International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878, Impact Factor (2014): 3.05

Content Based Text Classification Using Morkov Models

Khalid Hussain Zargar[1], Manzoor Ahmad Chachoo[2]

Department of computer Science, Mewar University,
Gangrar Chitorgarh, Rajasthan-312901

[2]Department of Computer Science, University of Kashmir,
Hazratbal Srinagar-190001

Abstract: ***Text categorization is the task of assigning predefined category to a set of documents. Several different models like SVM, Naïve Bayes, KNN have been used in the past. In this paper we present another approach to automatically assign a category to a document. Our approach is based on the use of Markov Models. We consider text as bag of words and use Hidden Markov Model to assign the most appropriate category to the text. The proposed approach is based on the fact that while creating documents the user uses the specific vocabulary related to the particular category. Hidden Morkov models have been widely used in automatic speech recognition, part of speech tagging, and information extraction but has not been used extensively for text categorization.***

Keywords: Text Classification, Information gain, HMM, Text Processing, Viterbi Algorithm, Precision, Recall.

1. Introduction

The problem of text classification has been a recent research area in the field of library science, computer science and in many other areas. Now a days having a large number of digitized text material manual classification becomes almost impractical consuming a lot of time and resources. Now is the time to find new and automatic approaches for text classification. In the recent past various methods of automatically classifying the text have been developed using machine learning algorithms like Naïve Bayes, Artificial Neural Networks, KNN, SVM, etc...A machine learning algorithm takes as input a set of labeled example documents (where the label indicates which category the example belongs to) and attempts to infer a function that will map new documents into their categories. In this paper we describe the process of automatic text classification based on Hidden Morkov model. "A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states." This approach of text categorization takes the same method of representing the documents as bag of words. HMMs has been used in many text and speech related applications like information retrieval, information extraction, text summarization but has been widely used for text classification. The purpose of this approach is to consider only the content of the document and not the structure of the document for classification purposes.

2. Hidden Morkov Model

"A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states."

An HMM can be defined as a 5-tuple (S, V, π, A, B).

S: Number of states in the Model (The catagories in our case). There is a finite set of states in a model. The states in an HMM are hidden, but there is a lot of significance to these states in defining an HMM. We denote the individual states as S1, S2, S3, …, Sn. S = { S1, S2, S3, … , Sn}.

V: Number of distinct symbols observable in states. These symbols correspond to the observable output of the system that is being modeled. We denote the individual states as v1, v2, v3, …, vM.

V = { v1, v2, v3, …, vM}.

A: State transition probability distribution A is transition array that store the state transition probabilities.

A={aij }, where aij stores the probability of state j following state i.

aij= P(qt=Sj/qt-1=Si), i ≥1 and j ≥ N the probability of moving from state Si to Sj at time t At each time t, a new state is entered which depends on the transition probability distribution of the state at time t–1 .Transition to the same state is also possible. An important point about transition probabilities is that they are independent of time; the probability of moving from state Si to state Sj is independent of time t.

B: Observation symbol probability distribution B = {bj(k)} is the output symbol array that stores the probability of an observation Vk being produced from the state j, independent of time t. Observation symbol probability or Output emission Probability estimates are also independent of time. The probability of a state emitting a particular output symbol does not vary with time.

B = { bj(k) } , bi(k) = P(xt = vk/qt = Sj) 1≤ j ≤ N and 1≤ k ≤ M

bi(k), the probability of emitting symbol vk, when state Sj is entered at time t. After each transition is made, a symbol is

output based on the output probability distribution, which depends only on the current state.

π: Initial state distribution.

π = {Πi} is the initial probability array that stores the probability of the system starting at state i in an observation. It is the probability of state Si being the start state in an observation sequence.

π = { Πi }, Πi = P (q1 = Si), 1≤ i ≤ N

Πi, the probability of being in state I at time t=1

A complete specification of an HMM consists of the above five elements, {S, V, π, A, B}. We usually use a compact notation λ=(A, B, π) to represent the above complete parameter set of HMM. As each aij represents the probability P (Sj/Si), the laws of probability require that the values of the outgoing arcs from a given state must sum to one. Same laws of probability apply to Initial Probabilities and Output Emission Probabilities.

2.2 Problems of Hidden Markov Model

To solve the problem of text categorization efficiently we need to tackle the following three issues related to HMM.

Likelihood:

Given the model parameters, compute the probability of a particular output sequence i.e Given an observation sequence O and a model λ,what is the probability of the observation sequence, P(O|λ)?

P(O|λ)= P(O1,O2,…, OT | λ) = ?

Decoding:

Given the model parameters, find the most likely sequence of (hidden) states which could have generated a given output sequence. i.e Given an observation sequence O and a model λ,

S* = arg max S=(s1, s2, …, sT) P(S, O | λ) = ?

Training:

Given a training sequence O, find a model λ, specified by parameters (A, B, π) to maximize P(O|λ) (we assume for now that Q and V are known).

P (O | λ = (A, B ,π)) < P (O | λ ' = (A' , B ' ,π'))

λ* = argmaxλ P(O|λ)

There are specific algorithms for each problem that explain the best way to solve them. The problem of likelihood is solved using the Forward and Backward iterative algorithms. The second problem is solved using the Viterbi Algorithm, also an iterative algorithm that outputs the best path by sequentially considering each observation symbol of O. The last problem which deals with training an HMM, can be solved by using Baum-Welch or Maximum Likelihood Estimation (MLE). The choice between these two algorithms can be made using the training data available for the learning process.

Forward Backward Algorithms

- **Forward Probability Calculation**

Define $\alpha_t(i) = P(Qt = Si , O_1, \ldots, O_t)$

Given a sequence of length T, and an HMM with transition matrix A and output probability B,

1. Initialization $\alpha_1(i) = \Pi(i) * B_i (o_1)$ for every I (i.e., every node S_I)
2. Iteration Step $\alpha_{t+1}(i) = (\sum_{j=1,N} \alpha_t(j) A_{j,i}) * B_i(o_{t+1})$ for every i

- **Backward Probability Calculation**

Define $\beta_t (i) = P(O_{t+1}, \ldots, O_T \mid Q_t = S_i)$ Given a sequence of length T, and HMM as before,

1. Initialization $\beta_t(i) = 1$
2. Iteration $\beta_t(i) = \sum_j A_{i,j} * Bj (o_{t+1}) * b_{t+1}(j)$

By the forward and backward variables, compute the probability of observation sequence, getting

$P(O|\lambda)= \sum_{i=1,N} \alpha_t(i) \beta_t(i)$ 1<=t<=T-1

Viterbi Algorithm:

The Viterbi algorithm is a dynamic programming algorithm that computes the most likely state transition path given an observed sequence of symbols. It involves the computation of

$\delta_j(t) = \max [P(C_1 .. C_{t-1} , N_1 .. N_{t-1}, C_t = j \mid \pi, A, B)] \; 1 \le t \le K$

C1...Ct-1

δj(t) identifies the probability of being in state j at time t. This is based on the prior hidden categories $C_1 .. C_{t-1}$ and the prior observed symbols $o_1 .. o_{t-1}$. The computation of this algorithm is done recursively at starting at the initial state at time t = 0 and looping till t = K.

$\delta_j(1) = \pi_j$ represents the initial state of the HMM
$\delta_j(t + 1) = \max [\delta_j(t)aijbjn] \; 1 \le j \le J$
i = 1...J
$\Psi_j(t + 1) = \text{argmax} [\delta_j(t)aijbjn] \; 1 \le j \le J$

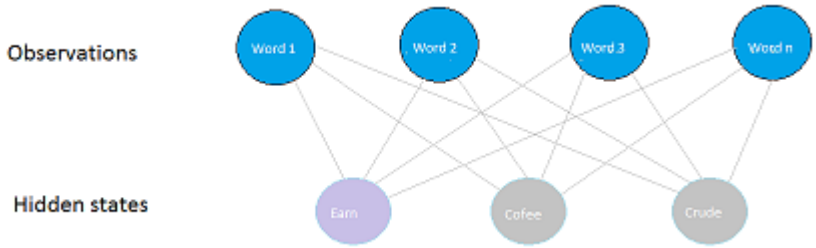

Figure 1: Text Categorization states in HMM

Related Works

HMM has been a useful statistical model for many applications in natural language processing for example part of speech tagging, speech recognition etc. In the field of information retrieval the model has been used very effectively - Miller et al 1999 and elke and Shauble 1994 (Information retrieval) Conroy and O'leary 2001 text summarization. Frasconi used HMM to classify multipage documents. The classification of the documents was based on the structure of the documents. Kwan yi and Jamshed Behesti used HMM to classify documents based on LCC(library of congress classification) scheme. Ludovic denoyer discussed the passage model for classification using HMM.

HMM based Text Classification

Document Preprocessing

In case of supervised learning document processing involves the following steps

- Removing the punctuation and numbers from the text.
- Tokenization and stemming, i.e. separating words by spaces and taking the root
- Removing the stop words like a, the ,and etc

After the documents are processed the documents are used feature selection. Since after removing the stop words from the dataset the number of words in the bag of words representation in the document set are huge. Feature selection involves reducing the number of features or words representing a particular category. Various methods have been used for feature reduction like chi-square, gini index, TFIDF etc. In this paper we have used the information gain approach of reducing the number of words for a given category.

$IG(t)=-\sum_{i=1...m} p(Ci)\log p(Ci) + P(t) \sum_{i=1...m} p(Ci/t) \log p(Ci/t) + P(t`) \sum_{i=1...m} P(Ci/t`)\log p(Ci/t`)$

- P(Ci) is the probability of category Ci
- P(t) is the parobability that t occurs in collection
- P(Ci/t) is the probability that a category is Ci given the term t appears
- P(Ci/t`) is the probability that the category is Ci given the term t does not appear

Table 1: Weighted Information Gain for Subset of Reuters Corpus

FACTORS	0.025
FALL	0.026
FARM	0.076
FARMER	0.025
FBC	0.029
FEBRUARY	0.091
FEE	0.029
FIGURES	0.025
FILES	0.029
FINANCE	0.025
FIRB	0.058

3 Building the Classifier

The HMM is trained using Reuters dataset on several categories. The goal of this phase is to build an HMM using preprocessed training data as input. The output of this learning process is the five parameters of the HMM, (S, V, π, A, B). We estimate the model parameters using the maximum likelihood estimate. Three sets of probabilities calculated using MLE is:

State transition probability distribution A

A={aij }: aij stores the probability of state j following state i.

N(si , sj): Number of times we move from state si to state sj
N(si): Number of transitions from state si.
V: entire vocabulary (all output symbols).
aij = P(qt = sj / qt-1 = si) = (N(si , sj) + 1) / (N(si) + N), i ≥1 and j ≥ N.

Transition probability gives the probability of changing from one particular category to another.

Observation symbol probability distribution:

B = {bj(k)} is the output symbol array that stores the probability of an
observation vk being produced from the state j, independent of time t.
B = {bj(k)} , bi(k) = P(xt = vk/qt = Sj) 1≤ j ≤ N and 1≤ k ≤ M.
N(k,i): Number of times state i has seen output symbol k.
N(s): Number of occurrences of state i.
V: entire vocabulary (all output symbols)
bi(k) = P(k|i) = (N(k,i) + 1) / (N(i) + |V|)
Initial state distribution: Π

Π= {Πi} is the initial probability array that stores the probability of the system starting at state si in an observation.
π = {Πi}, Πi = P (q1 = Si), 1≤ i ≤ N
Πi , the probability of being in state si at time t=1.
N(si): Number of times we start from state si.
N: Number of input sequences.
Πi = N(si) / N

Initial state probability matrix in our HMM model gives the probability for every particular catagory to be the first catagory in a sequence.

Categories for a block of text are determined by applying the Viterbi algorithm determined from the text. Viterbi algorithm involves multiplying many probabilities together. Since each of these numbers is less than one, we can end up working with numbers that are tiny enough to be indistinguishable from zero. To avoid this we worked with log of probabilities. The most likely path through the HMM is calculated and the categories are determined to be the unique collection of all categories from the path.

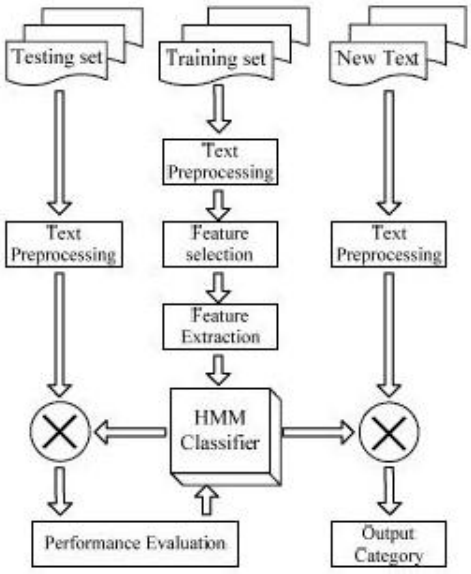

Figure 2: Text Categorization Process

Testing the Model

The model created is tested against the new document to be classified. The goal of this phase is to compute the most probable sequence of states for a given sequence of outputs using the HMM that we built from the training data. After the document is preprocessed the text set consists of all the tokens (output symbols) from the test set and the category set consists of C={c1,c2…cn}for which the model was created. After setting the HMM for each category and calculating P(wi/ λ) for each wi and using the viterbi algorithm for comparing the probability C_{max}=argmax P(wi/ λ) the final category of test set can be computed.

Table 2: Logarithmic Computations using viterbi algorithm - final category in bold

Time	*Token*	*Acq*	*Crude*	*Coffee*
0	Factors	-0.08	-0.48	-0.17
1	Agriculture	-.0045	-.0056	-0.038
2	crude	-9.84	-2.823	-3.677
3	File	-16.11	-12.15	-19.40
4	Finance	-19.86	-19.15	-25.19
5	Eugene	-25.91	-23.63	-27.68
6	Analyst	-30.26	-25.91	-34.18
7	Expert	-32.84	-26.69	-34.29

Performance Evaluation

The performance of text categorization model built is evaluated based on standard precision, recall, and F1 values. Precision, recall, and F1 values are calculated on the test data from the collection. Let TP be the number of true positives, i.e., the number of documents that both experts and the model agreed as belonging to the same category. Let FP be the number of false positives, i.e., the number of documents that are wrongly categorized by the model as belonging to that category.
Precision is defined as:

$$precision = \frac{TP}{TP + FP}$$

Let FN be the number of false negatives, that is, the number of documents that are not labeled as belonging to the category but should have been.

Recall is defined as:

$$recall = \frac{TP}{TP + FN}$$

The harmonic mean of precision and recall is called the F1 measure, and is defined as

$$F1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}}$$

4 Conclusion and Future works

We have presented a text classification model based on the hidden morkov model. The model was tested on the Reuters corpus. Only three categories from the Reuters corpus viz Acq, Crude, Coffee with 175 training documents and 60 test documents was used for training and testing the classifier respectively. The model uses only unigram tokens for training and testing. Since limited number of documents and terms were used, the performance of the model was satisfactory when compared to other machine learning text classification methods. However the performance of the model can be improved by using a large corpus of labeled training dataset. The performance can also be improved by considering two or more tokens together having some sort of semantic relation between them. By taking two or more tokens together the accuracy of the HMM based classifier can be improved. Also the structure of the documents under consideration can also be used to enhance the performance of HMM based text classification.

References

[1] Kwan yi and jamshid Behishti “Text categorization Model based on Hidden Morkov Model” CAIS/ACSI 2003R. Caves, Multinational Enterprise and Economic Analysis, Cambridge University Press, Cambridge, 1982. (book style)
[2] Ludovic Denoyer and Hugo Zaragoza (2001)“HMM based Passage Models for Document Classification and Ranking. 23rd BCS European Annual Colloquium on Information Retrieval, 2001
[3] Phil Blussom “A tutorial on Hidden Morkov Model” - 2004.
[4] P. Frasconi(2002) “ Hidden Morkov Model for Text Catagorization for Multipage Documents
[5] Honglak Lee and Andrew Y. Ng “ Spam Deobfuscation using Hidden Morkov Model”
[6] Flavia A. Barros, Eduardo F. A. “Hidden Markov Models and Text Classifiers for Information Extraction on Semi-Structured Texts”
[7] Max Bramer “Principles of Data mining” p-135- 152 2nd edition Springer.
[8] Sebastiani, F. (2002). Machine learning in automated text categorization. ACM Computing Surveys, vol. 34, no 1, p. 1-47.
[9] Grezegorz Szymanski and Zygmunt Ciota “Hidden Markov Models Suitable for Text Generation”
[10] J. Jiang and C. Zhai, \Extraction of coherent relevant passages using hidden markov models," ACM Trans. Inf. Syst., vol. 24, no. 3, pp. 295{319, 2006.
[11] D. R. H. Miller, T. Leek, and R. M. Schwartz, \Bbn at trec7: Using hidden markov models for information retrieval," in In Proceedings of TREC-7, 1999, pp. 133{142.
[12] Yiming Yang, Jan O. Pederson “A comparative study on feature selection in text categorization”
[13] Yang Jian, Wang Hai-hang. "Text Classification Algorithm based on Hidden Markov Model"."Journal of Computer Applications", pages 2348-2350, 2361, 2010