Exercise 1 Classification

e12045110 Maria de Ronde
e12045110 Quentin Andre
e11921655 Fabian Holzberger

April 23, 2021

Introduction

Applied Algorithms

Performance Metrics

Amazon Reviews Dataset

Dataset Description

Pre-Processing

Parameter-Tuning

Performance-Analysis

Congressional Voting Dataset

Dataset Description

Pre-Processing

Parameter-Tuning

Performance-Analysis

Email Spam Dataset(link to dataset)

Dataset Description

The task of the email spam dataset is to predict if an email is spam or not. The link above contains three datasets from which we choose two, namely the `lingSpam.csv` and `enormSpamSubset.csv` for our project, since they have no missing values and the same layout. The dataset contains 12604 emails where 43.11
Every email has a binary target-label assigned, such that a 0 marks non-spam and a 1 marks spam emails. In figure 2 the distribution of the characters per email is shown. We see that most emails have a lenght in the range of 100 to 10.000 characters.

Index	Body	Label
100	Subject: inexpensive online medication here pummel wah springtail cutler bodyguard we ship quality medications overnight to your door !...	1
6006	Subject: organizational changes we are pleased to announce the following organizational changes : enron global assets and services in order to increase senior management focus on our international businesses...	0

Figure 1: Structure of the Email-Spam Dataset

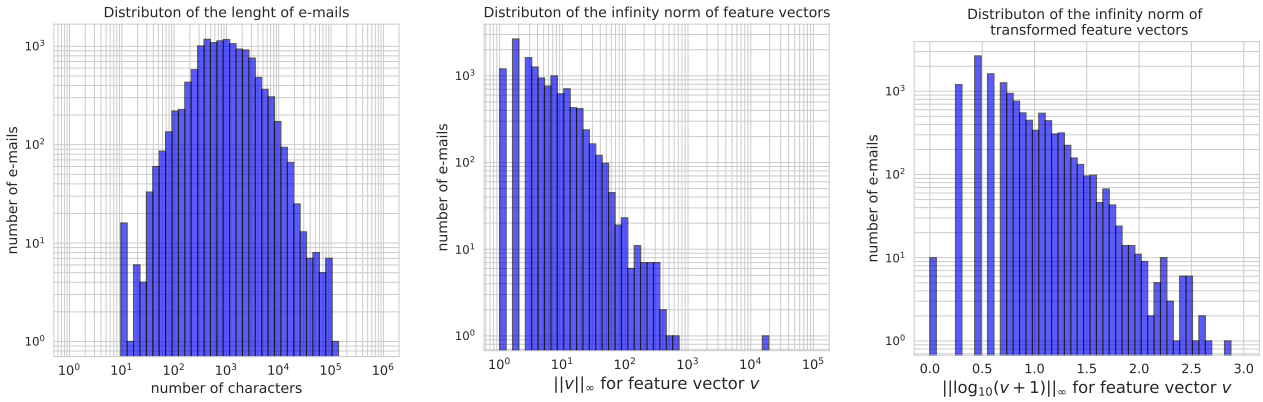

Figure 2: left: Distribution of e-mail lenghts, middle: Distribution of maximum norm in extracted word vectors with lenght 8000, right: Distribution of maximum norm in extracted word vectors with lenght 8000 after removing outlyers and applying logarithmic transformation

Pre-Processing

In the dataset are 213 duplicate emails that we first remove and then peform the train/test-split where the testset size is 20% of the original dataset. Next we apply the Bag of Words feature extractor to each email. The algorithm converts every email to a vector $v \in \mathbb{Z}^N$ of intergers. First we create a list of all words and count their occurences in all emails. Then we take the N most common words and count the occurences of the most common words in every email to get v. Before applying the Bag of word extractor we pre-process emails by the following steps:1. remove links (http...), 2. remove all characters exept alphabetical chars and numbers, 3. convert uppercase to lowercase, 4. split text-bodys into separate words, 5. lemmatize all words, 6. remove stopwords. For the steps 5., 6. we use nltk python package. By the preprocessing we reduce the number of distinct words from 126019 to 103759 words.
In figure 2 middle we see the distribution of the maximum norm of the extracted vectors. One can identify that the maximum norm spans several orders of magnitude from 0 to 10^5. Especially there is only one vector v with $||v||_\infty > 10^4$. Outlyers with $||v||_\infty > 10^3$ are therefor removed in the testset. Additionally we apply the logarithmic transformation $\log_{10}(x+1)$ to all the elements of a vector and obtain a $||\cdot||_\infty$ distrubution that is bounded by the maximum magnithude. Note that we add 1 to all components of a vector since this component is 0 after the logarithmic transformation. The distribution after the transformation is shown in figure 2 right.

Parameter-Tuning

Performance-Analysis

Bridges Dataset(link to dataset)

Dataset Description

Pre-Processing

Parameter-Tuning

Performance-Analysis

Conclusion