Using Support Vector Machines(SVM) learning algorithm and Markov Chains to build a spam classifier

Stuti Chugh

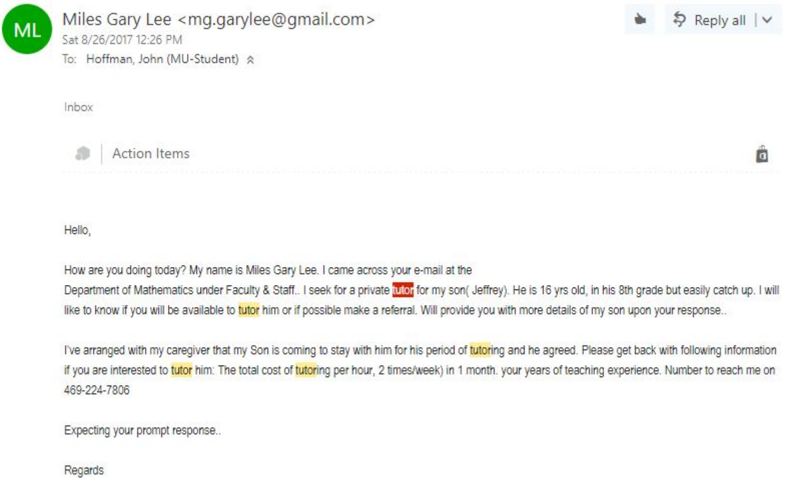
University of Missouri, Department of Mathematics

Background Information and Motivation

Spam Emails

- Time consuming
- Lead to many dangerous phishing activities
- Cause harmful viruses to download on our machine
- lead to a major breach in user's privacy
- Most email systems have spam filtering algorithms
- They learn patterns of similarity in spam emails to predict whether a certain email is spam or not based on the content of an email.
- Spam classifiers often tend to misclassify emails as well.
  - if there is an outlier or exceptional email that didn't match the spam traits
  - Here are the two examples of such a misclassifications in my personal email:

Famous Tutoring Email Phishing Scam: a False Negative



Miles Gary Lee <mg.garylee@gmail.com>
Sat 8/26/2017 12:26 PM
To: Hoffman, John (MU-Student)

Hello,

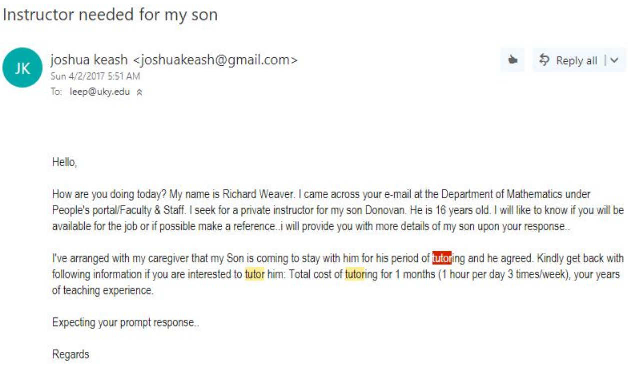
How are you doing today? My name is Miles Gary Lee. I came across your e-mail at the Department of Mathematics under Faculty & Staff.. I seek for a private tutor for my son(Jeffrey). He is 16 yrs old, in his 8th grade but easily catch up. I will like to know if you will be available to tutor him or if possible make a referral. Will provide you with more details of my son upon your response..

I've arranged with my caregiver that my Son is coming to stay with him for his period of tutoring and he agreed. Please get back with following information if you are interested to tutor him: The total cost of tutoring per hour, 2 times/week) in 1 month. your years of teaching experience. Number to reach me on 469-224-7806

Expecting your prompt response..

Regards

Famous Tutoring Email Phishing Scam: a False Negative



Instructor needed for my son

joshua keash <joshuakeash@gmail.com>
Sun 4/2/2017 5:51 AM
To: leep@uky.edu

Hello,

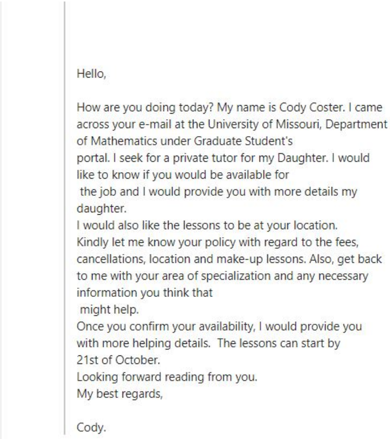
How are you doing today? My name is Richard Weaver. I came across your e-mail at the Department of Mathematics under People's portal/Faculty & Staff. I seek for a private instructor for my son Donovan. He is 16 years old. I will like to know if you will be available for the job or if possible make a reference..i will provide you with more details of my son upon your response..

I've arranged with my caregiver that my Son is coming to stay with him for his period of tutoring and he agreed. Kindly get back with following information if you are interested to tutor him: Total cost of tutoring for 1 months (1 hour per day 3 times/week), your years of teaching experience.

Expecting your prompt response..

Regards

Famous Tutoring Email Phishing Scam: a False Negative

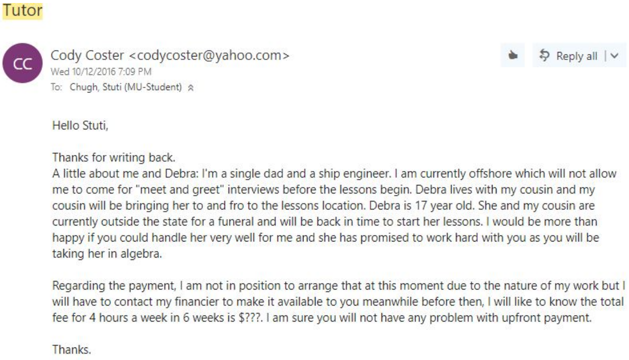


Hello,

How are you doing today? My name is Cody Coster. I came
across your e-mail at the University of Missouri, Department
of Mathematics under Graduate Student's
portal. I seek for a private tutor for my Daughter. I would
like to know if you would be available for
the job and I would provide you with more details my
daughter.
I would also like the lessons to be at your location.
Kindly let me know your policy with regard to the fees,
cancellations, location and make-up lessons. Also, get back
to me with your area of specialization and any necessary
information you think that
might help.
Once you confirm your availability, I would provide you
with more helping details. The lessons can start by
21st of October.
Looking forward reading from you.
My best regards,

Cody.

Famous Tutoring Email Phishing Scam: a False Negative



Tutor

Cody Coster <codycoster@yahoo.com>
Wed 10/12/2016 7:09 PM
To: Chugh, Stuti (MU-Student)

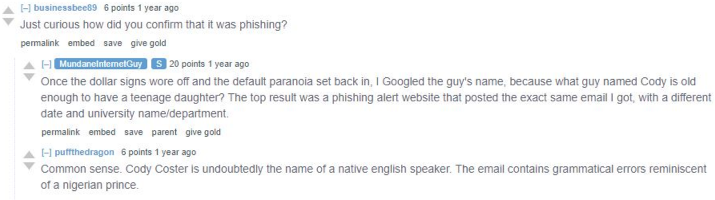
Hello Stuti,

Thanks for writing back.
A little about me and Debra: I'm a single dad and a ship engineer. I am currently offshore which will not allow me to come for "meet and greet" interviews before the lessons begin. Debra lives with my cousin and my cousin will be bringing her to and fro to the lessons location. Debra is 17 year old. She and my cousin are currently outside the state for a funeral and will be back in time to start her lessons. I would be more than happy if you could handle her very well for me and she has promised to work hard with you as you will be taking her in algebra.

Regarding the payment, I am not in position to arrange that at this moment due to the nature of my work but I will have to contact my financier to make it available to you meanwhile before then, I will like to know the total fee for 4 hours a week in 6 weeks is $???. I am sure you will not have any problem with upfront payment.

Thanks.

Famous Tutoring Email Phishing Scam: a False Negative



[–] businessbee89 6 points 1 year ago

Just curious how did you confirm that it was phishing?

permalink embed save give gold

[–] MundaneInternetGuy S 20 points 1 year ago

Once the dollar signs wore off and the default paranoia set back in, I Googled the guy's name, because what guy named Cody is old enough to have a teenage daughter? The top result was a phishing alert website that posted the exact same email I got, with a different date and university name/department.

permalink embed save parent give gold

[–] puffthedragon 6 points 1 year ago

Common sense. Cody Coster is undoubtedly the name of a native english speaker. The email contains grammatical errors reminiscent of a nigerian prince.

- Fraudulent Email Alert Announcement- The University of British Columbia
- Tutoring Scam Explained

Machine Learning Background

Definition

Two definitions of **Machine Learning** are offered. Arthur Samuel described it as: ”the field of study that gives computers the ability to learn without being explicitly programmed.” This is an older, informal definition.

Definition

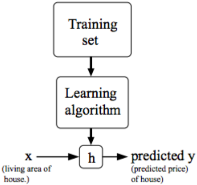
Tom Mitchell provides a more modern definition: ”A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”
Example: playing checkers.
E = the experience of playing many games of checkers
T = the task of playing checkers.
P = the probability that the program will win the next game.

Supervised Learning

- data set and correct answers are given.
- categorized into
    - **Regression**:
        - predict results within a continuous output
        - Given data about the size of houses on the real estate market, try to predict their price. Price as a function of size is a continuous output, so this is a regression problem
    - **Classification** :
        - predict results in a discrete output i.e. map input variables into discrete categories
        - Given a patient with a tumor, we have to predict whether the tumor is malignant or benign. Thus we are doing a binary classification of a tumor

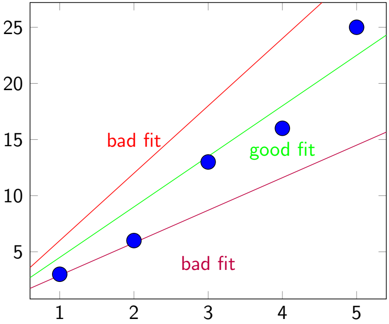
Logistic Regression

A standard supervised learning algorithm can be pictorially represented as follows:



- **input variables or input features**$= x^{(i)}$
- **output variable or target variable**$= y^{(i)}$
- $(x^{(i)}, y^{(i)})$ is one of the say, m training examples in the **training set**
- **hypothesis function** $h(x)$ such that $h\colon X \to Y$ is a good predictor of y given a corresponding values of x

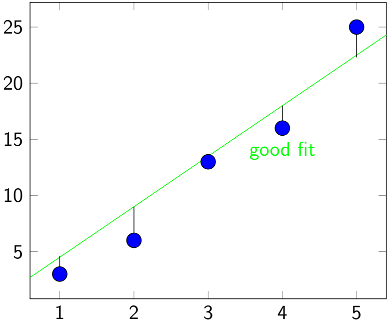
Regression



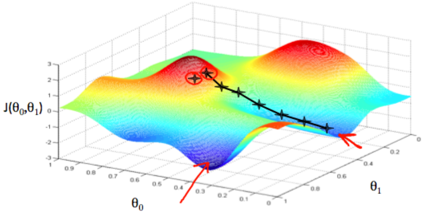
Thus, for a linear model, if we were to represent the hypothesis function as $h_\theta(x) = \theta_0 + \theta_1 x$ finding the best fitting line in this case implies finding the best possible values of **parameters** θ_0 and θ_1

Cost Function and Gradient Descents

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2 = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)^2 \tag{1}$$



Regression



- Need to find derivative of our cost function
- Slope of the tangent= derivative at that point and gives a direction to move towards
- make steps down the cost function in the direction with the steepest descent
- **learning rate** α=size of each step
- simultaneously update θ_i's until it converges to a minimum value

$$\theta_i = \theta_i - \alpha \frac{\partial J(\theta_i)}{\partial \theta_i} \forall \theta_i \tag{2}$$

Multiple Features, Feature Scaling, and mean normalization

$$x = \begin{bmatrix} area \\ bedrooms \\ location \\ age \end{bmatrix} \tag{3}$$

$$h_\theta(x^{(i)}) = \theta_0 + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \dots + \theta_n x_n^{(i)} \tag{4}$$

$x_j^{(i)}$= value of feature j in the i^{th} training example
$x^{(i)}$ = input(features) of thei^{th}training example
m= number of training examples
n = number of features
Thus our multivariable hypothesis looks something like this:

$$h_\theta(x) = \begin{bmatrix} \theta_0 & \theta_1 & \dots & \theta_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix} = \theta^T x \tag{5}$$

Feature Scaling and Normalization

repeat until convergence:

$$\theta_j = \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad \forall j \in (0, n) \tag{6}$$

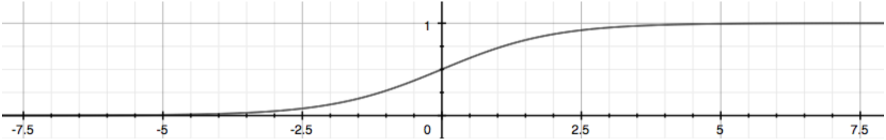
- **Feature scaling** involves dividing input values by the range of the input variable, resulting in a new range of just 1.
- **Mean normalization** involves subtracting the average value for an input variable from the values for that input variable resulting in a new average value for the input variable of just zero.

To implement both of these techniques, adjust your input values as shown in this formula: $x_i = \frac{x_i - \mu_i}{s_i}$ Here μ_i is the average of all values of feature i and s_i is the range of values(max-min) or the standard deviation.

Sigmoid Function

$$0 \leq h_\theta(x) \leq 1 \tag{7}$$

$$g(z) = \frac{1}{1 + e^{(-z)}} \tag{8}$$



$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \tag{9}$$

$$\begin{aligned} h_\theta(x) \geq 0.5 \; &\rightarrow y = 1 \\ h_\theta(x) \leq 0.5 \; &\rightarrow y = 0 \end{aligned} \tag{10}$$

$$\begin{aligned} \theta^T x \geq 0 \rightarrow y = 1 \\ \theta^T x < 0 \rightarrow y = 0 \end{aligned} \tag{11}$$

Classification, Cost Function, and Gradient Descent

$$\begin{aligned} J(\theta) &= \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)}) \\ Cost(h_\theta(x^{(i)}), y^{(i)}) &= -log(h_\theta(x)) \quad if \quad y = 1 \\ Cost(h_\theta(x^{(i)}), y^{(i)}) &= -log(1 - h_\theta(x)) \quad if \quad y = 0 \end{aligned} \tag{12}$$

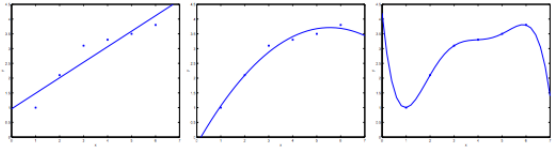
We can compress our cost function as follows:

$$Cost(h_\theta(x^{(i)}), y^{(i)}) = -ylog(h_\theta(x)) - (1 - y)log(1 - h_\theta(x)) \tag{13}$$

Repeat

$$\theta_j = \theta_j - \frac{\alpha}{m}\sum i = 1^m(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \tag{14}$$

Regularization

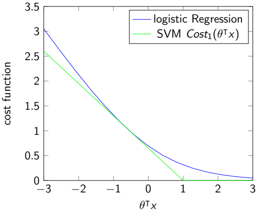


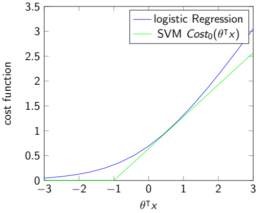
too many features = may lead to **overfitting** or **high variance**
less features= **underfitting** or **high bias**

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}log(h_\theta(x^{(i)}))-(1-y^{(i)})log(1-h_\theta(x^{(i)}))]+\frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2 \quad (15)$$

λ is the **regularization parameter**

Support Vector Machine(SVM) v/s Logistic Regression





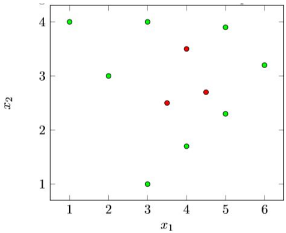
$$\min_{\theta} \sum_{i=1}^{m} [-y^{(i)} \mathbf{cost}_1(\theta^\intercal x^{(i)}) - (1 - y^{(i)}) \mathbf{cost}_0(\theta^\intercal x^{(i)})] + \frac{\lambda}{2} \sum_{j=1}^{n} \theta_j^2 \qquad (16)$$

SVM vs Logistic Regression

$$C\sum_{i=1}^{m}[-y^{(i)}\mathbf{cost}_1(\theta^{\intercal}x^{(i)}) - (1-y^{(i)})\mathbf{cost}_0(\theta^{\intercal}x^{(i)})] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2 \tag{17}$$

$$h_\theta(x) = 1 \quad if \quad \theta^{\intercal}x \geq 0 \quad and \quad 0 \quad otherwise. \tag{18}$$

Kernels



randomly choose three points $l^{(1)}$,$l^{(2)}$, and $l^{(3)}$.
new feature = proximity to **landmarks** $l^{(1)}$,$l^{(2)}$, and $l^{(3)}$.
Gaussian Kernel is a similarity function

- $f_1 = similarity(x, l^{(1)}) = e^{\left(-\frac{(\|x-l^{(1)}\|)^2}{2\sigma^2}\right)}$
- $f_2 = similarity(x, l^{(2)}) = e^{\left(-\frac{(\|x-l^{(2)}\|)^2}{2\sigma^2}\right)}$
- $f_3 = similarity(x, l^{(3)}) = e^{\left(-\frac{(\|x-l^{(3)}\|)^2}{2\sigma^2}\right)}$

Kernels

- Given $(x^{(1)}, y^{(1)}),x^{(2)}, y^{(2)}), ..., x^{(m)}, y^{(m)})$
- Choose $l^{(1)} = x^{(1)},l^{(2)} = x^{(2)},. . . . , l^{(m)} = x^{(m)}$
- Given example x:
  - $f_1 = similarity(x, l^{(1)})$
  - $f_2 = similarity(x, l^{(2)})$
  - ...
  - $f_m = similarity(x, l^{(m)})$

This gives us a feature vector as follows:

$$f = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_m \end{bmatrix} \tag{19}$$

SVM Cost function and training alogrithm

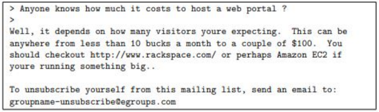
Our SVM does the following to make predictions:

- Hypothesis: Given x, compute features $f \in \mathbb{R}^{m+1}$
  - Predict "y=1" if $\theta^\intercal f \geq 0$
- Training:

$$\min_{\theta} C \sum_{i=1}^{m} [-y^{(i)} \mathbf{cost}_1(\theta^\intercal f^{(i)}) - (1 - y^{(i)}) \mathbf{cost}_0(\theta^\intercal f^{(i)})] + \frac{1}{2} \sum_{j=1}^{m} \theta_j^2$$

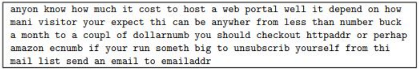
Step 1: Preprocessing Emails

- **Lower-casing**
- **Stripping HTML**
- **Normalizing URLs**: All URLs are replaced with the text httpaddr
- **Normalizing Email Addresses**: All email addresses are replaced with the text emailaddr
- **Normalizing Numbers**: All numbers are replaced with the text number
- **Normalizing Dollars**: All dollar signs ($) are replaced with the text dollar
- **Word Stemming**: Words are reduced to their stemmed form. For example, discount, discounts, discounted and discounting are all replaced with discount.
- **Removal of non-words**: Non-words and punctuation have been removed. All white spaces (tabs, newlines, spaces) have all been trimmed to a single space character



```
> Anyone knows how much it costs to host a web portal ?
>
Well, it depends on how many visitors youre expecting.  This can be
anywhere from less than 10 bucks a month to a couple of $100.  You
should checkout http://www.rackspace.com/ or perhaps Amazon EC2 if
youre running something big..

To unsubscribe yourself from this mailing list, send an email to:
groupname-unsubscribe@egroups.com
```



```
anyon know how much it cost to host a web portal well it depend on how
mani visitor your expect thi can be anywher from less than number buck
a month to a coupl of dollarnumb you should checkout httpaddr or perhap
amazon ecnumb if your run someth big to unsubscrib yourself from thi
mail list send an email to emailaddr
```

Extracting features: Vocabulary List

- create a vocabulary list of most frequently occurring words
- chosen words= all words occurring at least a 100 times in the spam corpus
- map each word in the preprocessed emails into a list of word indices that contains the index of the word in the vocabulary list
- $x_i \in \{0, 1\}$ for an email corresponds to whether the i-th word in the dictionary occurs in the email.

$$x = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^n.$$

Actual Project Result-Matlab

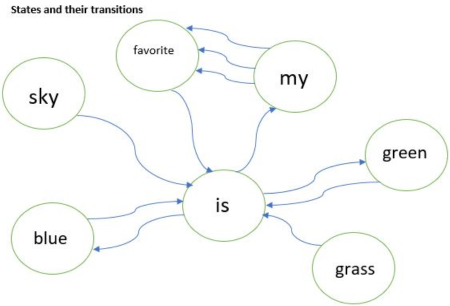
Application of Markov Chain in Spam filtering

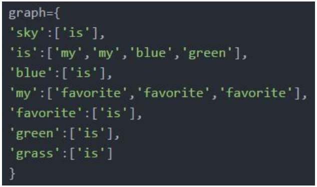
Before talking about how Markov Chain are used for Spam filtering We will talk about why they are so successful in natural language processing in general.

Consider the following 5 simple sentences of the English language:

- **sentence1**=”Sky is blue.”
- **sentence2**=”Blue is my favorite!”
- **sentence3**=”My favorite is green.”
- **sentence4**=”Grass is green.”
- **sentence4**=”Green is my favorite!”

Words as States and Their Respective Adjacency Lists





```
graph={
'sky':['is'],
'is':['my','my','blue','green'],
'blue':['is'],
'my':['favorite','favorite','favorite'],
'favorite':['is'],
'green':['is'],
'grass':['is']
}
```

sky is blue

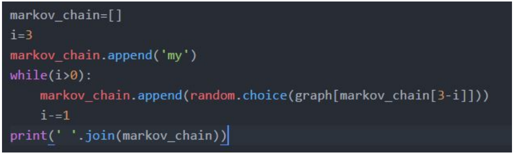
blue is my favorite

my favorite is green

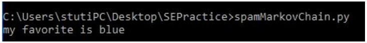
grass is green

green is my favorite

Markov Chain for Natural Language Processing

- **Markov Chain of order 1**: current letter determines next letter.
- $Pr(X_{i+1} = j|X_0 = k_0, X_1 = k_1, ..., X_{t-1} = k_{t-1}, X_t = i) = Pr(X_{t+1} = j|X_t = i)$
- In English(or any other language), there is proper structure, hence words are linked thus we can use Markov Chains.
- Random Walks using Transition Matrix
- 

```
markov_chain=[]
i=3
markov_chain.append('my')
while(i>0):
    markov_chain.append(random.choice(graph[markov_chain[3-i]]))
    i-=1
print(' '.join(markov_chain))
```

- 

```
C:\Users\stutiPC\Desktop\SEPractice>spamMarkovChain.py
my favorite is blue
```

- Generates a semantically accurate new sentence
- as opposed to a completely random selection of 4 words like-"my blue grass is"

Transition Matrix

$$P = \begin{bmatrix} 0 & sky & is & blue & my & favorite & green & grass \\ sky & P_{11} & P_{12} & P_{13} & P_{14} & P_{15} & P_{16} & P_{17} \\ is & P_{21} & P_{22} & P_{23} & P_{24} & P_{25} & P_{26} & P_{27} \\ blue & P_{31} & P_{32} & P_{33} & P_{34} & P_{35} & P_{36} & P_{37} \\ my & P_{41} & P_{42} & P_{43} & P_{44} & P_{45} & P_{46} & P_{47} \\ favorite & P_{51} & P_{52} & P_{53} & P_{54} & P_{55} & P_{56} & P_{57} \\ green & P_{61} & P_{62} & P_{63} & P_{64} & P_{65} & P_{66} & P_{67} \\ grass & P_{71} & P_{72} & P_{73} & P_{74} & P_{75} & P_{76} & P_{77} \end{bmatrix}$$

$$P_{ij} = Pr\{X_{n+1} = j(next\ word)|X_n = i(current\ word)\}.$$

Transition Matrix

$$P = \begin{bmatrix} 0 & sky & is & blue & my & favorite & green & grass \\ sky & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ is & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ blue & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ my & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ favorite & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ green & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ grass & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Transition Matrix

$$P = \left[\begin{array}{cccccccc}
0 & \mathit{sky} & \mathit{is} & \mathit{blue} & \mathit{my} & \mathit{favorite} & \mathit{green} & \mathit{grass} \\
\mathit{sky} & 1 & 1+1 & 1 & 1 & 1 & 1 & 1 \\
\mathit{is} & 1 & 1 & 1+1 & 1+2 & 1 & 1+1 & 1 \\
\mathit{blue} & 1 & 1+1 & 1 & 1 & 1 & 1 & 1 \\
\mathit{my} & 1 & 1 & 1 & 1 & 1+3 & 1 & 1 \\
\mathit{favorite} & 1 & 1+1 & 1 & 1 & 1 & 1 & 1 \\
\mathit{green} & 1 & 1+1 & 1 & 1 & 1 & 1 & 1 \\
\mathit{grass} & 1 & 1+1 & 1 & 1 & 1 & 1 & 1
\end{array}\right]$$

Transition Matrix

$$P = \left[\begin{array}{cccccccc}
0 & sky & is & blue & my & favorite & green & grass \\
sky & \frac{1}{8} & \frac{2}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\
is & \frac{1}{11} & \frac{1}{11} & \frac{2}{11} & \frac{3}{11} & \frac{1}{11} & \frac{2}{11} & \frac{1}{11} \\
blue & \frac{1}{8} & \frac{2}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\
my & \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \frac{4}{10} & \frac{1}{10} & \frac{1}{10} \\
favorite & \frac{1}{8} & \frac{2}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\
green & \frac{1}{8} & \frac{2}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\
grass & \frac{1}{8} & \frac{2}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8}
\end{array}\right]$$

Spam Transition Probabilities and Bayes' Theorem

- Treat each email as a sequence of words.
- Generate two transition matrices= One containing words occurring in spam and one for ham.
- Given an email find the probability of it being spam using Bayes' theorem:

$$P(Spam|Email) = \frac{P(Email|Spam)P(Spam)}{P(Email|Spam)P(Spam) + P(Email|Ham)P(Ham)}$$

- $P(Spam) = P(Ham) = 0.5$ since 'no prior belief'

Limitations,Challenges, and Possible Advancements

- Vectorization of transition matrix operations
- Dealing with words in email that aren't a state in the transition matrix by introducing an 'unkown' row and column or skipping
- Using log of likelihood as a features and using that with LIBSVM
- Multiclass classification of emails into work, school, promotional, job-related etc.
- Markov Chain with order $n > 1$
- Character-level Markovian Spam Filtering