Signed Rank-Turbulence Divergence and Spam Classification

Adeline Southard

Abstract

This project explores the utility of rank turbulence divergence as a metric for performing binary classification of e-mails as spam or ham (ham is email that is not spam).

To accomplish this classification a minor modification of the rank turbulence divergence described in [2] is utilized that we refer to as signed rank turbulence divergence, where the sign is the classification indicator, and the magnitude is a measure of the classification strength.

A Python script was developed to perform this classification. The script enabled the efficient evaluation of a variety of parameters on classification performance. In addition to the tuning parameter for measuring the rank turbulence divergence [5], a variety of parameters related to the cleaning of the spam assassin email corpus [2] were investigated. The inclusion or removal of stop words, hyperlinks, email addresses, punctuation, case, and/or numbers were investigated. In addition, lemmatization and stemming was also investigated.

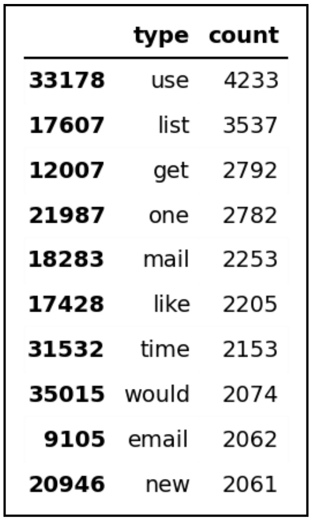
Considering the simplicity of this classification approach the performance of email classification was remarkably good and consistent more involved (and often less interpretable) neural network-based methods that require training. A benefit of the RTD approach to classification relative to other methods that require the tuning of many parameters (such as the weights in a neural network) is that it is immune to overfitting and hence the performance is expected to be generalizable (although this has not been verified beyond the spam assassin corpus).

I. Text Cleaning

Text cleaning was performed using the Python `email` and `nltk` package for the natural language processing.

The `email` package allowed for parsing of the body of the email, and the `nltk` package allowed for removal of stopwords and punctuation, performing lemmatization and stemming, and changing case.

The output from the corpus cleaning is a bag-of-words, ordered by count:



	type	count
33178	use	4233
17607	list	3537
12007	get	2792
21987	one	2782
18283	mail	2253
17428	like	2205
31532	time	2153
35015	would	2074
9105	email	2062
20946	new	2061

Figure 1: Preview of the bag-of-words produced by email cleaning. Words are labeled as `type` and number of word occurrences are labeled `count`. Words are sorted by `count`.

III. Rank-Turbulence Divergence

Rank-turbulence divergence (RTD) provides a measure of how different two datasets are. RTD is a rank-based, semi-positive definite metric that is linearly separable [2]. The linear separability property of RTD is leveraged to determine the RTD contribution for an individual email, which is in turn utilized for classification.

It is defined as:

$$D_{\alpha}^{R}(R_1 \parallel R_2) = \frac{1}{\mathscr{N}_{1,2;\alpha}} \frac{\alpha+1}{\alpha} \sum_{\tau \in R_{1,2;\alpha}} \left| \frac{1}{[r_{\tau,1}]^{\alpha}} - \frac{1}{[r_{\tau,2}]^{\alpha}} \right|^{1/(\alpha+1)} \tag{1}$$

Where r is the rank of a type (in this case, a given word) τ, α is a tunability parameter, and the normalization factor $\mathscr{N}_{1,2;\alpha}$, such that the RTD of disjoint datasets is 1. Hence, the normalized RTD is bounded by 0 for identical datasets and 1 for disjoint datasets. Note that $[N_1 + \frac{1}{2}N_2]$ and $[N_2 + \frac{1}{2}N_1]$ in the equation for the normalization constant are a consequence of using the average rank for types that occur with the same frequency.

$$\mathscr{N}_{1,2;\alpha} = \frac{\alpha+1}{\alpha} \sum_{\tau \in R_1} \left| \frac{1}{[r_{\tau,1}]^{\alpha}} - \frac{1}{[N_1 + \frac{1}{2}N_2]^{\alpha}} \right|^{1/(\alpha+1)} + \frac{\alpha+1}{\alpha} \sum_{\tau \in R_2} \left| \frac{1}{[N_2 + \frac{1}{2}N_1]^{\alpha}} - \frac{1}{[r_{\tau,2}]^{\alpha}} \right|^{1/(\alpha+1)} \tag{2}$$

It follows that the RTD for a single type (word) in a bag-of-words model is:

$$D_{\alpha}^{R} = \frac{1}{\mathscr{N}_{1,2;\alpha}} \frac{\alpha+1}{\alpha} \left| \frac{1}{[r_{\tau,1}]^{\alpha}} - \frac{1}{[r_{\tau,2}]^{\alpha}} \right|^{1/(\alpha+1)} \tag{3}$$

Using this model to mark difference in spam and ham datasets begs the question of three cases: What is the RTD for:

$$D^R_\alpha(\text{spam, ham}) \tag{4}$$
$$D^R_\alpha(\text{spam, spam}) \tag{5}$$
$$D^R_\alpha(\text{ham, ham}) \tag{6}$$

Given that RTD is a measure of difference, could it be a decent method of classification of spam and ham? This may be possible, but, for RTD to be sufficient, we would want (4) to be *large* (that spam and ham word RTDs are *more* different) and (5) and (6) to be *small* (that spam and spam and ham and ham word RTDs are *less* different).

First, we plot the RTD for (4). This can be achieved by plotting spam words versus ham words. An investigation needs to be done on what α maximizes the RTD difference between spam and ham, spam and spam and ham and ham, but for now $\alpha = 0$ will be used.

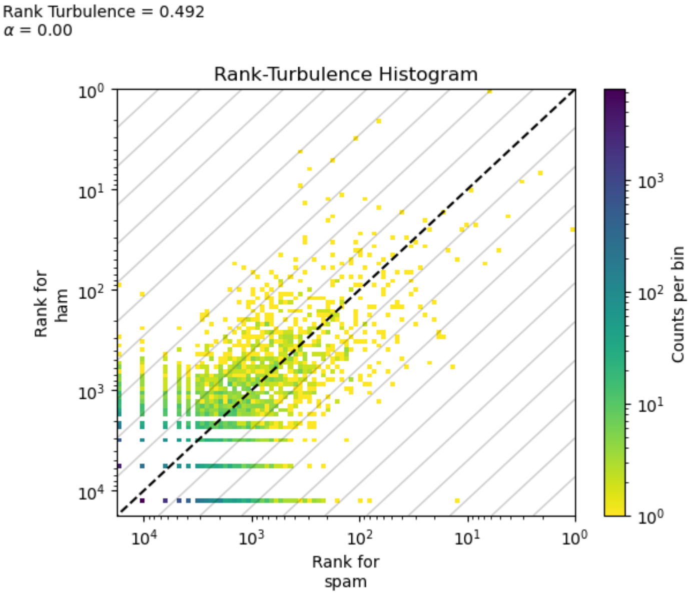
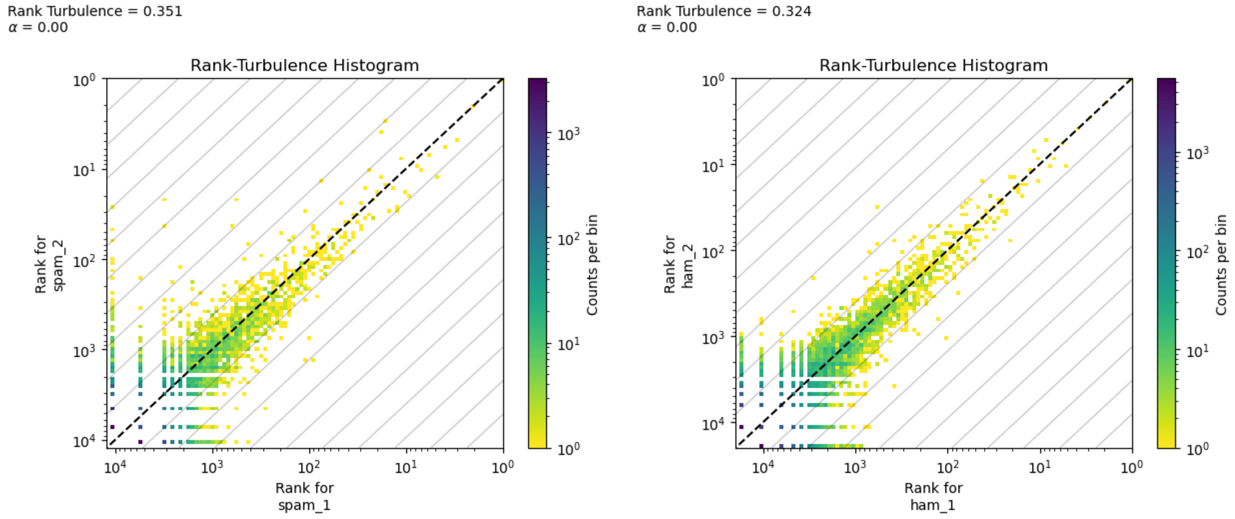


Figure 2: RTD for spam emails plotted against ham emails. α value used was 0 and the resulting RTD value was 0.492, as shown on the plot.

For the latter two cases, shuffling and then splitting the datasets supplies random halves for spam and ham, which can be plotted.



(a) Spam versus Spam

(b) Ham versus Ham

Figure 3: Same-type RTD plots. α used for both plots was 0 and the resulting RTD for the spam and ham sets were 0.351 and 0.324, respectively.

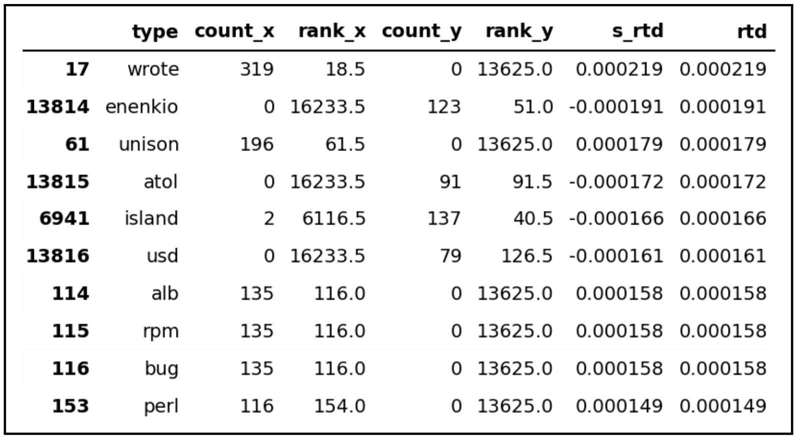
Notice that the RTD results and plots show a different story for the two datasets; spam and ham have a greater RTD than spam and spam and ham and ham. This is a good sign, as it means that RTD could be a method for classifying spam from ham, given the RTD of each word.

IV. Signed Rank-Turbulence Divergence

How would we assign a “spaminess” value to a word using RTD? Ideally, we would want to assign based on the sign of the word, but recall from (3) that the RTD for a word is always of positive sign. This justifies the existence of a **signed** rank-turbulence divergence. After calculating RTD, reapplying the sign:

$$D_{\alpha}^{SR} = \frac{r_{\tau,2} - r_{\tau,1}}{|r_{\tau,2} - r_{\tau,1}|} \frac{1}{\mathcal{N}_{1,2;\alpha}} \frac{\alpha+1}{\alpha} \left| \frac{1}{[r_{\tau,1}]^{\alpha}} - \frac{1}{[r_{\tau,2}]^{\alpha}} \right|^{1/(\alpha+1)} \tag{7}$$

This application creates a signed (by reapplying the sign), weighted (by applying the tuning parameter α), and normalized (by dividing by the normalization factor) value for each word in our bag-of-words: The “signed rank-turbulence divergence” (SRTD).



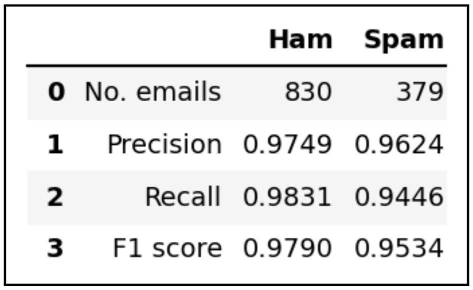
	type	count_x	rank_x	count_y	rank_y	s_rtd	rtd
17	wrote	319	18.5	0	13625.0	0.000219	0.000219
13814	enenkio	0	16233.5	123	51.0	-0.000191	0.000191
61	unison	196	61.5	0	13625.0	0.000179	0.000179
13815	atol	0	16233.5	91	91.5	-0.000172	0.000172
6941	island	2	6116.5	137	40.5	-0.000166	0.000166
13816	usd	0	16233.5	79	126.5	-0.000161	0.000161
114	alb	135	116.0	0	13625.0	0.000158	0.000158
115	rpm	135	116.0	0	13625.0	0.000158	0.000158
116	bug	135	116.0	0	13625.0	0.000158	0.000158
153	perl	116	154.0	0	13625.0	0.000149	0.000149

Figure 4: Preview of the final merged table created using signed rank-turbulence divergence: The ranks for each word in the ham and spam datasets are displayed, and the normal and signed RTD are calculated.

This is now an applicable model for finding the spaminess of a word based off of a training dataset of spam and ham emails, then classifying by summing the SRTDs for each word in an email and judging its sign.

V. Results

After training the model by obtaining the SRTDs for each word in a traning portion of the emails from the Spamassassin corpus, the remaining emails for testing can be classified by their SRTD and checked against their true spam status.



		Ham	Spam
0	No. emails	830	379
1	Precision	0.9749	0.9624
2	Recall	0.9831	0.9446
3	F1 score	0.9790	0.9534

Figure 5: Summary statistics for the results from the testing emails after classification with the SRTDs

To measure the effectiveness of SRTD as a spam classification method, the training emails' status was measured using the training email set (80/20 split, training and testing) and The precision, recall, and F1 score between the real status and the classified status were measured.

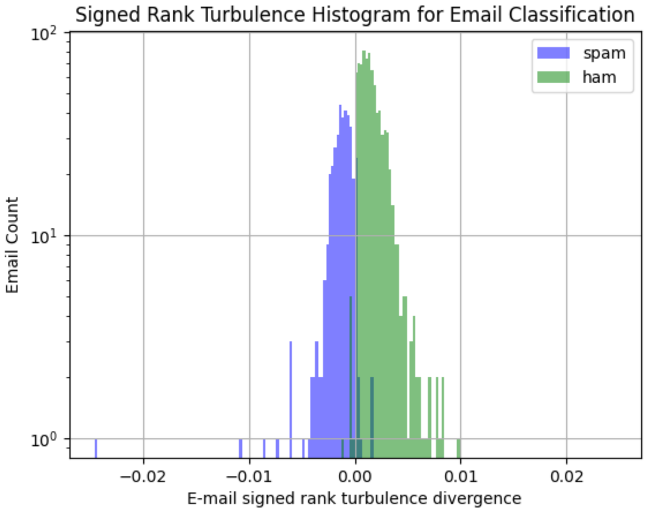


Figure 6: Histogram showing email SRTDs and their identity (spam or ham)

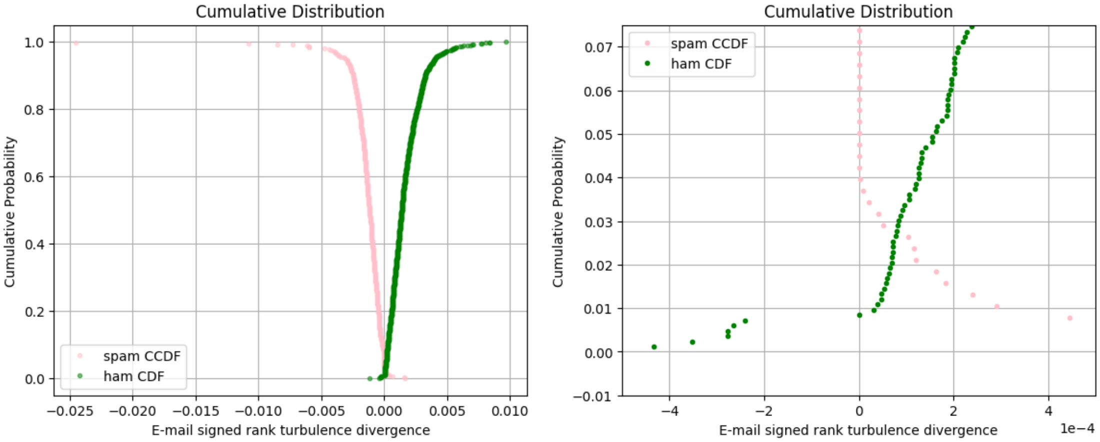


Figure 7: Cumulative distribution showing email SRTDs and their identity (spam or ham)

To show classification effectiveness visually, a histogram and cumulative distribution were plotted. Both depict the overlap where ham and spam emails were correctly (and incorrectly) identified.

References

[1] Ahmed et al. “Machine Learning Techniques for Spam Detection in Email and IoT Platforms: Analysis and Research Challenges”. In: *Security and Communication Networks* 2022 (2022), p. 19. DOI: `https://doi.org/10.1155/2022/1862888`.

[2] Dodds et al. “Allotaxonometry and rank-turbulence divergence: a universal insturment for comparing complex systems”. In: *EPJ Data Science* 12.37 (2023). DOI: `https://doi.org/10.1140/epjds/s13688-023-00400-x`.

[3] Kaddoura et al. “A systematic literature review on spam content detection and classification”. In: *PeerJ Comput Sci.* 8.e830 (2022). DOI: `https://doi.org/10.7717/peerj-cs.830`.

[4] Emmanuel Gbenga Dada et al. “Machine learning for email spam filtering: review, approaches and open research problems”. In: *Heliyon* 5.6 (2019), e01802. ISSN: 2405-8440. DOI: `https://doi.org/10.1016/j.heliyon.2019.e01802`. URL: `https://www.sciencedirect.com/science/article/pii/S2405844018353404`.

[5] The Apache Software Foundation. *Spamassassin*. 2023. URL: `https://github.com/apache/spamassassin` (visited on 12/14/2023).