Machine learning basics with applications to email spam detection

Lydia Song, Lauren Steimle, Xiaoxiao Xu, and Dr. Arye Nehorai

Department of Electrical and Systems Engineering, Washington University in St. Louis

Washington University in St. Louis School of Engineering & Applied Science

Abstract

Machine learning is a branch of artificial intelligence concerned with the creation and study of systems that can learn from data. A machine learning system could be trained to distinguish between spam and non-spam (ham) emails. We aim to study current methods in machine learning to identify the best techniques to use in spam filtering. We found that the One-Nearest Neighbor algorithm achieved the best performance.

Motivation

Email has become one of the most important forms of communication. In 2013, there were about 180 billion emails sent per day worldwide and 65% of the emails sent were spam emails. Links in spam emails may lead to users to websites with malware or phishing schemes. Therefore, an effective spam filtering technology is a significant contribution to the sustainability of the cyberspace and to our society.

Many spam filters rely on Domain Name System-Based Blackhole Lists to keep track of IP addresses that send large amounts of spam so that future email from these addresses can be rejected. However, spammers are circumventing these lists by using larger numbers of IP addresses. Current blacklisting techniques could be paired with content-based spam filtering methods to increase effectiveness.

Methods

Machine learning systems operate in two stages: training and classification.

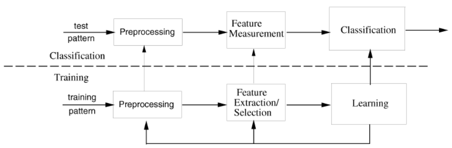


Image credit: "Statistical pattern recognition: A review"

The goal of preprocessing is to remove any noise and normalize the data, and create a compact representation of the data. In training mode, the classifier will determine input patterns from a set of training data and determine how to partition the feature space. In testing mode, the classifier assigns testing data to a class based on their features. Performance results are determined from these classifications.

A dataset of 1000 emails from the Text Retrieval Conference (TREC) 2007 corpus was used to train and test the classifiers.

Preprocessing

Spam Email in Web Browser

Your history shows that your last order is ready for refilling.

Thank you,

Sam Mcfarland
Customer Services

Spam Email in Data Set

```
Content-Class: urn:content-classes:appointment
To: "Gnitpick" <gnitpick@speedy.uwaterloo.ca>
Subject: Please Read This C
Date: Tue, 29 Apr 2003 02:40:43 -0700
MIME-Version: 1.0
Content-Type: multipart/alternative;
     boundary="--9309848958849437758"
Status: 0
Content-Length: 427
Lines: 20

----9309848958849437758
Content-Type: text/plain;
     charset="iso-0230-6"
Content-Transfer-Encoding: quoted-printable

Your history shows that your last order is ready for
refilling.

Thank you,

Sam  Mcfarland
Customer Services


----9309848958849437758--
```

1. Tokenize

tokens= [$\boxed{\text{'your'}}$, 'history', $\boxed{\text{'shows'}}$, $\boxed{\text{'that'}}$, $\boxed{\text{'your'}}$, 'last', 'order', $\boxed{\text{'is'}}$, $\boxed{\text{'ready'}}$, $\boxed{\text{'for'}}$, 'refilling', 'thank', $\boxed{\text{'you'}}$, 'sam', 'mcfarland', 'customer services']

2. Filter words

filtered_words=['history', 'last', 'order', 'ready', 'refilling', 'thank', 'sam', 'mcfarland', 'customer', 'services']

3. Bag of words

bag of words=['histori', 'last', 'order', 'readi', 'refil', 'thank', 'sam', 'mcfarland', 'custom', 'servi']

4. Create a feature matrix

$$
\begin{array}{c|cccc}
 & \text{'histori'} & \text{'last'} & & \text{'servi'} \\
\hline
\text{Email 1} & 1 & 1 & \dots & 1 \\
\text{Email 2} & 0 & 3 & \dots & 5 \\
 & \vdots & \vdots & \ddots & \vdots \\
\text{Email m} & 2 & 1 & \dots & 6
\end{array}
$$

Dimensionality Reduction - Hashing Trick

Without Hashing, the dimensionality of the feature matrix with 70 emails is 9403. After Hashing, the dimensionality is reduced to the number of hash buckets (572, 1024, or 2048).

bag of words=['histori', 'last', 'order', 'readi', 'refil', 'thank', 'sam', 'mcfarland', 'custom', 'servi']

Calculate the hash index unrestrained to the bucket size

hash= [6959089342692012399, 5971037325615305956, 8134888095742582275, ……… 1112741823675571392]

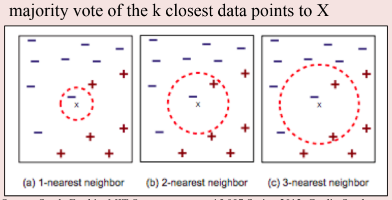
Mod bucket size to get final hash index within the bucket size

hashed index=[367, 228, 515, 632, 549, 122, 629, 473, 192]

Classifiers

k-Nearest Neighbors

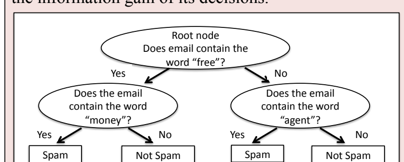
Predict the label of a data point X by using a majority vote of the k closest data points to X



Source: Seyda Ertekin, MIT Opencourseware, 15.097 Spring 2012. Credit: Seyda Ertekin.

Decision Tree

Map observations about a data point's features to determine its class. The tree is constructed to maximize the information gain of its decisions.



Naïve Bayesian

Compare $p(Spam|F_1,\dots,F_n)$ and $p(Ham|F_1,\dots,F_n)$

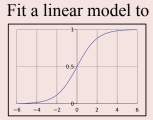
$$p(C|F_1,\dots,F_n)=\frac{p(C)p(F_1,\dots,F_n|C)}{p(F_1,\dots,F_n)} \qquad p(C|F_1,\dots,F_n)\propto p(C)p(F_1,\dots,F_n|C)$$

Assume that the features are independent of each other

$$\underset{c}{\arg\max}\; p(C=c)\prod_{i=1}^{n}p(F_i=f_i|C=c)$$

Logistic Regression

Fit a linear model to the feature space $z=\theta_0+\sum_{i=1}^{n}\theta_i X_i$



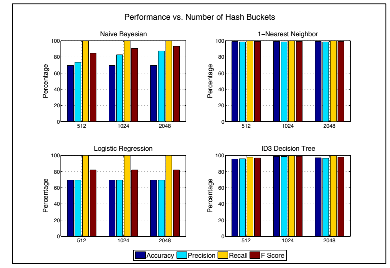
$$p(Spam|z)=\frac{1}{1+e^{-z}}$$

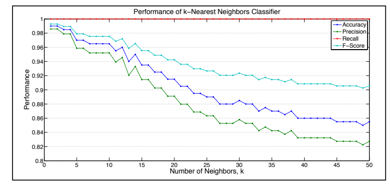
$$p(Ham|z)=1-p(Spam|z)=\frac{e^{-z}}{1+e^{-z}}$$

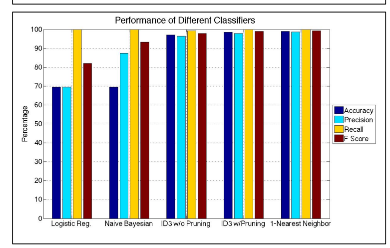
Results

Key performance measures

- Accuracy – percentage of correctly identified emails
- Precision – percentage of emails classified as spam that were actually spam
- Recall – percentage of spam emails that were accurately classified
- F-score – 2*Precision*Recall / (Precision + Recall)







Conclusions

The One-Nearest Neighbor algorithm had the best performance with 99.00% accuracy, 98.58% precision, and 100% recall.

All of the algorithms had very high recall performance and lower precision. This suggests it is easy to classify spam emails correctly and more difficult to classify ham emails correctly.

Literature Cited

A. Jain, R. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Trans*. PAMI, 22(1):4–37, 2000.

A. Ramachandran, D. Dagon, and N. Feamster, "Can DNS-based blacklists keep up with bots?," in CEAS 2006 The Second Conference on Email and Anti-Spam, 2006.

G. Cormack, "Email spam filtering: A systematic review," Foundations and Trends in Information Retrieval, vol. 1, no. 4, pp. 335–455, 2008.

Christina V, Karpagavalli S and Suganya G (2010), 'A Study on Email Spam Filtering Techniques', International Journal of Computer Applications (0975 – 8887), Vol. 12- No. 1, pp. 7-9

Acknowledgements

Thank you to our research mentors Xiaoxiao Xu and Arye Nehorai for their guidance and support throughout this project. We would also like to thank Ben Fu, Yu Wu, and Killian Weinberger for their help in implementing the preprocessing, decision tree, and hash tables, respectively.

Further Information

We encourage those that are interested in learning more about statistical pattern recognition and machine learning to look to Andrew Ng's Machine Learning course at www.coursera.org.