3-15-2019

Stochastic Methods to Find Maximum Likelihood for Spam E-mail Classification

Seyed M. -H. Mansourbeigi
Utah State University, phy_math_ee@yahoo.com

Stochastic Methods to Find Maximum Likelihood for Spam E-mail Classification

Seyed M-H Mansourbeigi[1],
[1] Department of Computer Science, College of Engineering,
Utah State University, 4205 old Main Hill, Logan Utah 84322
smansourbeigi@aggiemail.usu.edu

Abstract. The increasing volume of unsolicited bulk e-mails leads to the need for reliable stochastic spam detection methods for the classification of the received sequence of e-mails. When a sequence of emails is received by a recipient during a time period, the spam filters have already classified them as spam or not spam. Due to the dynamic nature of the spam, there might be emails marked as not spam but are actually real spams and vice versa. For the sake of security, it is important to be able to detect real spam emails. This paper utilizes stochastic methods to refine the preliminary spam detection and to find maximum likelihood for spam e-mail classification. The method is based on the Bayesian theorem, hidden Markov model (HMM), and the Viterbi algorithm.

1 Introduction

Spams are unwanted emails that the user does not want to have in his inbox. Email Spam is one of the major problems of the today's internet, bringing financial damage to companies and annoying individual users. Spam is not only offensive and annoying; it causes loss of productivity, decreases bandwidth and costs companies a lot of money. Blocking spam email is considered a priority for network administrators and security researchers. Spam filters are employed to assist the user in deciding if an email is worth reading or not. There have been tremendous research efforts in this field that resulted in a lot of commercial spam filtering products, such as: methods for construction of filters to eliminate unwanted messages [1], comparison between the performances of machine learning-based classifiers in filtering email spam [2], [3].

The challenging problem with spam filtering is the dynamic nature of the spam. The characteristics (e.g. topics, frequent terms) of spam e-mail vary rapidly over time as

spammers always seek to invent new strategies to bypass spam filters [3]. One cannot develop a filter and immediately implement it, because it will not have any basis for classifying a document as spam or not spam. When a sequence of emails is received by a recipient during a time period, the spam filters have already classified them as spam or not spam. Due to the dynamic nature of the spam, there might be emails marked as not spam but are actually real spams. It is important to be able to detect real spam emails not only for personal convenience, but also for security. This paper proposes a stochastic method to refine the preliminary spam detection. The method is based on the Bayesian theorem, hidden Markov model (HMM), and the Viterbi algorithm.

In the spam detection ground work, the Bayesian filtering works by evaluating the probability of different words appearing in legitimate and spam mails and then classifying them based on those probabilities [4], [5]. A hidden Markov model (HMM) is a simple dynamic Bayesian network that is characterized by the initial state, state transition, and emission probabilities. This statistical model is derived by assuming that the system under consideration is a Markov process with hidden states. The HMM is able to correct deliberate misspellings, incorrect segmentations (adding /removing spaces), and other word modifications [6], [7], [8].

The approach in this paper answers the following questions:

a) How to find the probabilities (transition, initial, and emission) for the Hidden Markov model?

b) What is the initial probability that a random email is spam or not spam?

c) If the indication (from the emails' header) for a finite sequence of emails on a finite time period, is marked as spam or not spam, what is the legitimate sequence of emails, based on the stochastic model.

2 Hidden Markov Model (HMM)

Given a sequence of states in different time frames (marked as spam and marked as not spam), what are the most likely states that are congruent with the derived probabilities and stochastic model based on training data. The training data in this case is the confirmed and real emails identified as spam (S) or not spam (NS).

Suppose the training data is given by the sequence of emails:

S, S, S, S, NS, NS, S, S, S, S, NS, NS, S, S, S,..,..NS…,….S…,…NS...,…S… (**1**)

The transition probabilities are observed based on the number of times that the switch from spam to not spam, not spam to spam happens and also the number of times that the switch does not happen. These numbers are

$$S \rightarrow S \quad 85 \text{ times} \qquad NS \rightarrow S \quad 35 \text{ times} \tag{2}$$

$$S \rightarrow NS \quad 15 \text{ times} \qquad NS \rightarrow NS \quad 65 \text{ times} \tag{3}$$

Therefore the transition probabilities are shown in Fig. 1.

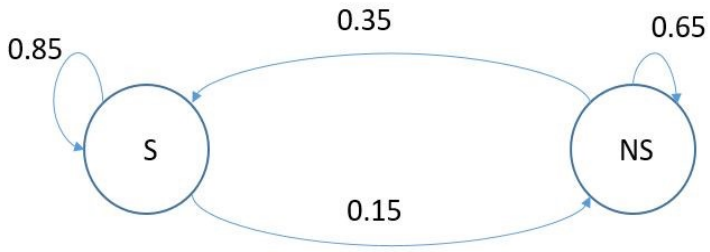

Fig. 1. Transition probabilities

Based on the transition probabilities, the initial probabilities are calculated:

$$P(S) = 0.85\ P(S) + 0.35\ P(NS) \qquad P(NS) = 0.15\ P(S) + 0.65\ P(NS) \tag{4}$$

Since P(S) + P(NS) =1, P(S) = 0.7 and P(NS) =0.3. Meaning that without any knowledge or assumption from the header of emails, P(NS) (the probability that an email is not spam) = 0.3 (a priori).

The hidden states are marked as spam (MS) and marked as not spam (MNS). In order to find emission probabilities, the Bayes' theorem applies to conditional probabilities calculation. The Bayes theorem states the prior probability (a priori): unconditional probabilities of our hypothesis before we get any data or any new evidence. Simply speaking, it is the state of our knowledge before the data is observed. The posteriori probability is a conditional probability about our hypothesis (our state of knowledge) after we revised based on the new data [9].

For the hidden state MS, a posteriori, P(S | MS) = 0.85. Meaning that the probability of the state to be spam is 0.85 if the hidden spam is marked as spam. For the hidden state MNS,

P(S | MNS) = 0.15. Meaning that if hidden state marked as not spam, then the probability of the state to be spam is 0.15. Similarly, a posteriori, P(NS | MS) = 0.35 and P(NS | MNS) = 0.65.

Based on the training data, the emission probabilities emitted from the hidden states are shown in Fig. 2.

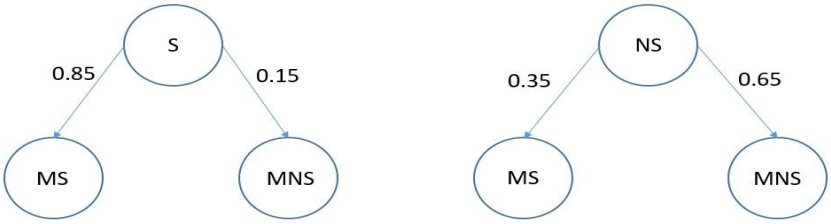

Fig. 2. Emission probabilities

Based on all the probabilities the HMM is shown in Fig. 3.

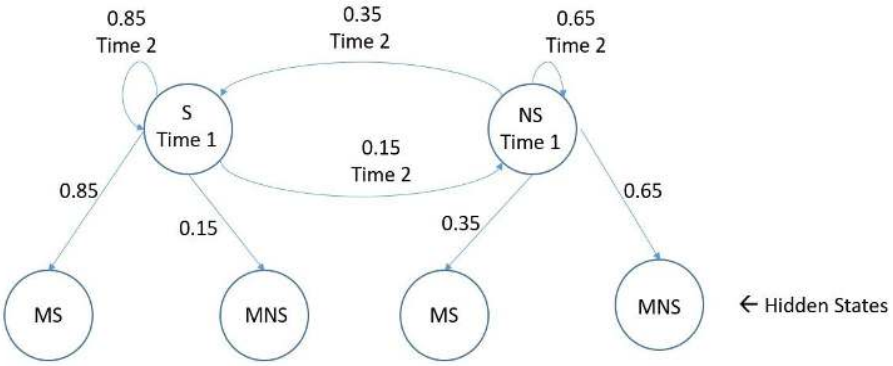

Fig. 3. HMM

3 Maximum likelihood estimation (MLE) and Viterbi algorithm

The Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states, the Viterbi path that results in a sequence of observed events especially in the context of hidden Markov models [10]. For a simple case in which there

are two emails with the hidden states at time period Time 1 and Time 2, given respectively by MS and MNS, what are the actual states based on HMM?

For the two states S and NS there are $2^2 = 4$ transitions:

$$S \rightarrow S \qquad S \rightarrow NS \qquad NS \rightarrow S \qquad NS \rightarrow NS \tag{5}$$

Based on the HMM Fig. 3, the probabilities associated with the four transition states are as in Fig. 4.

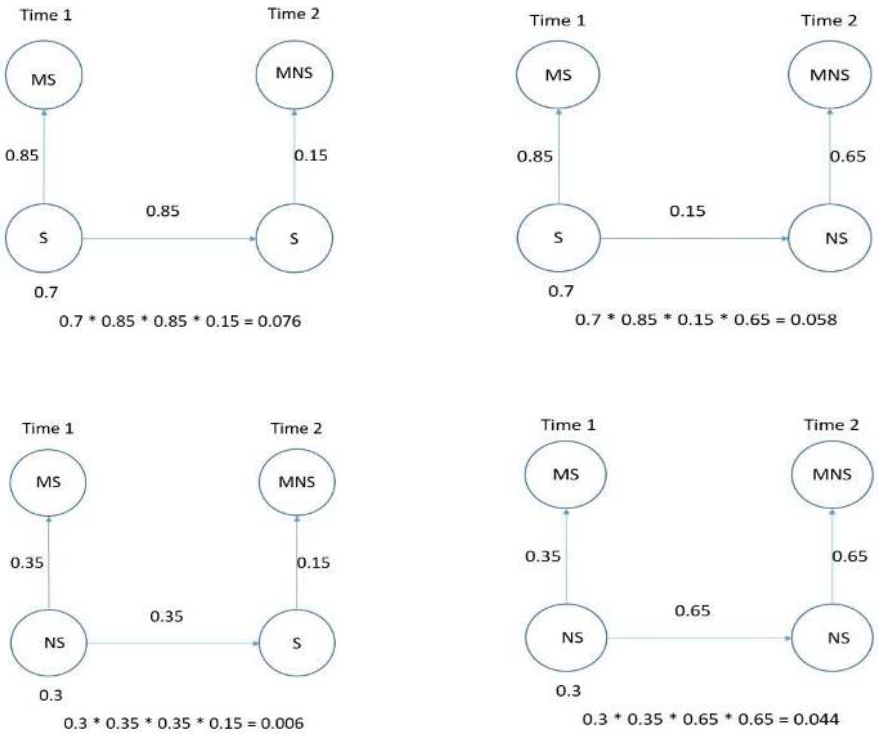

Fig. 4. Probabilities associated with the four transition states

Therefore, MLE for this case is 0.076. Meaning that the actual transition states based on HMM is S → S for MS and MNS.

For the three emails with the hidden states at time period Time 1, Time 2, and Time 3, given respectively by MS, MNS, and MS, based on HMM, the actual states are calculated as follows.

For the two states S and NS there are $2^3 = 8$ transitions:

$$S \rightarrow S \rightarrow S \quad NS \rightarrow S \rightarrow NS \quad S \rightarrow NS \rightarrow S \quad NS \rightarrow NS \rightarrow NS \qquad (\mathbf{6})$$

$$NS \rightarrow S \rightarrow S \quad S \rightarrow S \rightarrow NS \quad NS \rightarrow NS \rightarrow S \quad S \rightarrow NS \rightarrow NS \qquad (\mathbf{7})$$

Based on the HMM, comparing the probabilities associated with the eight transition states, the MLE for this case is MLE = 0.7 * 0.85 * 0.85 * 0.15 * 0.85 * 0.85 = 0.055. Meaning that the actual transition states based on HMM is S → S → S, for MS, MNS, and MS.

With the same token for the case of a sequence with 4 time periods, there are $2^4 = 16$ paths that are required to be checked for MLE. For 5 time periods, there are $2^5 = 32$ paths and for 6 time periods there will be $2^6 = 64$ paths. It is seen that the number of paths increases exponentially.

In order to reduce the number of paths from 32, 64, and ... to only one path, the dynamical programming and specifically Viterbi algorithm is the natural approach to achieve the solution.

Consider the sequence of hidden states MS, MS, MNS, MNS, MNS, MS given for the time periods Time1, 2, 3, 4, 5, 6 respectively. By utilizing the Viterbi algorithm, based on HMM (Fig. 3), the MLE probabilities for the S and NS states are:
For the time period Time 1, between the two probabilities

$$P(S) = 0.7 * 0.85 = 0.595 \qquad P\ (NS) = 0.3 * 0.35 = 0.105 \qquad (\mathbf{8})$$

The MLE is 0.595. Therefore the state S is the first candidate at Time 1.
For the time period Time 2, between the four probabilities

$$P(S) = 0.595 * 0.85 * 0.85 = 0.429 \quad P(S) = 0.105 * 0.35 * 0.85 = 0.031 \qquad (\mathbf{9})$$

$$P\ (NS) = 0.595 * 0.15 * 0.35 = 0.031 \quad P\ (NS) = 0.105 * 0.65 * 0.35 = 0.024 \qquad (\mathbf{10})$$

The MLE is 0.429. Therefore the state S is the next candidate at Time 2.
For the time period Time 3, between the four probabilities

$$P(S) = 0.429 * 0.85 * 0.15 = 0.055 \quad P(S) = 0.031 * 0.35 * 0.15 = 0.002 \qquad (\mathbf{11})$$

$$P\ (NS) = 0.429 * 0.15 * 0.65 = 0.042 \quad P\ (NS) = 0.031 * 0.65 * 0.65 = 0.013 \qquad (\mathbf{12})$$

The MLE is 0.055. Therefore the state S is the next candidate at Time 3.
Up to here the sequence is S → S → S.
For the time period Time 4, between the four probabilities

$$P(S) = 0.055 * 0.85 * 0.15 = 0.007 \quad P(S) = 0.042 * 0.35 * 0.15 = 0.002 \quad (13)$$

$$P\ (NS) = 0.055 * 0.15 * 0.65 = 0.005 \quad P\ (NS) = 0.042 * 0.65 * 0.65 = 0.018 \quad (14)$$

The MLE is 0.018. Therefore the state NS is the candidate at Time 4.
Therefore the sequence is extended by S → S → S → NS.
For the time period Time 5, between the four probabilities

$$P(S) = 0.007 * 0.85 * 0.15 = 0.0009 \quad P(S) = 0.018 * 0.35 * 0.15 = 0.001 \quad (15)$$

$$P\ (NS) = 0.007 * 0.15 * 0.65 = 0.0007 \quad P\ (NS) = 0.018 * 0.65 * 0.65 = 0.008 \quad (16)$$

The MLE is 0.008. Therefore the state NS is the candidate at Time 5.
Therefore the sequence of states is S → S → S → NS → NS.
For the time period Time 6, between the four probabilities

$$P(S) = 0.001 * 0.85 * 0.85 = 0.0007 \quad P(S) = 0.008 * 0.35 * 0.85 = 0.0024 \quad (17)$$

$$P\ (NS) = 0.001 * 0.15 * 0.35 = 0.00005 \quad P\ (NS) = 0.008 * 0.65 * 0.35 = 0.0018 \quad (18)$$

The MLE is 0.0024, the state S is the candidate at Time 6.
At this point the sequence is S → S → S → NS → NS → S.

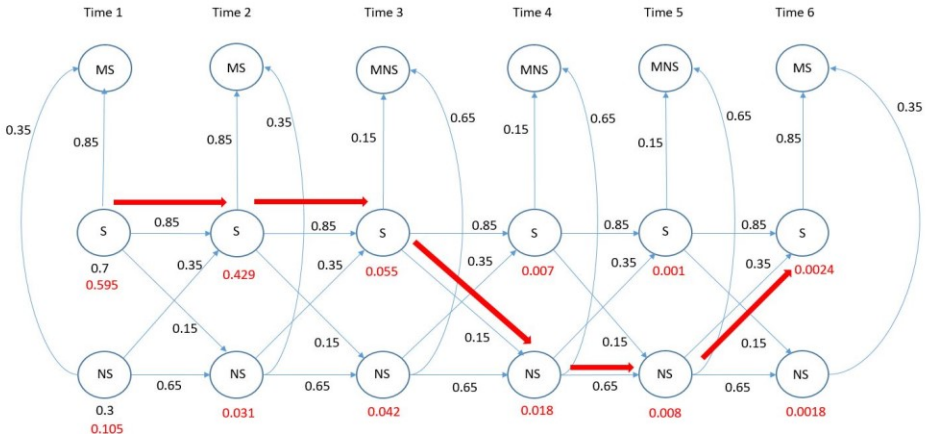

Fig. 5. For the sequence MS, MS, MNS, MNS, MNS, MS the real states are:
S → S → S → NS → NS → S

The concept of conditional probability and Bayesian theory is essential and profound [11]

$$P(S|MS) = P(MS \text{ AND } S)/P(MS) = (P(MS \mid S) * P(S))/P(MS) = (P(MS \mid S) * P(S))/((P(MS|S) * P(S)) * (P(MS|\sim S) * P(\sim S))) \quad (19)$$

4 The python program for Viterbi algorithm

```
# solving with dynamical programming and viterbi algorithm
# Transition Probabilities
# S = spam       NS =   not spam
# MS = marked as a spam  MNS = marked as not spam -- mean
/feelguess it is spam or feel/guess it is not spam
p_ss = 0.85  # p marked as a spam goes to p marked as a spam
- probability for next time around
p_sr = 0.15  # p marked as a spam goes to p marked as a not
spam - probability for next time around
p_rs = 0.35  # p marked as a not spam goes to p marked as a
spam - probability for next time around
p_rr = 0.65  # p marked as a not spam goes to p marked as a
not spam - probability for next time around
# Initial Probabilities
p_s = (p_rs)/(p_rs + p_sr)
p_r = 1 - p_s
#print(p_s)
#print(p_r)
#p_s = 0.7  # p to be marked as a spam - overall probability
#p_r = 0.3  # p to be marked not as a spam - overall
probability
# Emission Probabilities
p_sh = 0.85   # p to be marked as a spam and turns a spam  -
probability from hidden states
p_sg = 0.15   # p to be marked as a spam and turns not a
spam - probability from hidden states
p_rh = 0.35   # p to be marked not as a spam and turns a
spam - probability from hidden states
p_rg = 0.65   # p to be marked not as a spam and turns not a
spam - probability from hidden states
finals = ['S' , 'NS' , 'NS' , 'S' , 'NS', 'S' , 'S' , 'S' ,
'NS' , 'S' , 'NS']
probabilities = []
primary = []
```

```
if finals[0] == 'S':
    probabilities.append((p_s*p_sh, p_r*p_rh))
else:
    probabilities.append((p_s*p_sg, p_r*p_rg))
for i in range(1,len(finals)):
    yesterday_sunny, yesterday_rainy = probabilities[-1]
    if finals[i] == 'A':
        today_sunny = max(yesterday_sunny*p_ss*p_sh,
yesterday_rainy*p_rs*p_sh)
        today_rainy = max(yesterday_sunny*p_sr*p_rh,
yesterday_rainy*p_rr*p_rh)
        probabilities.append((today_sunny, today_rainy))
    else:
        today_sunny = max(yesterday_sunny*p_ss*p_sg,
yesterday_rainy*p_rs*p_sg)
        today_rainy = max(yesterday_sunny*p_sr*p_rg,
yesterday_rainy*p_rr*p_rg)
        probabilities.append((today_sunny, today_rainy))
print(probabilities)
for p in probabilities:
    if p[0] > p[1]:
        primary.append('MS')
    else:
        primary.append('MNS')
print (primary)
print(finals)
```

For the marked sequence ['MS', 'MS', 'MNS', 'MS', 'MNS', 'MS', 'MS', 'MS', 'MS', 'MS', 'MS'] as an input, by executing the above python program, the output is: ['S', 'NS', 'NS', 'S', 'NS', 'S', 'S', 'S', 'NS', 'S', 'NS'].

The method in this paper can be utilized to address the following future works:

1 – Speech recognition
2 – Genetics DNA sequences
3 – Vehicle location
4 – Text tagging WORDS to TAGS (the definite articles, preposition, verb, noun)

References

1. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A Bayesian Approach to Filtering Junk E-mail. In: AAAI'98 Workshop on Learning for Text Categorization (1998)

2. Al-Jarrah, O., Khaterz, I., Al-Duwairi, B.: Identifying Potentially Useful Email Header Features for Email Spam Filtering. ICDS The Sixth International Conference on Digital Society (2012)
3. Awad, W.A., ELseuofi, S.M.: Machine Learning Methods for Spam E-mail Classification. Iternational journal of Computer Science and Information Technology (IJCSIT), Vol. 3, No. 1 (2011) 173-184
4. Eberhardt, J.J.: Bayesian Spam Detection. Scholarly Horizons: University of Minnesota, Morris Journal, Vol. 2, Issue 1 (2015)
5. Freeman, D.M.: Using Naive Bayes to Detect Spammy Names in Social Networks. Proceedings of the 2013 ACM Workshop on Artificial Intelligence and Security, AISec '13, New York, NY, USA (2013) 3-12
6. Lee, S., Jeong, I., Choi, S.: Dynamically Weighted Hidden Markov Model for Spam Deobfuscation. IJCAI Proceedings of the 20th International Joint Conference on Artificial Intelligence (2007) 2523-2529
7. Lee, H., Ng, A.Y.: Spam Deobfuscation Using a Hidden Markov Model. Proc. 2nd Conference on Email and Anti-Spam. Stanford University, CA, USA (2005)
8. Jaswal, V., Sood, N.: Spam Detection System Using Hidden Markov Model. International Journal of Advanced Reserach in Computer Science and Software Engineering, Vol. 3, Issue 7 (2013)
9. Roy, S., Patra, A., Sau, S., Mandal, K., Kunar, S.: An Efficient Spam Filtering Techniques for Email Account. American Journal of Engineering Research (AJER), Vol. 2, Issue-10 (2013) 63-73
10. Viterbi, A. J.: Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. In: IEEE Transactions on Information Theory 13(2) (1967) 260-267
11. Papoulis, A.: Probability, Random Variables, and Stochastic Processes. 3rd edn. McGraw-Hill Series in Electrical Engineering (1991)