ recall	4.5%	4.5%	0.0%	3.3%	100%	100%
’relevant’ precision	3.3%	2.6%	n.a.	6.2%	1.8%	1.8%
’not relevant’ recall	98.2%	97.6%	100%	99.1%	0.0%	0.0%
’not relevant’ precision	98.7%	98.7%	98.2%	98.2%	n.a.	n.a.
’relevant’ F-score	3.8%	3.3%	n.a.	4.3%	3.5%	3.5%
’not relevant’ F-score	98.4%	98.1%	99.1%	98.6%	n.a.	n.a.
’relevant’ percentage	1.8%	1.8%	1.8%	1.8%	1.8%	1.8 %
Time (test)	129 sec	2504 sec	79 sec	66 sec	21 sec	216 sec
Time (training)	1110 sec	1146 sec	395 sec	97664 sec	34 sec	95 sec

Table F.18: Performance results (set 3), based on e-mail messages in the ENRON dataset and using the feature word frequencies. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’.

	MLE (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	1571	1568	1607	1593	16	19
# False Positives	28	33	0	13	1607	1604
# False Negatives	22	20	17	17	0	0
# True Positives	0	2	0	0	16	16
# True Negatives	1571	1566	1607	1593	0	3
Accuracy	96.9%	96.7%	99.0%	98.2%	1.0%	1.2%
Error rate	3.1%	3.3%	1.0%	1.8%	99.0%	98.8%
’relevant’ recall	0.0%	9.1%	0.0%	0.0%	100%	100%
’relevant’ precision	0.0%	5.7%	n.a.	0.0%	1.0%	1.0%
’not relevant’ recall	98.2%	97.9%	100%	99.2%	0.0%	0.2%
’not relevant’ precision	98.6%	98.7%	99.0%	98.9%	n.a.	100%
’relevant’ F-score	n.a.	7.0%	n.a.	n.a.	2.0%	2.0%
’not relevant’ F-score	98.4%	98.3%	99.5%	99.0%	n.a.	0.4%
’relevant’ percentage	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%
Time (test)	129 sec	2726 sec	78 sec	65 sec	19 sec	213 sec
Time (training)	1099 sec	1079 sec	382 sec	107303 sec	35 sec	95 sec

Table F.19: Performance results (set 4), based on e-mail messages in the ENRON dataset and using the feature word frequencies. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’.

	MLE (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	1582	1547	1599	1586	22	22
# False Positives	17	52	0	13	1599	1599
# False Negatives	22	22	25	25	0	0
# True Positives	0	0	0	0	22	22
# True Negatives	1582	1547	1599	1586	0	0
Accuracy	97.6%	95.4%	98.5%	97.7%	1.4%	1.4%
Error rate	2.4%	4.6%	1.5%	2.3%	98.6%	98.6%
’relevant’ recall	0.0%	0.0%	0.0%	0.0%	100%	100%
’relevant’ precision	0.0%	0.0%	n.a.	0.0%	1.4%	1.4%
’not relevant’ recall	98.9%	96.7%	100%	99.2%	0.0%	0.0%
’not relevant’ precision	98.6%	98.6%	98.5%	98.4%	n.a.	n.a.
’relevant’ F-score	n.a.	n.a.	n.a.	n.a.	2.8%	2.8%
’not relevant’ F-score	98.7%	97.6%	99.2%	98.8%	n.a.	n.a.
’relevant’ percentage	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%
Time (test)	129 sec	2251 sec	67 sec	67 sec	21 sec	219 sec
Time (training)	1574 sec	1083 sec	412 sec	11793 sec	37 sec	97 sec

Table F.20: Performance results (set 5), based on e-mail messages in the ENRON dataset and using the feature word frequencies. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’.

TREC dataset

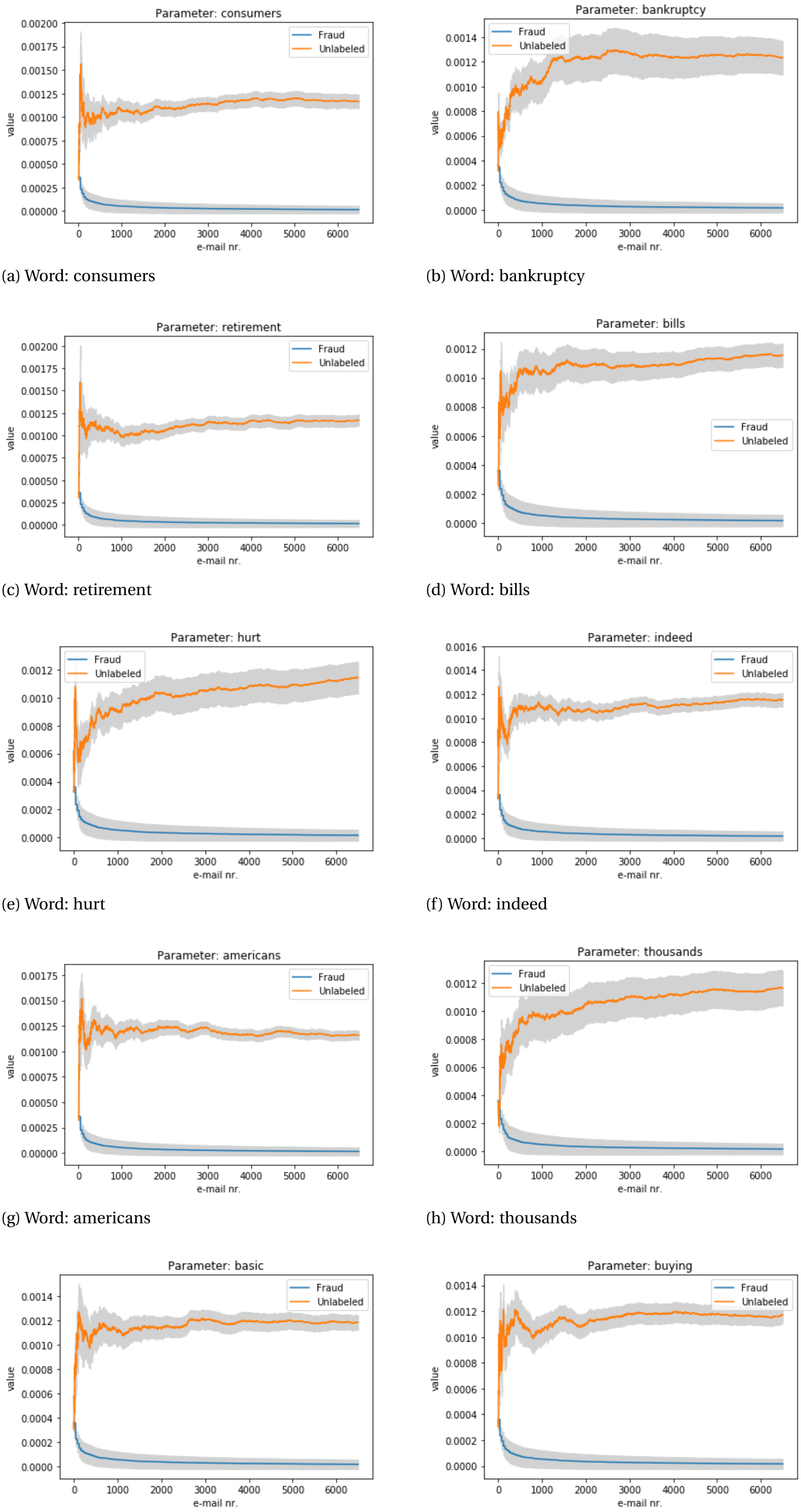
	ML (GM)	BE (GM)	ML (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	14022	13732	9760	14208	6421
# False Positives	39	33	4834	379	1
# False Negatives	590	887	406	64	8229
# True Positives	9030	8733	9562	9556	1391
# True Negatives	4992	4998	198	4652	5030
Accuracy	95.7%	93.7%	65.1%	97.0%	43.8%
Error rate	4.3%	6.3%	34.9%	3.0%	56.2%
’relevant’ recall	93.9%	90.8%	95.9%	99.3%	14.5%
’relevant’ precision	99.6%	99.6%	66.4%	96.2%	99.9%
’not relevant’ recall	99.2%	99.3%	3.9%	92.5%	100%
’not relevant’ precision	89.4%	84.9%	32.8%	98.6%	37.9%
’relevant’ F-score	96.7%	95.0%	78.5%	97.7%	25.3%
’not relevant’ F-score	94.0%	91.5%	7.0%	95.5%	55.0%
’relevant’ percentage	65.7%	65.7%	65.7%	65.7%	65.7%
Time (test)	52940 sec	158426 sec	3304 sec	962 sec	1596 sec
Time (training)	99722 sec	655 sec	20326 sec	634 sec	2178 sec

Table F.21: Performance results (set 1), based on e-mail messages in the TREC dataset and using the feature word frequencies. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’. The performance results of the discriminative model based on Bayesian estimation are not available (the model took to long too compute).

F.2. Model word occurrences

F.2.1. Parameter analysis

ENRON



(a) Word: consumers

(b) Word: bankruptcy

(c) Word: retirement

(d) Word: bills

(e) Word: hurt

(f) Word: indeed

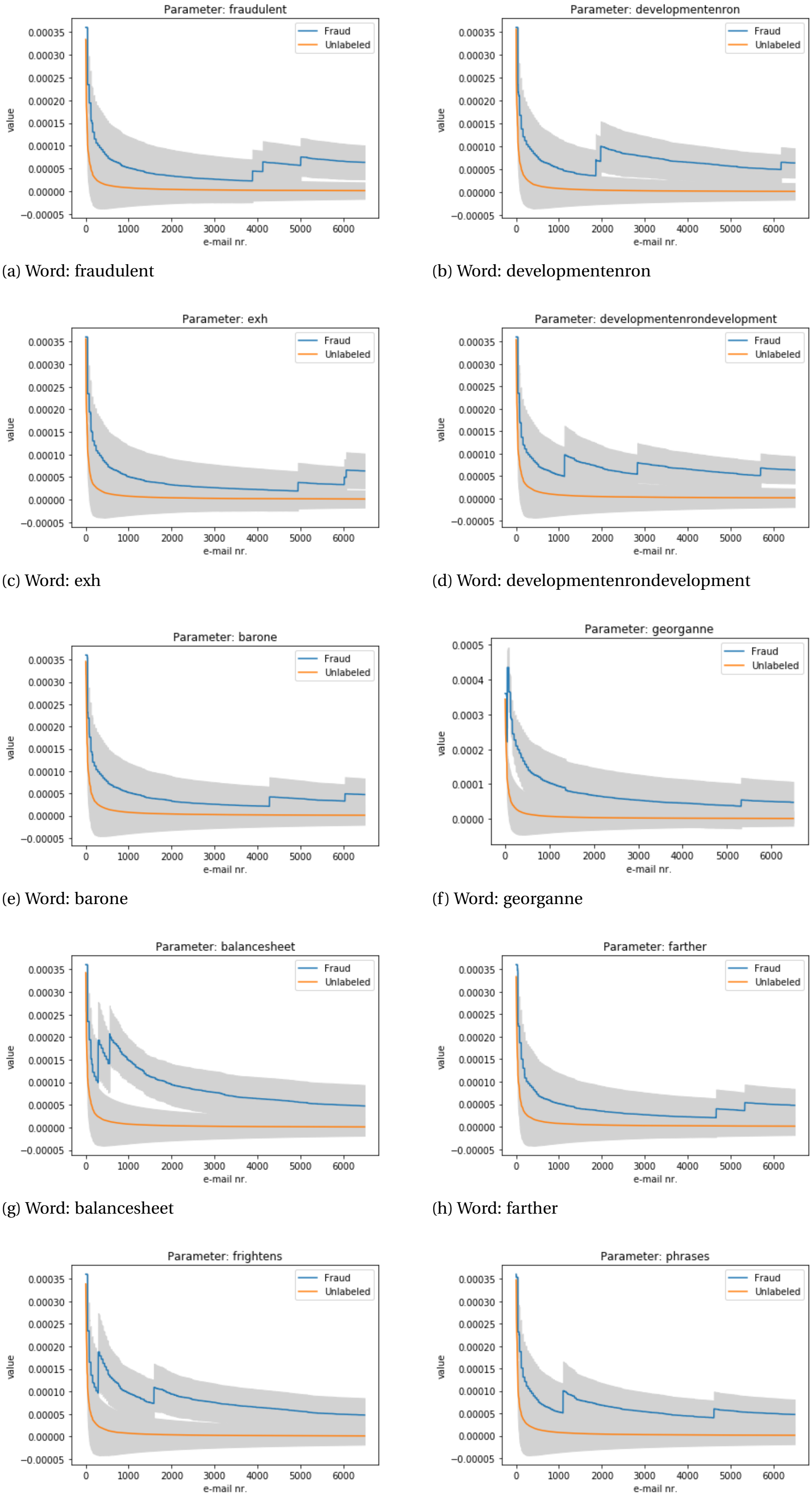
(g) Word: americans

(h) Word: thousands

(i) Word: basic

(j) Word: buying

Figure F.11: Training process of model based on word occurrences for words identifiable as unlabeled (ENRON).



(a) Word: fraudulent

(b) Word: developmentenron

(c) Word: exh

(d) Word: developmentenrondevelopment

(e) Word: barone

(f) Word: georganne

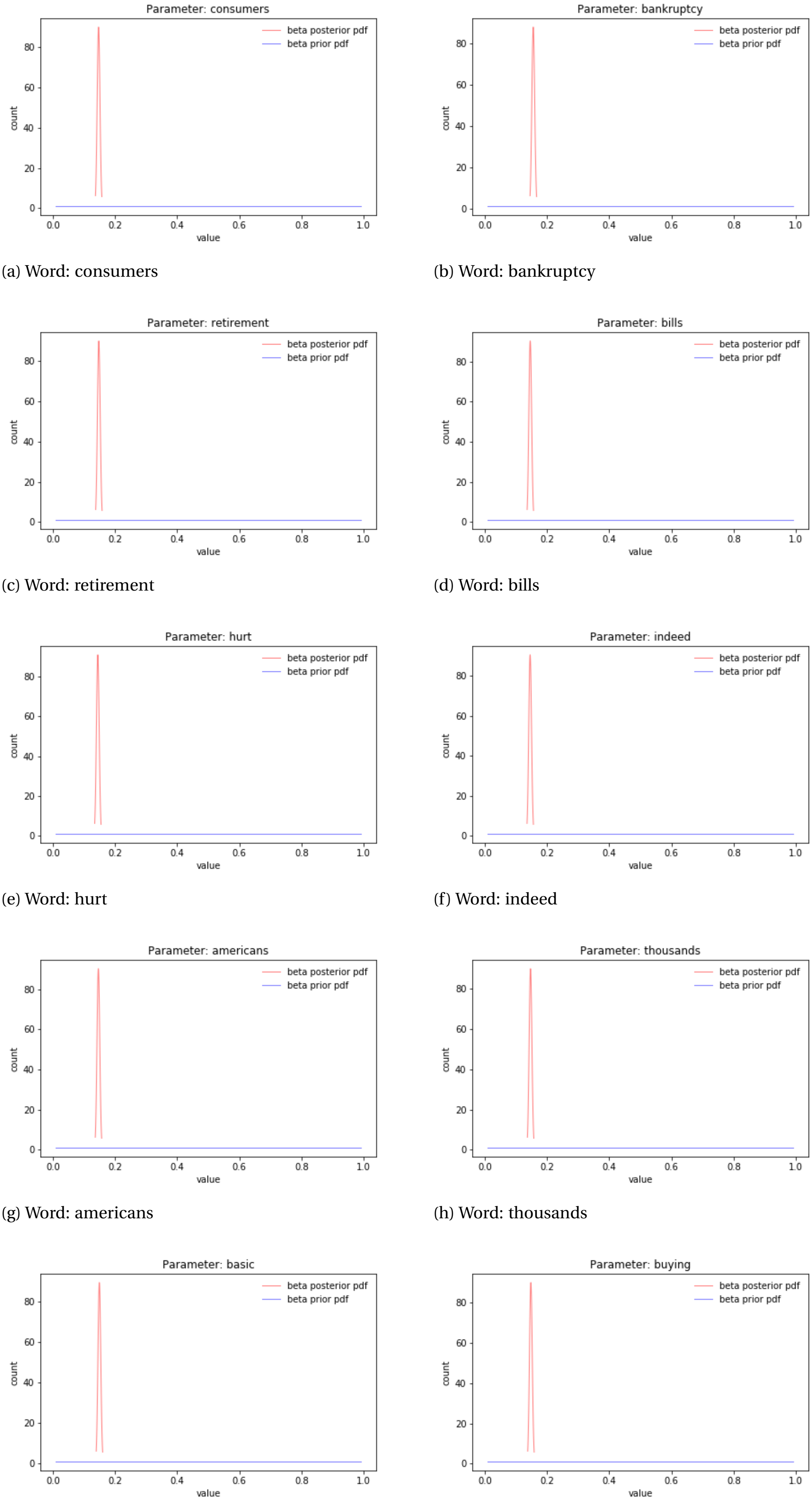
(g) Word: balancesheet

(h) Word: farther

(i) Word: frightens

(j) Word: phrases

Figure F.12: Training process of model based on word occurrences for words identifiable as relevant (ENRON).



(a) Word: consumers

(b) Word: bankruptcy

(c) Word: retirement

(d) Word: bills

(e) Word: hurt

(f) Word: indeed

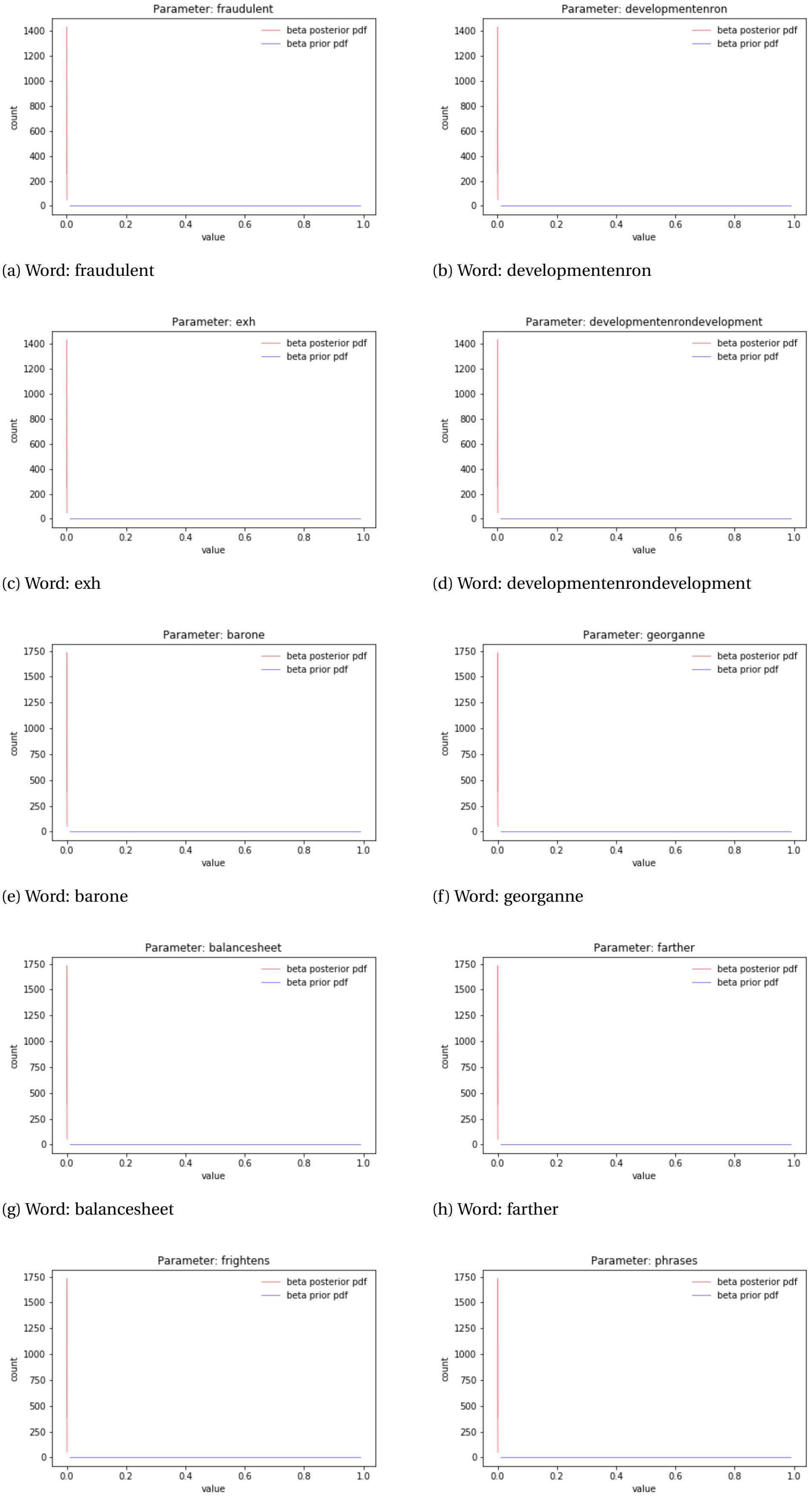
(g) Word: americans

(h) Word: thousands

(i) Word: basic

(j) Word: buying

Figure F.13: Training process generative model for the feature of word occurrences with Bayesian estimation words identifiable as not relevant (ENRON dataset). The two lines indicate the prior and posterior marginal distribution of the corresponding word.



(a) Word: fraudulent

(b) Word: developmentenron

(c) Word: exh

(d) Word: developmentenrondevelopment

(e) Word: barone

(f) Word: georganne

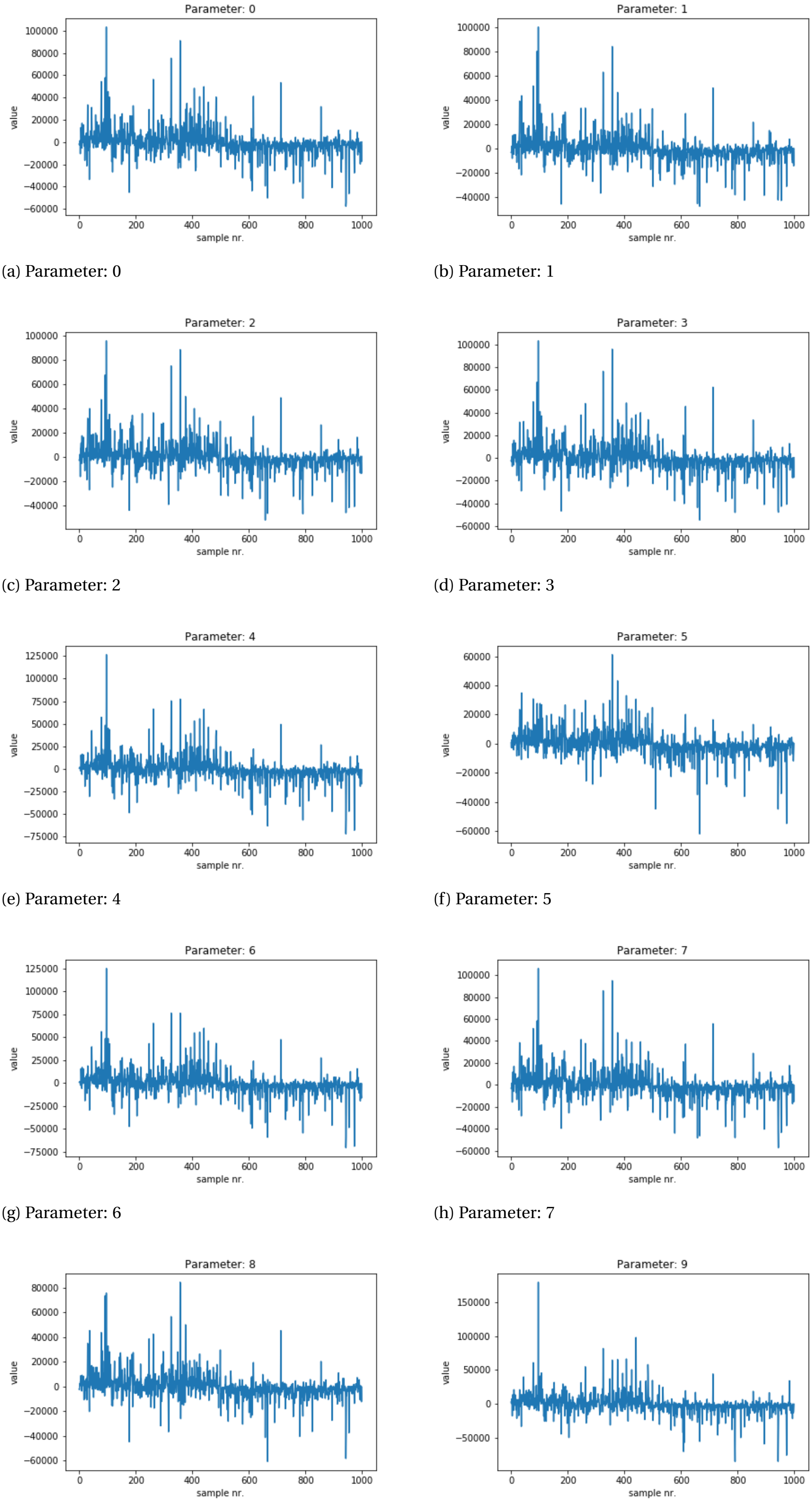
(g) Word: balancesheet

(h) Word: farther

(i) Word: frightens

(j) Word: phrases

Figure F.14: Training process generative model for the feature of word occurrences with Bayesian estimation words identifiable as relevant (ENRON dataset). The two lines indicate the prior and posterior marginal distribution of the corresponding word.



(a) Parameter: 0

(b) Parameter: 1

(c) Parameter: 2

(d) Parameter: 3

(e) Parameter: 4

(f) Parameter: 5

(g) Parameter: 6

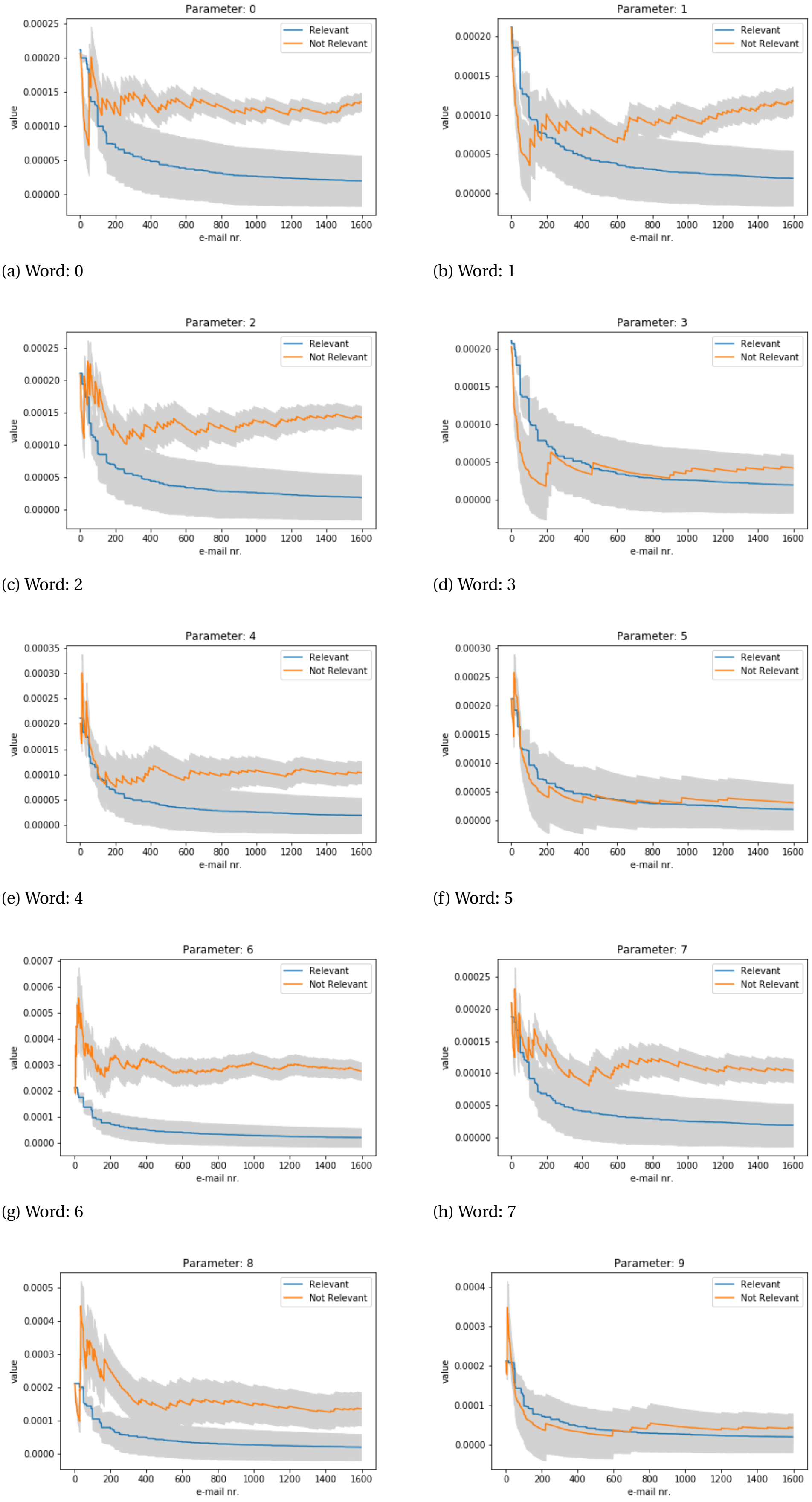
(h) Parameter: 7

(i) Parameter: 8

(j) Parameter: 9

Figure F.15: Training process discriminative model for the feature of word occurrences with Bayesian estimation (ENRON dataset).The blue line indicates the sample values on which the parameter value is based. The titles of each plot are numbered because it is now 100% known to which word the parameter used by the package is related.

confidential dataset



(a) Word: 0

(b) Word: 1

(c) Word: 2

(d) Word: 3

(e) Word: 4

(f) Word: 5

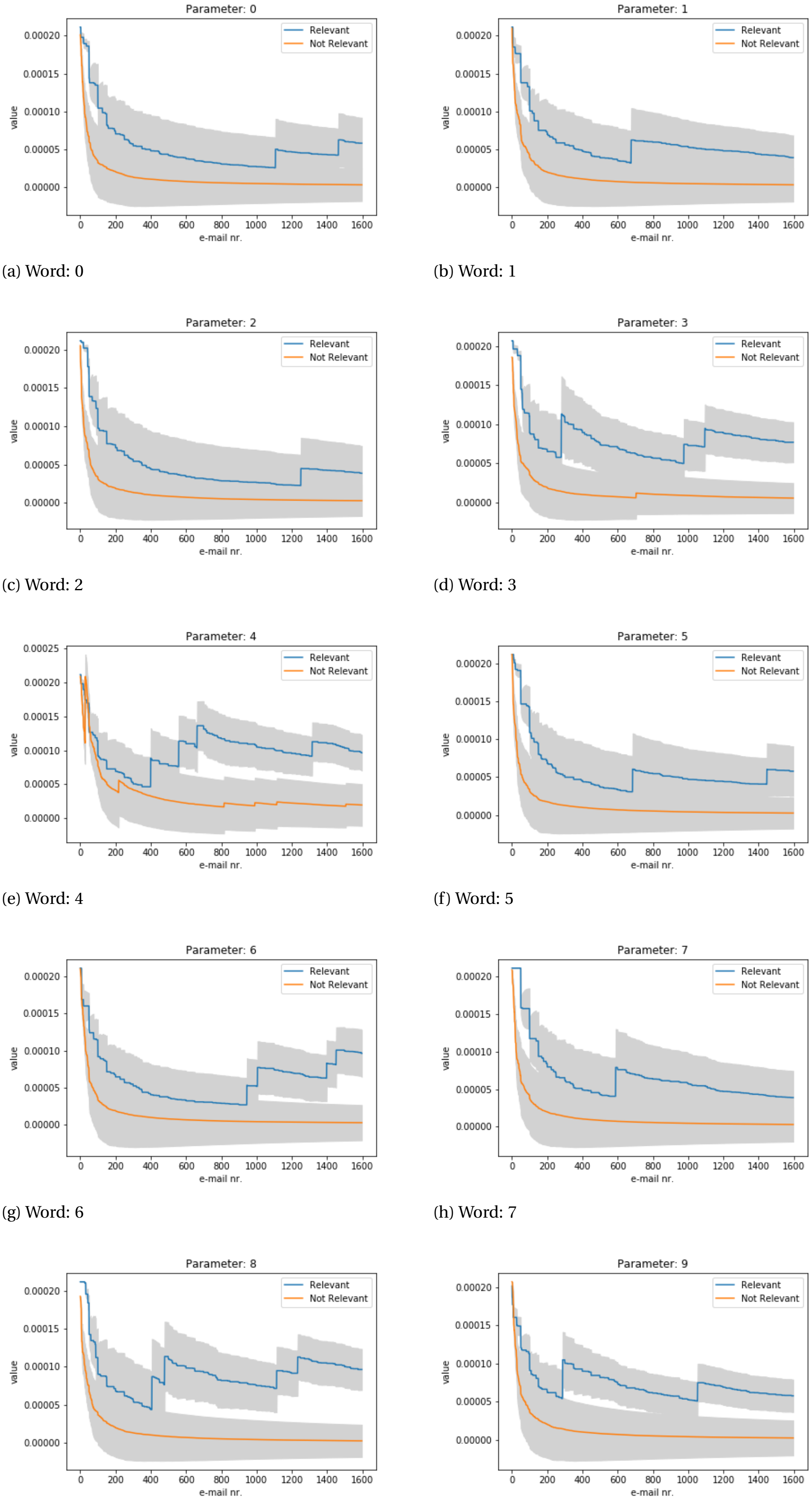
(g) Word: 6

(h) Word: 7

(i) Word: 8

(j) Word: 9

Figure F.16: Training process generative model for the feature of word occurrences with MLE for words identifiable as not relevant (confidential dataset). The words are numbered due to the confidentiality of the dataset.



(a) Word: 0

(b) Word: 1

(c) Word: 2

(d) Word: 3

(e) Word: 4

(f) Word: 5

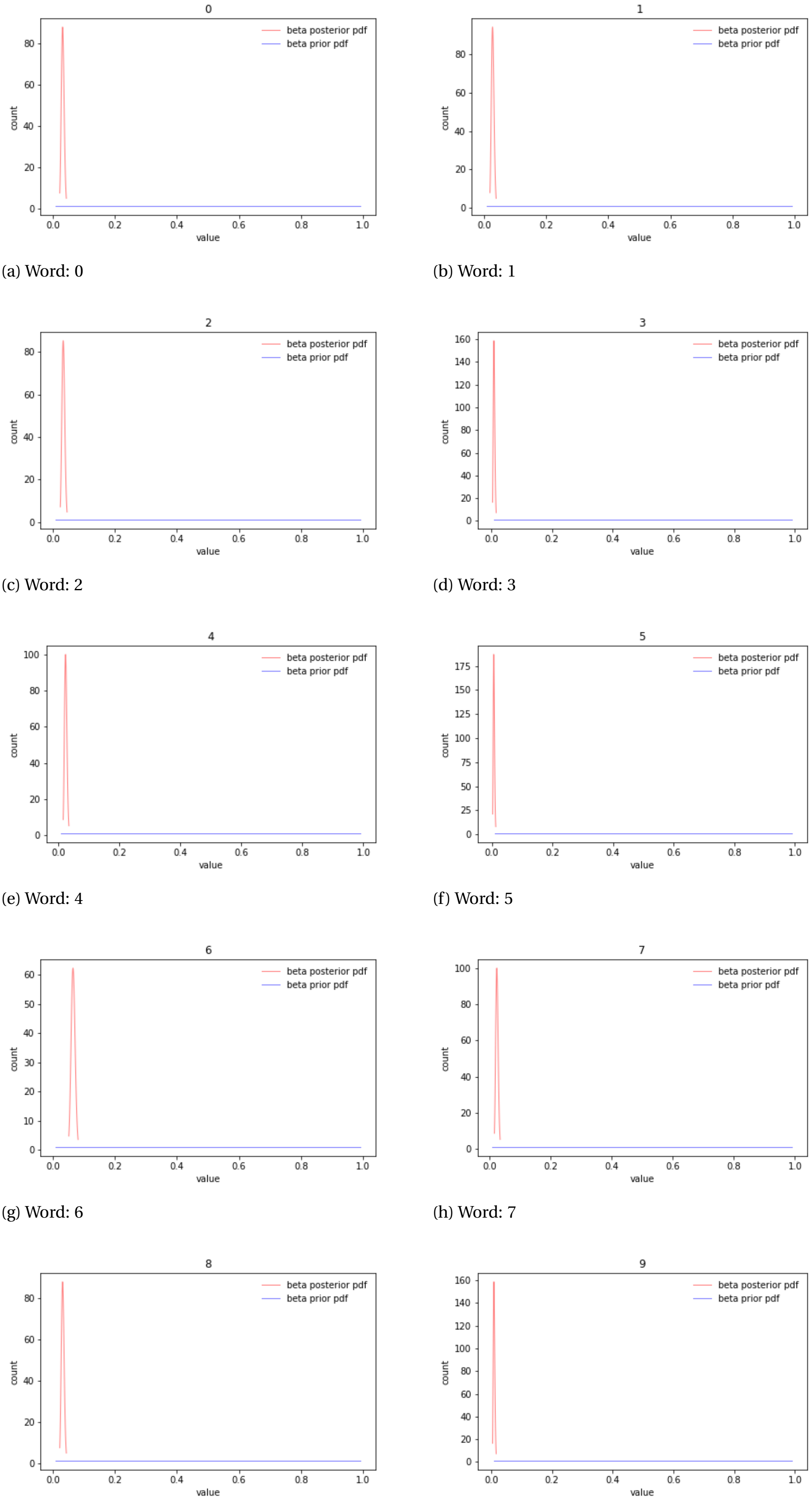
(g) Word: 6

(h) Word: 7

(i) Word: 8

(j) Word: 9

Figure F.17: Training process generative model for the feature of word occurrences with MLE for words identifiable as relevant (confidential dataset). The words are numbered due to the confidentiality of the dataset.



(a) Word: 0

(b) Word: 1

(c) Word: 2

(d) Word: 3

(e) Word: 4

(f) Word: 5

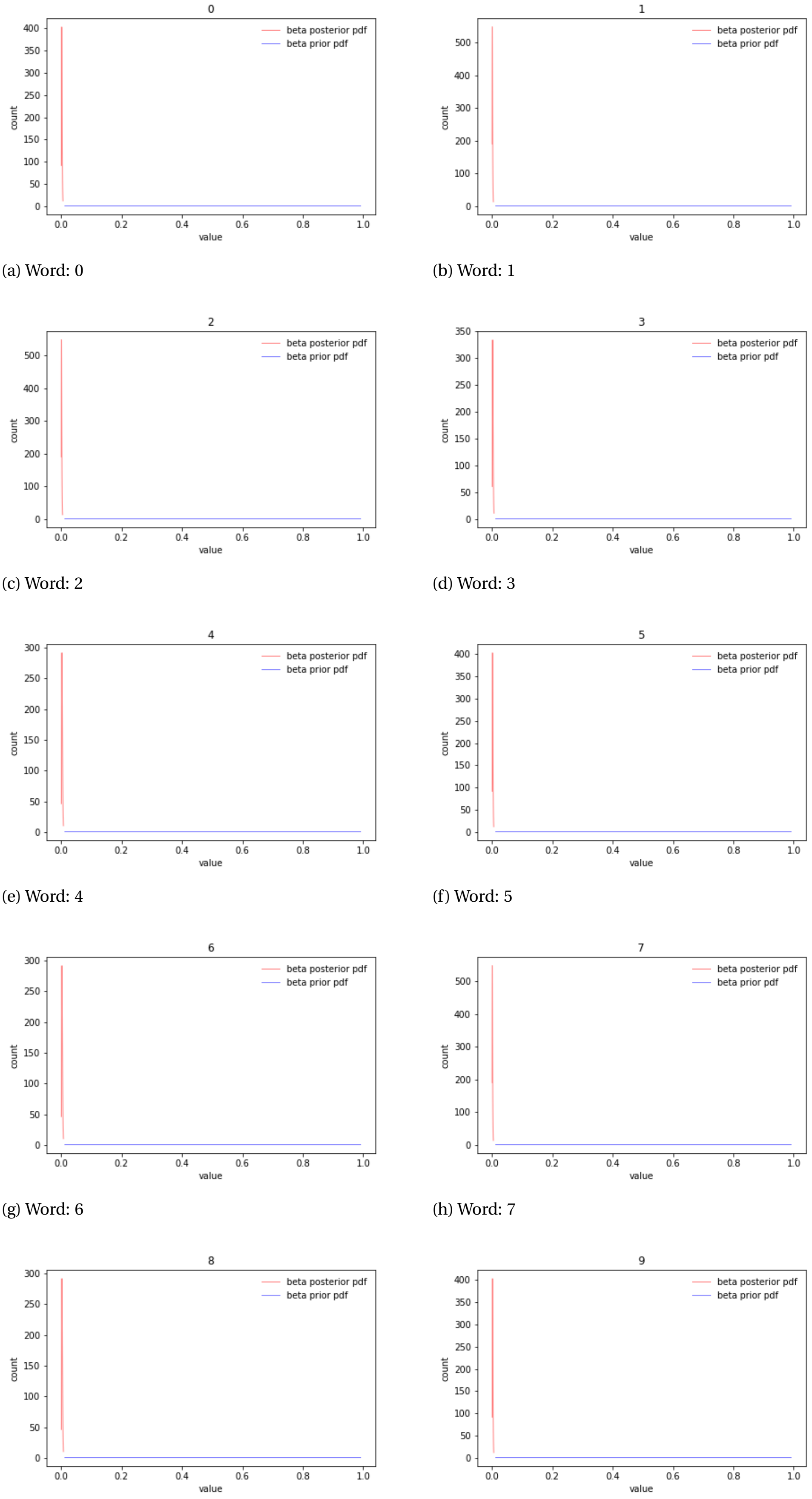
(g) Word: 6

(h) Word: 7

(i) Word: 8

(j) Word: 9

Figure F.18: Training process generative model for the feature of word occurrences with Bayesian estimation words identifiable as not relevant (confidential dataset). The two lines indicate the prior and posterior marginal distribution of the corresponding word. The words are numbered due to the confidentiality of the dataset.



(a) Word: 0

(b) Word: 1

(c) Word: 2

(d) Word: 3

(e) Word: 4

(f) Word: 5

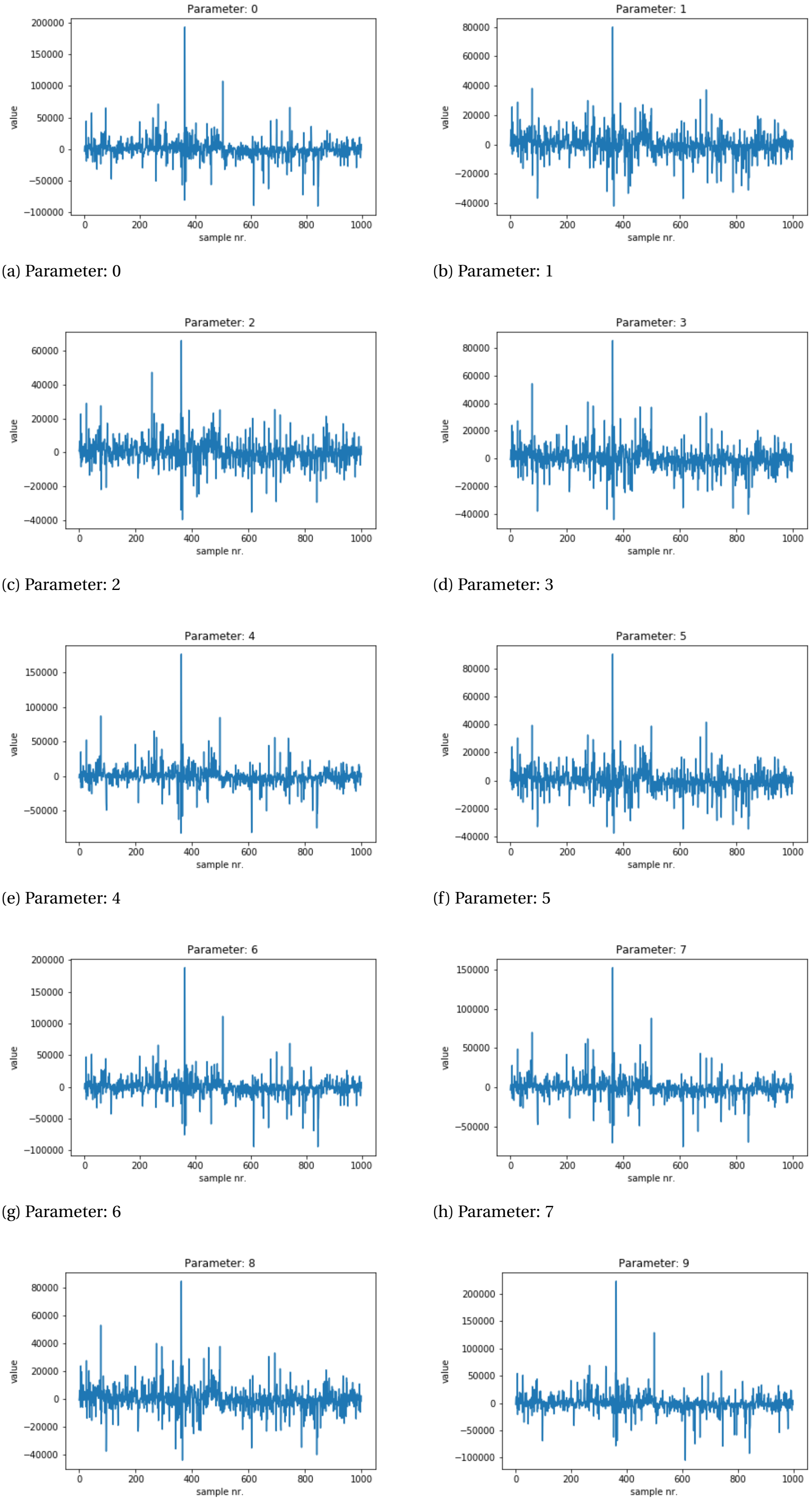
(g) Word: 6

(h) Word: 7

(i) Word: 8

(j) Word: 9

Figure F.19: Training process generative model for the feature of word occurrences with Bayesian estimation words identifiable as relevant (confidential dataset). The two lines indicate the prior and posterior marginal distribution of the corresponding word. The words are numbered due to the confidentiality of the dataset.



(a) Parameter: 0

(b) Parameter: 1

(c) Parameter: 2

(d) Parameter: 3

(e) Parameter: 4

(f) Parameter: 5

(g) Parameter: 6

(h) Parameter: 7

(i) Parameter: 8

(j) Parameter: 9

Figure F.20: Training process discriminative model with Bayesian estimation for word occurrences (confidential dataset). The blue line indicates the sample values on which the parameter value is based. As can be noted the sample values show much difference.

F.2.2. Results classification

confidential dataset

	ML (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	359	32	367	360	241	32
# False Positives	5	367	0	13	137	367
# False Negatives	32	0	32	26	18	0
# True Positives	0	32	0	6	14	32
# True Negatives	359	0	367	354	227	0
Accuracy	90.7%	8.0%	92.0%	90.2%	60.9%	8.0%
Error rate	9.3%	92.0%	8.0%	9.8%	29.1%	92.0%
’relevant’ recall	0.0%	100%	0.0%	18.8%	43.8%	100%
’relevant’ precision	0.0%	8.0%	n.a.	31.6%	9.3%	8.0%
’not relevant’ recall	98.6%	0.0%	100%	96.5%	62.4%	0.0%
’not relevant’ precision	91.8%	n.a.	92.0%	93.2%	92.7%	n.a.
’relevant’ F-score	n.a.	14.8%	n.a.	n.a.	n.a.	16.5%
’not relevant’ F-score	95.1%	n.a.	95.8%	94.8%	74.6%	n.a.
’relevant’ percentage	8.0%	8.0%	8.0%	8.0%	8.0%	8.0%
Time (test)	4135 sec	560 sec	75 sec	35 sec	177 sec	161 sec
Time (training)	416 sec	429 sec	140 sec	19065 sec	48 sec	48 sec

Table F.22: Performance results (set 1), based on the feature word occurrences of e-mail messages in a confidential dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’.

	MLE (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	370	20	379	362	258	20
# False Positives	1	379	0	24	126	379
# False Negatives	20	0	20	13	7	0
# True Positives	0	20	0	7	13	20
# True Negatives	370	0	379	355	245	0
Accuracy	94.6%	5.0%	95.0%	90.7%	66.0%	5.0%
Error rate	5.4%	95.0%	5.0%	9.3%	34.0%	95.0%
’relevant’ recall	0.0%	100%	0.0%	35.0%	65.0%	100%
’relevant’ precision	0.0%	5.0%	n.a.	22.6%	94.0%	5.0%
’not relevant’ recall	99.7%	0.0%	100%	93.7%	66.0%	0.0%
’not relevant’ precision	94.9%	n.a.	95.0%	96.5%	97.2%	n.a.
’relevant’ F-score	n.a.	9.5%	n.a.	27.5%	76.9%	9.5%
’not relevant’ F-score	97.2%	n.a.	97.4%	95.1%	78.6%	n.a.
’relevant’ percentage	5.0%	5.0%	5.0%	5.0%	5.0%	5.0%
Time (test)	4123 sec	567 sec	81 sec	35 sec	176 sec	161 sec
Time (training)	421 sec	449 sec	139 sec	19489 sec	50 sec	49 sec

Table F.23: Performance results (set 2), based on the feature word occurrences of e-mail messages in a confidential dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’.

	MLE (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	361	31	368	365	225	31
# False Positives	4	368	0	12	162	368
# False Negatives	31	0	31	22	9	0
# True Positives	0	31	0	9	22	31
# True Negatives	361	0	368	356	203	0
Accuracy	91.2%	7.8%	92.2%	91.5%	56.8%	7.8%
Error rate	8.8%	92.2%	7.8%	8.5%	43.2%	92.2%
’relevant’ recall	0.0%	100%	0.0%	29.0%	71.0%	100%
’relevant’ precision	0.0%	7.8%	n.a.	42.9%	12.0%	7.8%
’not relevant’ recall	98.9%	0.0%	100%	96.7%	55.6%	0.0%
’not relevant’ precision	92.1%	n.a.	92.2%	94.2%	95.8%	n.a
’relevant’ F-score	n.a.	14.5%	n.a.	34.6%	20.5%	14.5%
’not relevant’ F-score	95.4%	n.a.	95.9%	95.4%	70.4%	n.a.
’relevant’ percentage	7.8%	7.8%	7.8%	7.8%	7.8%	7.8%
Time (test)	4255 sec	577 sec	65 sec	33 sec	177 sec	159 sec
Time (training)	440 sec	453 sec	141 sec	19643 sec	50 sec	51 sec

Table F.24: Performance results (set 3), based on the feature word occurrences of e-mail messages in a confidential dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’.

	MLE (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	364	22	377	367	226	22
# False Positives	5	377	0	14	154	377
# False Negatives	22	0	22	18	11	0
# True Positives	0	22	0	4	11	22
# True Negatives	364	0	377	363	215	0
Accuracy	93.1%	5.5%	94.5%	92.0%	57.8%	5.5%
Error rate	6.9%	94.5%	5.5%	8.0%	42.2%	94.5%
’relevant’ recall	0.0%	100%	0.0%	18.2%	50.0%	100%
’relevant’ precision	0.0%	5.5%	n.a.	22.2%	6.7%	5.5%
’not relevant’ recall	98.6%	0.0%	100%	96.3%	58.3%	0.0%
’not relevant’ precision	94.3%	n.a.	94.5%	95.3%	95.1%	n.a.
’relevant’ F-score	n.a.	10.4%	n.a.	20.0%	11.8%	10.4%
’not relevant’ F-score	96.4%	n.a.	97.2%	95.8%	72.3%	n.a.
’relevant’ percentage	5.5%	5.5%	5.5%	5.5%	5.5%	5.5%
Time (test)	4140 sec	569 sec	80 sec	34 sec	175 sec	161 sec
Time (training)	423 sec	431 sec	138 sec	19639 sec	49 sec	49 sec

Table F.25: Performance results (set 4), based on the feature word occurrences of e-mail messages in a confidential dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’.

	MLE (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	358	31	368	367	332	31
# False Positives	4	368	0	7	149	368
# False Negatives	32	0	31	25	12	0
# True Positives	0	31	0	6	19	31
# True Negatives	358	0	368	361	213	0
Accuracy	91.1%	7.8%	92.2%	92.0%	59.0%	7.8%
Error rate	8.9%	92.2%	7.8%	8.0%	41.0%	92.2%
’relevant’ recall	0.0%	100%	0.0%	19.4%	61.3%	100%
’relevant’ precision	0.0%	7.8%	n.a.	46.2%	11.3%	7.8%
’not relevant’ recall	98.9%	0.0%	100%	98.1%	58.7%	0.0%
’not relevant’ precision	92.0%	n.a.	92.2%	93.5%	94.7%	n.a.
’relevant’ F-score	n.a.	14.5%	n.a.	27.3%	19.1%	14.5%
’not relevant’ F-score	95.3%	n.a.	95.9%	95.7%	72.5%	n.a.
’relevant’ percentage	7.8%	7.8%	7.8%	7.8%	7.8%	7.8%
Time (test)	4277 sec	582 sec	69 sec	34 sec	176 sec	159 sec
Time (training)	437 sec	443 sec	141 sec	19707 sec	51 sec	51 sec

Table F.26: Performance results (set 5), based on the feature word occurrences of e-mail messages in a confidential dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’.

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	171	0	13
$10^2 > B > 10^{\frac{3}{2}}$	0	14	0	2
$10^{\frac{3}{2}} > B > 10$	0	14	0	1
$10 > B > 10^{\frac{1}{2}}$	0	20	0	1
$10^{\frac{1}{2}} > B > 1$	0	8	0	1
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	0	0	25	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	10	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	3	0	30	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	1	0	7	0
$10^{-2} > B$	10	0	65	0

Table F.27: Bayes Factor results word occurrences for the generative model with maximum likelihood estimation based on the top 1000 words (set 1)

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	171	0	13
$10^2 > B > 10^{\frac{3}{2}}$	0	14	0	2
$10^{\frac{3}{2}} > B > 10$	0	14	0	1
$10 > B > 10^{\frac{1}{2}}$	0	20	0	1
$10^{\frac{1}{2}} > B > 1$	0	8	0	1
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	0	0	25	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	10	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	3	0	30	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	1	0	7	0
$10^{-2} > B$	10	0	65	0

Table F.28: Bayes Factor results word occurrences for the generative model with maximum likelihood estimation based on the top 1000 words (set 2)

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	171	0	13
$10^2 > B > 10^{\frac{3}{2}}$	0	14	0	2
$10^{\frac{3}{2}} > B > 10$	0	14	0	1
$10 > B > 10^{\frac{1}{2}}$	0	20	0	1
$10^{\frac{1}{2}} > B > 1$	0	8	0	1
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	0	0	25	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	10	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	3	0	30	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	1	0	7	0
$10^{-2} > B$	10	0	65	0

Table F.29: Bayes Factor results word occurrences for the generative model with maximum likelihood estimation based on the top 1000 words (set 3)

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	171	0	13
$10^2 > B > 10^{\frac{3}{2}}$	0	14	0	2
$10^{\frac{3}{2}} > B > 10$	0	14	0	1
$10 > B > 10^{\frac{1}{2}}$	0	20	0	1
$10^{\frac{1}{2}} > B > 1$	0	8	0	1
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	0	0	25	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	10	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	3	0	30	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	1	0	7	0
$10^{-2} > B$	10	0	65	0

Table F.30: Bayes Factor results word occurrences for the generative model with maximum likelihood estimation based on the top 1000 words (set 4)

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	171	0	13
$10^2 > B > 10^{\frac{3}{2}}$	0	14	0	2
$10^{\frac{3}{2}} > B > 10$	0	14	0	1
$10 > B > 10^{\frac{1}{2}}$	0	20	0	1
$10^{\frac{1}{2}} > B > 1$	0	8	0	1
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	0	0	25	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	10	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	3	0	30	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	1	0	7	0
$10^{-2} > B$	10	0	65	0

Table F.31: Bayes Factor results word occurrences for the generative model with maximum likelihood estimation based on the top 1000 words (set 5)

ENRON dataset

	ML (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	1552	28	1596	1587	1135	28
# False Positives	44	1596	0	12	478	1596
# False Negatives	27	0	28	25	10	0
# True Positives	0	28	0	3	17	28
# True Negatives	1552	0	1596	1584	1118	0
Accuracy	95.6%	1.7%	98.3%	97.7%	69.9%	1.7%
Error rate	4.4%	98.3%	1.7%	2.3%	30.1%	98.3%
’relevant’ recall	0.0%	100%	0.0%	10.7%	63.0%	100%
’relevant’ precision	0.0%	1.7%	n.a.	20.0%	3.4%	1.7%
’not relevant’ recall	97.2%	0.0%	100%	99.2%	70.1%	0.0%
’not relevant’ precision	98.3%	n.a.	98.3%	98.4%	99.1%	n.a.
’relevant’ F-score	n.a.	3.3%	n.a.	13.9%	6.5%	3.3%
’not relevant’ F-score	97.7%	n.a.	99.1%	98.8%	82.1%	n.a.
’relevant’ percentage	1.7%	1.7%	1.7%	1.7%	1.7%	1.7%
Time (test)	20081 sec	2311 sec	79 sec	64 sec	410 sec	211sec
Time (training)	1896 sec	1499 sec	394 sec	52578 sec	87 sec	88 sec

Table F.32: Performance results (set 1), based on the feature word occurrences of e-mail messages in the ENRON dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’.

	MLE (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	1554	25	1599	1580	1059	25
# False Positives	45	1599	0	21	550	1599
# False Negatives	23	0	25	23	13	0
# True Positives	0	25	0	2	10	25
# True Negatives	1554	0	1599	1578	1049	0
Accuracy	95.8%	1.5%	98.5%	97.3%	65.3%	1.5%
Error rate	4.2%	98.5%	1.5%	2.7%	34.7%	98.5%
’relevant’ recall	0.0%	100%	0.0%	8.0%	43.5%	100%
’relevant’ precision	0.0%	1.5%	n.a.	8.7%	1.8%	1.5%
’not relevant’ recall	97.2%	0.0%	100%	98.7%	65.6%	0.0%
’not relevant’ precision	98.5%	n.a.	98.5%	98.6%	98.8%	n.a.
’relevant’ F-score	n.a.	3.0%	n.a.%	8.3%	3.5%	3.0%
’not relevant’ F-score	97.8%	n.a.	99.2%	98.6%	78.8%	n.a.
’relevant’ percentage	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%
Time (test)	20318 sec	2304 sec	78 sec	63 sec	411 sec	246 sec
Time (training)	1369 sec	1269 sec	387 sec	50980 sec	90 sec	89 sec

Table F.33: Performance results (set 2), based on the feature word occurrences of e-mail messages in the ENRON dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’.

	MLE (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	1554	25	1594	1587	1161	30
# False Positives	45	1599	0	10	450	1594
# False Negatives	23	0	30	27	13	0
# True Positives	0	25	0	3	17	30
# True Negatives	1554	0	1594	1584	1144	0
Accuracy	95.8%	1.5%	98.2%	97.7%	71.5%	1.8%
Error rate	4.2%	98.5%	1.8%	2.3%	28.5%	98.2%
’relevant’ recall	0.0%	100%	0.0%	10.0%	56.7%	100%
’relevant’ precision	0.0%	1.5%	n.a.%	23.1%	3.6%	1.8%
’not relevant’ recall	97.2%	0.0%	100%	99.4%	71.8%	0.0%
’not relevant’ precision	98.5%	n.a.	98.2%	98.3%	98.9%	n.a.
’relevant’ F-score	n.a.	3.0%	n.a.%	14.0%	6.8%	3.5%
’not relevant’ F-score	97.8%	n.a.	99.1%	98.8%	83.2%	n.a.
’relevant’ percentage	1.8%	1.8%	1.8%	1.8%	1.8%	1.8%
Time (test)	20149 sec	2237 sec	89 sec	66 sec	415 sec	201 sec
Time (training)	1326 sec	1213 sec	389 sec	72402 sec	89 sec	88 sec

Table F.34: Performance results (set 3), based on the feature word occurrences of e-mail messages in the ENRON dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’.

	MLE (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	1551	17	1607	1592	1103	17
# False Positives	43	1607	0	15	508	1607
# False Negatives	30	0	17	17	12	0
# True Positives	0	17	0	0	4	17
# True Negatives	1551	0	1607	1592	1099	0
Accuracy	95.5%	1.0%	99.0%	98.0%	68.0%	1.0%
Error rate	4.5%	99.0%	1.0%	2.0%	32.0%	99.0%
’relevant’ recall	0.0%	100%	0.0%	0.0%	25.0%	100%
’relevant’ precision	0.0%	1.0%	n.a.%	0.0%	0.8%	1.0%
’not relevant’ recall	97.3%	0.0%	100%	99.1%	68.4%	0.0%
’not relevant’ precision	98.1%	n.a.	99.0%	98.9%	98.9%	n.a.
’relevant’ F-score	n.a.	2.0%	n.a.%	n.a.	1.6%	2.0%
’not relevant’ F-score	97.7%	n.a.	99.5%	99.0%	80.9%	n.a.
’relevant’ percentage	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%
Time (test)	20272 sec	2235 sec	99 sec	74 sec	421 sec	235 sec
Time (training)	1493 sec	1210 sec	506 sec	60949 sec	91 sec	89 sec

Table F.35: Performance results (set 4), based on the feature word occurrences of e-mail messages in the ENRON dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’.

	MLE (GM)	BE (GM)	ML (DM)*	BE (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	1564	25	1599	1587	1130	25
# False Positives	43	1599	0	14	480	1599
# False Negatives	16	0	25	23	11	0
# True Positives	0	25	0	2	11	25
# True Negatives	1564	0	1599	1585	1119	0
Accuracy	96.4%	1.5%	98.5%	97.7%	69.7%	1.5%
Error rate	3.6%	98.5%	1.5%	2.3%	30.3%	98.5%
’relevant’ recall	0.0%	100%	0.0%	8.0%	50.0%	100%
’relevant’ precision	0.0%	1.5%	n.a.	12.5%	2.2%	1.5%
’not relevant’ recall	97.3%	0.0%	100%	99.1%	70.0%	0.0%
’not relevant’ precision	99.0%	n.a.	98.5%	98.6%	99.0%	n.a.
’relevant’ F-score	n.a.	3.0%	n.a.	9.8%	4.2%	3.0%
’not relevant’ F-score	98.1%	n.a.	99.2%	98.8%	82.0%	n.a.
’relevant’ percentage	1.5%	1.5%	1.5%	1.5%	1.5%	1.5%
Time (test)	21250 sec	2284 sec	77 sec	74 sec	414 sec	199 sec
Time (training)	1349 sec	1229 sec	483 sec	51788 sec	90 sec	89 sec

Table F.36: Performance results (set 5), based on the feature word occurrences of e-mail messages in the ENRON dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’.

TREC dataset

	ML (GM)	BE (GM)	ML (DM)*	ML (GM)*	BE (GM)*
# Predicted = given	9620	5032	7858	11466	5032
# False Positives	5031	0	2811	22	0
# False Negatives	0	9968	4331	3163	9968
# True Positives	9620	0	5637	6457	0
# True Negatives	0	5032	2221	5009	5032
Accuracy	65.7%	33.5%	52.4%	78.3%	33.5%
Error rate	34.3%	66.5%	47.6%	21.7%	66.5%
’relevant’ recall	100%	0.0%	56.6%	67.1%	0.0%
’relevant’ precision	65.7%	n.a.	66.7%	99.7%	n.a.
’not relevant’ recall	0.0%	100%	44.1%	99.6%	100%
’not relevant’ precision	n.a.	33.5%	33.9%	61.3%	33.5%
’relevant’ F-score	79.3%	n.a.	61.2%	80.2%	n.a.
’not relevant’ F-score	n.a.	50.2%	38.3%	75.9%	50.2%
’relevant’ percentage	65.7%	65.7%	65.7%	65.7%	65.7%
Time (test)	3603 sec	98208 sec	2752 sec	6315 sec	1084 sec
Time (training)	110929 sec	1201 sec	20612 sec	980 sec	985 sec

Table F.37: Performance results (set 1), based on the feature word occurrences of e-mail messages in the TREC dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation). * = based on the top 500 words identified as as best corresponding the category ’relevant’ and the top 500 words identified as best corresponding to the category ’not relevant’. The performance results of the discriminative model based on Bayesian estimation are not available (the model took too long to compute).

F.3. Model length e-mail

F.3.1. Parameter analysis

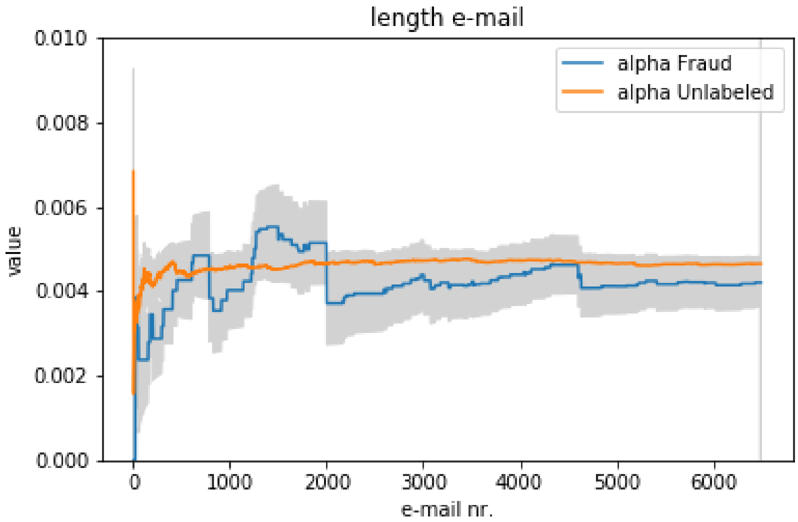


Figure F.21: Training process generative model for feature length e-mail with MLE (ENRON dataset). The two lines indicate how the value of the parameter corresponding to the words are trained. In these figures 'Legit' is the category of the unlabeled e-mails.

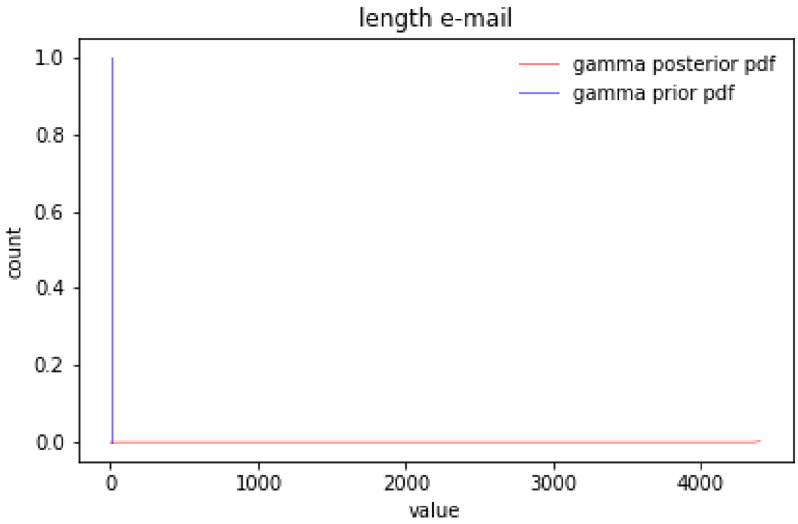


Figure F.22: Training process generative model for feature length e-mail with Bayesian estimation (ENRON dataset). The two lines indicate how the value of the parameter corresponding to the words are trained. In these figures 'Legit' is the category of the unlabeled e-mails.

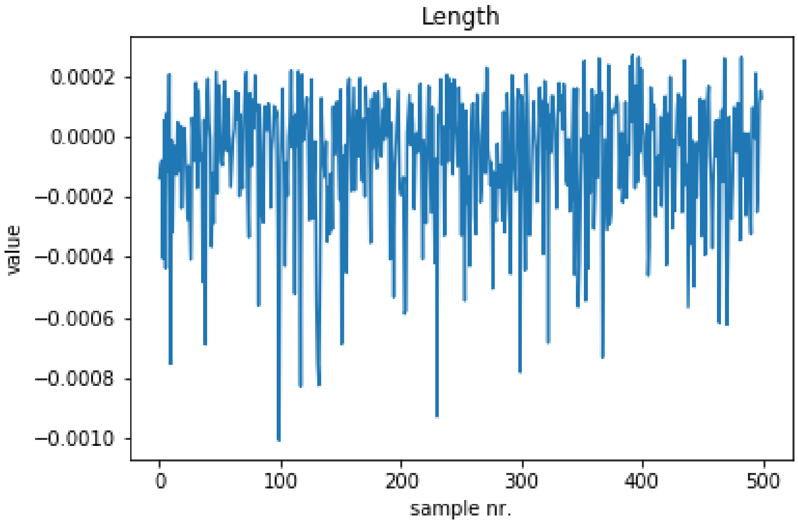


Figure F.23: Training process discriminative model for feature length e-mail with Bayesian estimation (ENRON dataset). The blue line indicates the sample values on which the parameter value is based. As can be seen the sample values are quite close to each other.

F.3.2. Results classification

confidential dataset

	ML (GM)	BE (GM)	ML (DM)	BE (DM)
# Predicted = given	333	32	367	367
# False Positives	39	367	0	0
# False Negatives	27	0	32	32
# True Positives	5	32	0	0
# True Negatives	328	0	367	367
Accuracy	83.5%	8.0%	92.0%	92.0%
Error rate	16.5%	92.0%	8.0%	8.0%
’relevant’ recall	15.6%	100%	0.0%	0.0%
’relevant’ precision	11.4%	8.0%	n.a.	n.a.
’not relevant’ recall	89.4%	0.0%	100%	100%
’not relevant’ precision	92.4%	n.a.	92.0%	92.0%
’relevant’ F-score	13.2%	14.8%	n.a.	n.a.
’not relevant’ F-score	90.9%	n.a.	95.8%	95.8%
’relevant’ percentage	8.0%	8.0%	8.0%	8.0%
Time (test)	6 sec	139 sec	5 sec	16 sec
Time (training)	18 sec	18 sec	17 sec	134 sec

Table F.38: Performance results (set 1), based on the feature length e-mail of e-mail messages in a confidential dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation).

	MLE (GM)	BE (GM)	ML (DM)	BE (DM)
# Predicted = given	318	20	379	379
# False Positives	64	379	0	0
# False Negatives	17	0	20	20
# True Positives	3	20	0	0
# True Negatives	315	0	379	379
Accuracy	79.7%	5.0%	95.0%	95.0%
Error rate	20.3%	95.0%	5.0%	5.0%
’relevant’ recall	15.0%	100%	0.0%	0.0%
’relevant’ precision	4.5%	5.0%	n.a.	n.a.
’not relevant’ recall	83.1%	0.0%	100%	100%
’not relevant’ precision	94.9%	n.a.	95.0%	95.0%
’relevant’ F-score	6.9%	9.5%	n.a.	n.a.
’not relevant’ F-score	88.6%	n.a.	97.4%	97.4%
’relevant’ percentage	5.0%	5.0%	5.0%	5.0%
Time (test)	6 sec	139 sec	5 sec	17 sec
Time (training)	18 sec	18 sec	17 sec	137 sec

Table F.39: Performance results (set 2), based on the feature length e-mail of e-mail messages in a confidential dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation).

	MLE (GM)	BE (GM)	ML (DM)	BE (DM)
# Predicted = given	337	31	368	368
# False Positives	32	368	0	0
# False Negatives	30	0	31	31
# True Positives	1	31	0	0
# True Negatives	336	0	368	368
Accuracy	84.5%	7.8%	92.2%	92.2%
Error rate	15.5%	92.2%	7.8%	7.8%
’relevant’ recall	3.2%	100%	0.0%	0.0%
’relevant’ precision	3.0%	7.8%	n.a.	n.a.
’not relevant’ recall	91.3%	0.0%	100%	100%
’not relevant’ precision	91.8%	n.a.	92.2%	92.2%
’relevant’ F-score	3.1%	14.5%	n.a.	n.a.
’not relevant’ F-score	91.5%	n.a.	95.9%	95.9%
’relevant’ percentage	7.8%	7.8%	7.8%	7.8%
Time (test)	5 sec	140 sec	5 sec	15 sec
Time (training)	19 sec	19 sec	17 sec	131 sec

Table F.40: Performance results (set 3), based on the feature length e-mail of e-mail messages in a confidential dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation).

	MLE (GM)	BE (GM)	ML (DM)	BE (DM)
# Predicted = given	330	22	377	377
# False Positives	50	377	0	0
# False Negatives	19	0	22	22
# True Positives	3	22	0	0
# True Negatives	327	0	377	377
Accuracy	82.7%	5.5%	94.5%	94.5%
Error rate	17.3%	94.5%	5.5%	5.5%
’relevant’ recall	13.6%	100%	0.0%	0.0%
’relevant’ precision	5.7%	5.5%	n.a.	n.a.
’not relevant’ recall	86.7%	0.0%	100%	100%
’not relevant’ precision	94.5%	n.a.	94.5%	94.5%
’relevant’ F-score	8.0%	10.4%	n.a.	n.a.
’not relevant’ F-score	90.4%	n.a.	97.2%	97.2
’relevant’ percentage	5.5%	5.5%	5.5%	5.5%
Time (test)	6 sec	138 sec	5 sec	16 sec
Time (training)	18 sec	18 sec	17 sec	134 sec

Table F.41: Performance results (set 4), based on the feature length e-mail of e-mail messages in a confidential dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation).

	MLE (GM)	BE (GM)	ML (DM)	BE (DM)
# Predicted = given	331	31	368	368
# False Positives	42	368	0	0
# False Negatives	26	0	31	31
# True Positives	5	31	0	0
# True Negatives	326	0	368	368
Accuracy	83.0%	7.8%	92.2%	92.2%
Error rate	17.0%	92.2%	7.8%	7.8%
’relevant’ recall	16.1%	100%	0.0%	0.0%
’relevant’ precision	10.6%	7.8%	n.a.	n.a.
’not relevant’ recall	88.6%	0.0%	100%	100%
’not relevant’ precision	92.6%	n.a.	92.2%	92.2%
’relevant’ F-score	12.8%	14.5%	n.a.	n.a.
’not relevant’ F-score	90.6%	n.a.	95.9%	95.9%
’relevant’ percentage	7.8%	7.8%	7.8%	7.8%
Time (test)	5 sec	139 sec	5 sec	15 sec
Time (training)	18 sec	19 sec	17 sec	131 sec

Table F.42: Performance results (set 5), based on the feature length e-mail of e-mail messages in a confidential dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation).

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	0	0	0
$10^2 > B > 10^{\frac{3}{2}}$	0	0	0	0
$10^{\frac{3}{2}} > B > 10$	0	0	0	0
$10 > B > 10^{\frac{1}{2}}$	0	0	0	0
$10^{\frac{1}{2}} > B > 1$	0	328	0	27
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	5	0	39	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	0	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	0	0	0	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	0	0	0	0
$10^{-2} > B$	0	0	0	0

Table F.43: Bayes Factor results of e-mail lengths for the generative model with maximum likelihood estimation (set 1)

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	0	0	0
$10^2 > B > 10^{\frac{3}{2}}$	0	0	0	0
$10^{\frac{3}{2}} > B > 10$	0	0	0	0
$10 > B > 10^{\frac{1}{2}}$	0	0	0	0
$10^{\frac{1}{2}} > B > 1$	0	315	0	17
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	3	0	64	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	0	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	0	0	0	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	0	0	0	0
$10^{-2} > B$	0	0	0	0

Table F.44: Bayes Factor results of e-mail lengths for the generative model with maximum likelihood estimation (set 2)

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	0	0	0
$10^2 > B > 10^{\frac{3}{2}}$	0	0	0	0
$10^{\frac{3}{2}} > B > 10$	0	0	0	0
$10 > B > 10^{\frac{1}{2}}$	0	0	0	0
$10^{\frac{1}{2}} > B > 1$	0	336	0	30
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	1	0	32	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	0	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	0	0	0	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	0	0	0	0
$10^{-2} > B$	0	0	0	0

Table F.45: Bayes Factor results of e-mail lengths for the generative model with maximum likelihood estimation (set 3)

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	0	0	0
$10^2 > B > 10^{\frac{3}{2}}$	0	0	0	0
$10^{\frac{3}{2}} > B > 10$	0	0	0	0
$10 > B > 10^{\frac{1}{2}}$	0	0	0	0
$10^{\frac{1}{2}} > B > 1$	0	327	0	19
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	3	0	50	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	0	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	0	0	0	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	0	0	0	0
$10^{-2} > B$	0	0	0	0

Table F.46: Bayes Factor results of e-mail lengths for the generative model with maximum likelihood estimation (set 4)

Bayes factor	True Positives	True Negatives	False Positives	False Negatives
$B > 10^2$	0	0	0	0
$10^2 > B > 10^{\frac{3}{2}}$	0	0	0	0
$10^{\frac{3}{2}} > B > 10$	0	0	0	0
$10 > B > 10^{\frac{1}{2}}$	0	0	0	0
$10^{\frac{1}{2}} > B > 1$	0	326	0	26
$B = 1$	0	0	0	0
$1 > B > 10^{-\frac{1}{2}}$	5	0	42	0
$10^{-\frac{1}{2}} > B > 10^{-1}$	0	0	0	0
$10^{-1} > B > 10^{-\frac{3}{2}}$	0	0	0	0
$10^{-\frac{3}{2}} > B > 10^{-2}$	0	0	0	0
$10^{-2} > B$	0	0	0	0

Table F.47: Bayes Factor results of e-mail lengths for the generative model with maximum likelihood estimation (set 5)

ENRON dataset

	ML (GM)	BE (GM)	ML (DM)	BE (DM)
# Predicted = given	1420	27	1596	1596
# False Positives	181	1596	0	0
# False Negatives	23	0	28	28
# True Positives	5	27	0	0
# True Negatives	1415	0	1596	1596
Accuracy	87.4%	1.7%	98.3%	98.3%
Error rate	12.6%	98.3%	1.7%	1.7%
'relevant' recall	17.9%	100%	0.0%	0.0%
'relevant' precision	2.7%	1.7%	n.a.	n.a.
'not relevant' recall	88.7%	0.0%	100%	100%
'not relevant' precision	98.4%	n.a.	98.3%	98.3%
'relevant' F-score	4.7%	3.3%	n.a.	n.a.
'not relevant' F-score	93.3%	n.a.	99.1%	99.1%
'relevant' percentage	1.7%	1.7%	1.7%	1.7%
Time (test)	11 sec	188 sec	10 sec	26 sec
Time (training)	35 sec	37 sec	36 sec	179 sec

Table F.48: Performance results (set 1), based on the feature length e-mail of e-mail messages in the ENRON dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation).

	MLE (GM)	BE (GM)	ML (DM)	BE (DM)
# Predicted = given	187	23	1599	1599
# False Positives	1433	1599	0	0
# False Negatives	4	0	25	25
# True Positives	21	23	0	0
# True Negatives	166	0	1599	1599
Accuracy	11.5%	1.4%	98.5%	98.5%
Error rate	88.5%	98.6%	1.5%	1.5%
'relevant' recall	84.0%	100%	0.0%	0.0%
'relevant' precision	1.4%	1.4%	n.a.	n.a.
'not relevant' recall	10.4%	0.0%	100%	100%
'not relevant' precision	97.6%	n.a.	98.5%	98.5%
'relevant' F-score	2.8%	2.8%	n.a.	n.a.
'not relevant' F-score	18.8%	n.a.	99.2%	99.2%
'relevant' percentage	1.5%	1.5%	1.5%	1.5%
Time (test)	13 sec	182 sec	9 sec	26 sec
Time (training)	37 sec	40 sec	40 sec	169 sec

Table F.49: Performance results (set 2), based on the feature length e-mail of e-mail messages in the ENRON dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation).

	MLE (GM)	BE (GM)	ML (DM)	BE (DM)
# Predicted = given	1423	30	1594	1594
# False Positives	176	1594	0	0
# False Negatives	25	0	30	30
# True Positives	5	30	0	0
# True Negatives	1418	0	1594	1594
Accuracy	87.6%	1.8%	98.2%	98.2%
Error rate	12.4%	98.2%	1.8%	1.8%
’relevant’ recall	16.7%	100%	0.0%	0.0%
’relevant’ precision	2.8%	1.8%	n.a.	n.a.
’not relevant’ recall	89.0%	0.0%	100%	100%
’not relevant’ precision	98.3%	n.a.	98.2%	98.2%
’relevant’ F-score	4.8%	3.5%	n.a.	n.a.
’not relevant’ F-score	93.4%	n.a.	99.1%	99.1%
’relevant’ percentage	1.8%	1.8%	1.8%	1.8%
Time (test)	13 sec	187 sec	12 sec	29 sec
Time (training)	37 sec	40 sec	39 sec	177 sec

Table F.50: Performance results (set 3), based on the feature length e-mail of e-mail messages in the ENRON dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation).

	MLE (GM)	BE (GM)	ML (DM)	BE (DM)
# Predicted = given	186	16	1607	1607
# False Positives	1435	1607	0	0
# False Negatives	3	0	17	17
# True Positives	14	16	0	0
# True Negatives	172	0	1607	1607
Accuracy	11.5%	1.0%	99.0%	99.0%
Error rate	88.5%	99.0%	1.0%	1.0%
’relevant’ recall	82.4%	100%	0.0%	0.0%
’relevant’ precision	1.0%	1.0%	n.a.	n.a.
’not relevant’ recall	10.7%	0.0%	100%	100%
’not relevant’ precision	98.3%	n.a.	99.0%	99.0%
’relevant’ F-score	2.0%	2.0%	n.a.	n.a.
’not relevant’ F-score	19.3%	n.a.	99.5%	99.5%
’relevant’ percentage	1.0%	1.0%	1.0%	1.0%
Time (test)	13 sec	189 sec	10 sec	27 sec
Time (training)	37 sec	40 sec	38 sec	178 sec

Table F.51: Performance results (set 4), based on the feature length e-mail of e-mail messages in the ENRON dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation).

	MLE (GM)	BE (GM)	ML (DM)	BE (DM)
# Predicted = given	1438	22	1599	1599
# False Positives	164	1599	0	0
# False Negatives	22	0	25	25
# True Positives	3	22	0	0
# True Negatives	1435	0	1599	1599
Accuracy	88.5%	1.4%	98.5%	98.5%
Error rate	11.5%	98.6%	1.5%	1.5%
’relevant’ recall	12.0%	100%	0.0%	0.0%
’relevant’ precision	1.8%	1.4%	n.a.	n.a.
’not relevant’ recall	89.7%	0.0%	100%	100%
’not relevant’ precision	98.5%	n.a.	98.5%	98.5%
’relevant’ F-score	3.1%	2.8%	n.a.	n.a.
’not relevant’ F-score	93.9%	n.a.	99.2%	99.2%
’relevant’ percentage	1.5%	1.5%	1.5%	1.5%
Time (test)	13 sec	184 sec	10 sec	26 sec
Time (training)	38 sec	38 sec	38 sec	162 sec

Table F.52: Performance results (set 5), based on the feature length e-mail of e-mail messages in the ENRON dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation).

TREC dataset

	ML (GM)	BE (GM)	ML (DM)	BE (DM)
# Predicted = given	9920	5149	9968	10098
# False Positives	4465	182	5032	4778
# False Negatives	615	9320	0	124
# True Positives	9353	300	9968	9844
# True Negatives	567	4849	0	254
Accuracy	66.1%	35.1%	66.5%	67.3%
Error rate	33.9%	64.9%	33.5%	32.7%
’relevant’ recall	93.8%	3.1%	100%	98.8%
’relevant’ precision	67.7%	62.2%	66.5%	67.3%
’not relevant’ recall	11.3%	96.4%	0.0%	5.0%
’not relevant’ precision	48.0%	34.2%	n.a.	67.2%
’relevant’ F-score	78.6%	5.9%	79.9%	80.1%
’not relevant’ F-score	18.3%	50.5%	n.a.	9.3%
’relevant’ percentage	65.7%	65.7%	65.7%	65.7%
Time (test)	166 sec	10792 sec	1104 sec	697 sec
Time (training)	679 sec	655 sec	204 sec	6426 sec

Table F.53: Performance results (set 1), based on the feature length e-mail of e-mail messages in the TREC dataset. (GM = Generative model, DM = Discriminative Model, ML = Maximum Likelihood Estimation, BE = Bayesian Estimation).