Smart Inbox:A comparison based approach to classify the incoming mails

Jai Prakash Verma
Department of Computer Science & Engineering
Institute of Technology
Nirma University
Ahmedabad
Email: jaiprakash.verma@nirmauni.ac.in

Sapan H Mankad
Department of Computer Science & Engineering
Institute of Technology
Nirma University
Ahmedabad
Email: sapanmankad@nirmauni.ac.in

***Abstract*—Smart Inbox - this concept has recently been in demand due to well developing era of data analysis so as to make it comfortable for future prediction. Statistical Learning and Data Mining are the fields which are growing rapidly and have started capturing the attention of many of the business organizations to ease their work with the help of classification and prediction.In this paper,we have made an effort to use one of the classification methods, *Bayesian Learning* to categorize the incoming mails to a specific mail recipient. It uses the available knowledge to classify the training data and then predicts the status of the new incoming mail. We compare the formula based analytical result with that obtained through algorithms supported by Weka tool and suggest the most suitable way to use bayesian technique.**

I. Introduction

Data mining is a process of analyzing business data (often stored in a data warehouse) to uncover hidden trends and patterns and establish relationships [1]. Data mining is normally performed by expert analysts who use specialized software tools.One of its disciplines is using statistical techniques to discover subtle relationships between data items, and the construction of predictive models based on them,which is referred to as Predictive Analysis.

Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of object whose class label is unknown. The derived model is based on the analysis of set of training data (i.e., data objects whose class label is known). The derived model may be represented in various forms such as classification (IF-THEN) rules, decision trees, mathematical formulae, or neural networks.Hence the output of the classification depends upon the typs of classifier used. In our case,we have used probabilistic approach to find the classification of mails, which is represented in terms of probabilities of success or failure.

The essense of the Smart Inbox in making it *"smart"* is that it uses the approach of analyzing the available dataset so as to turn it into a simplified and knowledgeable form which predicts the behaviour of incoming mails and categorizes them into three classes (Highest,Medium and Lowest priority) through the classification produced by Bayesian approach.

A. Bayesian Classification

Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probabilities that a given tuple belongs to a particular class [2]. Bayesian classification based on Bayes theorem. Studies comparing classification algorithms have found a simple Bayesian classifier known as the naive bayes classifier to be comparable in performance with decision tree and selected neural network classifiers. Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases.

Naive Bayesian classifier assume that the effect of an attribute value on given class is independent of the value of the other attributes. This assumption is called class conditional independence. It is made to simplify the computations involved and, in this sense, is considered naive.

Sr. No.	Name	From Contact List?	Frequency in a week	Read or not Read?	% Read	Preference
1	Devendra Vashi	1	3	2	66.67	H
2	Dinajadav	0	7	5	71.43	H
3	Saurin Parikh	1	2	2	100.00	H
4	Parik Rana	0	1	0	0.00	L
5	Sapan Mankad	1	1	1	100.00	H
6	Preeti Katheria	1	2	1	50.00	M
7	Sonia Mittal	1	2	1	50.00	M
8	Priyanka Sharma	1	2	1	50.00	M
9	Sharada Valiviti	1	4	2	50.00	M
10	H. K. Patel	0	3	1	33.33	L
11	Gaurang rawal	1	1	1	100.00	H
12	Prakash Kadalia	1	1	1	100.00	H
13	Sujal Soni	0	1	1	100.00	H
14	Hiral Patel	0	5	2	40.00	M
15	Zunnun Narmawala	1	3	1	33.33	M
16	S. N. Pradhan	1	1	1	100.00	H
17	International ConferanceITS	0	1	0	0.00	L
18	Dingant Oza	0	1	1	100.00	H
19	K. Agrawal	1	1	0	0.00	L
20	Deepika Shukla	1	1	1	100.00	H

Fig. 1. Details of unique mails from a Live Inbox for a week

1) Bayes Rule: Bayes' Rule is a Rule of probability theory originally stated by the Reverend Thomas Bayes. It can be seen as a way of understanding how the probability that a theory is true is affected by a new piece of evidence. It has been used in a wide variety of contexts, ranging from marine biology to the development of "Bayesian" spam blockers for email systems. In the philosophy of science, it has been used to try to clarify the relationship between theory and evidence. Many insights in the philosophy of science involving confirmation, falsification, the relation between science and pseudoscience, and other topics can be made more precise, and sometimes

extended or corrected, by using Bayes' Theorem.
The Bayes Rule or Bayes Theorem is given by equation 1.

$$P(h_1|x_i) = \frac{P(x_i|h_1)P(h_1)}{(P(x_i|h_1)P(h_1) + P(x_i|h_2)P(h_2))} \quad (1)$$

Here $P(h_1|x_i)$ is called the posterior probability, while $P(h_1)$ is the prior probability associated with hypothesis h_1. $P(x_i)$ is the probability of the occurrence of data value x_i and $P(x_i|h_1)$ is the conditional probability that, given a hypothesis, the tuple satisfies it.

Where there are m different hypotheses we have equation 2:

$$P(x_i) = \sum_{j=1}^{m} P(x_i|h_j)P(h_j) \quad (2)$$

Thus, we have what follows in equation 3,

$$P(h_1|x_i) = \frac{P(x_i|h_1)P(h_1)}{P(x_i)} \quad (3)$$

Attribute	Value	Count			Probabilities		
		Lowest	Medium	Highest	Lowest	Medium	Highest
Email From Contact List	1 (Yes)	1	5	7	1/4	5/6	7/10
	0 (No)	3	1	3	3/4	1/6	3/10
% Read of E-Mail	(0,10]	3	0	0	3/4	0	0
	(11,20]	0	0	0	0	0	0
	(21,30]	0	0	0	0	0	0
	(31,40]	1	2	0	1/4	2/6	0
	(41,50]	0	4	0	0	4/6	0
	(51,60]	0	0	0	0	0	0
	(61,70]	0	0	1	0	0	1/10
	(71,80]	0	0	1	0	0	1/10
	(81,90]	0	0	0	0	0	0
	(91,100]	0	0	8	0	0	8/10

Fig. 2. Probability associated with attributes

Bayes rule allows us to assign probabilities of hypotheses given a data value, $P(h_j|x_i)$. Here we discuss tuples when in actually each x_i may be an attribute value or other data label. Each hi may be an attribute value, set of attribute values (such as a range), or even a combination of attribute values.

II. Smart Inbox Concept

Our goal was to classify a specific inbox that receives mails from various places [5].In order to execute any mining algorithm, we were required to have a dataset consisting of almost every possible details related to inbox.

A. Initial Preference Allocation

We have used an existing mailbox(not a fresh or newly created one) for our analysis.In this case,data were already available with us.The case is when a new mail account is generated,how can the preference of arriving mails be judged?Hence,we suggest the account holder(the mail box owner) to answer a trivial questionnarie which may include the details as mentioned below:

- **Association:**field of work with which the owner is associated.
- **Subscription:**newsletters and online resources which the owner wishes to subscribe.

We may consider these parameters to set up the initial preference for any new mail.

B. Pseudocode

We selected the inbox of the first author as our trial base,and followed the procedure as depicted in given algorithm.

1) Identify attributes
   - Whether the mail is in Contact List?
   - Whether the mail is moved to Spam Folder?
   - Whether the mail is responded?
   - Whether the mail is received frequently from the same source?
   - Whether the mail is read?
2) Determine the class label
   - Three classes based on the priority:Highest (H),Medium(M),Lowest(L)
3) Specify the priority rule:
   - Preference Rule (mail not from contact list)
     %Read in [66.66,100] ⇒ class:H
     %Read in [33.33,66.66)⇒ class:M
     %Read in [0,33.33) ⇒ class:L
   - Preference Rule (mail from contact list)
     %Read in [50,100] ⇒ class:H
     %Read in [25,50) ⇒ class:M
     %Read in [0,25) ⇒ class:L
4) Prepare training data from the above rules
5) Execute the Bayesian Classification algorithm on the unobserved sample to predict its class label.

Figure 1 shows the screenshot of the training data,which is prepared by following the presented pseudocode.The main task of the algorithm is to gather the important factors from the concerned mailbox and perform attribute selection which can be used to judge the importance of any incoming mail. As described here,it is observed that the training data consists of various binary attributes which evaluates the unclassified mail.The classification criteria is given by Preference Rule in order to classify the training tuples.The information obtained from these independent attributes is used to find the knowledge about the dependent attribute i.e;class label.

C. Analytical Implementation

To facilitate classification, we divided the *%Read* attribute into ten intervals.Figure 2 shows the counts and subsequent probabilities associated with the attributes. With these training data, we estimate the prior probabilities:

$$P(Lowest) = 4/20 = 0.2$$
$$P(Medium) = 6/20 = 0.3$$
$$P(Highest) = 10/20 = 0.5$$

By using these value and the associated probabilities of from_contact_mail and *%Read*, we obtained the following estimates:

$$P(t|Lowest) = 1/4 * 1/4 = 1/16 = 0.0625$$
$$P(t|Medium) = 5/6 * 2/6 = 5/8 = 0.2778$$

Unknown Instance	Expected Class Label(on the basis of Priority Rule)	Calculated Class Label (on the basis of Bayesian mathematical computation)	(on the basis of Weka Naive Bayes approach)	(on the basis of Weka - Simple Naive Bayes approach)	(on the basis of Weka - Naive Bayes Updateable approach)
t1 = (a , 1, 40)	M	M	M	H	M
t2 = (b , 0, 10)	L	L	L	L	L
t3 = (c , 1, 75)	H	H	H	H	H
t4 = (d , 1, 60)	H	H	M	M	M
t5 = (e , 0, 32)	L	L	L	L	L
Accuracy	-	100%	80%	60%	80%

TABLE I
COMPARISON OF VARIOUS BAYESIAN FLAVOURS

$$P(t|Highest) = 7/10 * 0 = 0$$

Combining the above probabilities, we get
Likelihood of being lowest priority e-mail $= 0.2 * 0.0625 = 0.0125$
Likelihood of being medium priority e-mail $= 0.3 * 0.2778 = 0.08333$
Likelihood of being highest priority e-mail $= 0.5 * 0 = 0$

We estimate $P(t)$ by summing up these individual likelihood values since newly arriving e-mail will be having either lowest,medium or highest preference:

$$P(t) = 0.0125 + 0.08333 + 0 = 0.09583$$

Finally, we obtained the actual probabilities of each event:

$$P(Lowest|t) = (0.0625 * 0.2)/0.09583 = 0.1304$$
$$P(Medium|t) = (0.2778 * 0.3)/0.09583 = 0.8696$$
$$P(Highest|t) = (0 * 0.5)/0.09583 = 0$$

Therefore, based on these probabilities, we classify the newly arrived e-mail with Medium preference because it has got maximum probability.

D. Result Interpretation

We use the training data values to classify unknown samples. For example, we considered five newly arrived mail in the inbox as shown in Table I,where the second value represents that the sender's email-id is available in the contact list and third value represents that *%Read* of the E-Mail that is calculated from the available inbox data.We have shown a sample calculation of the analytical computation of Bayesian formula.We have derived a classification model using the 20 records,tested the model on these five unknown samples and performed the comparison of the mentioned techniques. Weka supports various flavours of Bayesian Classification.As mentioned in [3],*NaiveBayes* approach estimates Posterior probability,*NaiveBayesSimple* uses a simple Naive Bayes classifier modelled by a normal distribution,and *NaiveBayesUpdateable* is the updateable version of NaiveBayes.The output produced by Weka for Simple Naive Bayesian is given in Appendix.

We found that the analytical approach is fully identical to our preference based prediction,whereas Naive Bayes approach of Weka gives the most accurate result among the three.

III. FUTURE SCOPE

We have implemented this concept at a primitive level using mathematical approach.It can be extended to achieve better efficiency using any of the more efficient classifiers such as neural networks.ANN with back propogation can be more capable to extract information from the training data. Further,different attributes may be assigned different weights to determine the priorities dynamically. A comparison based analysis of several classifiers on a relatively large dataset can give significantly reliable output.

IV. CONCLUSION

Bayesian Classification is a probabilistic approach that predicts an unobserved sample using a set of training data.It always shows the output in binary format. We found here that Simple Naive Bayes approach is more suitable for analyzing such types of data.Although better performance based comparison can be achieved in case of large volume of data.

REFERENCES

[1] Margrate H. Dunham. *Data Mining : Introductory and Advance Topics*. Pearson Education.
[2] Jiawei Han and Micheline Kamber. *Data Mining : concept and Techniques*. Elseveir Pubilcation.
[3] Henry Xiao. Analysis of dataset 1. Technical report, Queen's University.
[4] Weka 3.6.0 Waikato University,NewZealand http://www.cs.waikato.ac.nz/ ml/weka/
[5] Google Priority Inbox http://mail.google.com/support/bin/answer.py?hl=en&answer=186531

APPENDIX

EXECUTION OF NAIVE BAYES ALGORITHM IN WEKA

In this appendix,we present the analysis of one of the Bayesian algorithms using Weka discussed in this paper.It depicts the statistical analysis of Naive Bayes approach.

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayesUpdateable
Relation:     boxData
Instances:    25
Attributes:   4
              contact
              frequency
              read
              preference
Test mode:    evaluate on training data

=== Classifier model (full training set) ===

Naive Bayes Classifier

                Class
Attribute            H      L      M
               (0.46) (0.25) (0.29)
====================================
contact
  mean            0.75 0.1667 0.8571
  std. dev.      0.433 0.3727 0.3499
  weight sum        12      6      7
  precision          1      1      1

frequency
  mean           2.8333      2 2.8571
  std. dev.      4.6518 3.0551 0.9897
  weight sum         12      6      7
  precision           2      2      2

read
  mean           2.4306 5.5556 0.5952
  std. dev.       5.758 9.2128  1.458
  weight sum         12      6      7
  precision      4.1667 4.1667 4.1667


Time taken to build model: 0 seconds

=== Predictions on training set ===

 inst\#, actual, predicted, error, probability distribution
     1     1:H        3:M      +   0.141  0.015 *0.844
     2     1:H        1:H         *0.704  0.296  0
     3     1:H        3:M      +   0.119  0.015 *0.866
     4     2:L        2:L          0.401 *0.509  0.091
     5     1:H        1:H         *0.617  0.077  0.305
     6     3:M        3:M          0.119  0.015 *0.866
     7     3:M        3:M          0.119  0.015 *0.866
     8     3:M        3:M          0.119  0.015 *0.866
     9     3:M        3:M          0.141  0.015 *0.844
    10     2:L        3:M      +   0.207  0.226 *0.567
    11     1:H        1:H         *0.617  0.077  0.305
    12     1:H        1:H         *0.617  0.077  0.305
    13     1:H        2:L      +   0.401 *0.509  0.091
    14     3:M        3:M          0.207  0.226 *0.567
    15     3:M        3:M          0.141  0.015 *0.844
    16     1:H        1:H         *0.617  0.077  0.305
    17     2:L        2:L          0.401 *0.509  0.091
    18     1:H        2:L      +   0.401 *0.509  0.091
    19     2:L        1:H      +  *0.617  0.077  0.305
    20     1:H        1:H         *0.617  0.077  0.305
    21     3:M        3:M          0.405  0.06  *0.535
    22     2:L        2:L          0.312 *0.688  0
    23     1:H        1:H         *0.99   0.01   0
    24     1:H        3:M      +   0.452  0.055 *0.492
    25     2:L        2:L          0.014 *0.986  0

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances          18               72%
Incorrectly Classified Instances         7               28%
Kappa statistic                          0.5793
Mean absolute error                      0.2753
Root mean squared error                  0.3712
Relative absolute error                 64.8058%
Root relative squared error             80.7519%
Total Number of Instances               25

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.583     0.077      0.875     0.583     0.7        0.821    H
                 0.667     0.105      0.667     0.667     0.667      0.895    L
                 1         0.222      0.636     1         0.778      0.921    M
Weighted Avg.    0.72      0.124      0.758     0.72      0.714      0.866

=== Confusion Matrix ===

 a b c   <-- classified as
 7 2 3 | a = H
 1 4 1 | b = L
 0 0 7 | c = M
```