Mingyao Pang\_MXP180013 Ni Wei\_NXW180000

**CS 6375.002 Machine Learning Final Project Report**

Contents

[1. Problem Statement 1](#_Toc7950186)

[2. Approaching Methods 1](#_Toc7950187)

[3. Background 1](#_Toc7950188)

[3.1. Naïve Bayes 1](#_Toc7950189)

[3.2. Hidden Markov Model 2](#_Toc7950190)

[3.3. The Viterbi Algorithm in HMM 2](#_Toc7950191)

[3. Dataset 3](#_Toc7950192)

[4. Proposed Model 4](#_Toc7950193)

[4.1.Naïve Bayes 4](#_Toc7950194)

[4.2. Markov Hidden Model Combined with Naïve Bayes 4](#_Toc7950195)

[5. Experiment 5](#_Toc7950196)

[6. Result 5](#_Toc7950197)

[6.1. Result from Naïve Bayes model 5](#_Toc7950198)

[6.2. Result from the combination of HMM and Naïve Bayes model 5](#_Toc7950199)

[7. Conclusion 6](#_Toc7950200)

# 1. Problem Statement

This project aims to design an algorithm to recognize the offline handwriting documents with high accuracy. In this project, only lowercase letters are considered.

# 2. Approaching Methods

In the original design, Naïve Bayes model was proposed to tackle the problem. By using the Naïve Bayes, we achieved the accuracy rate as 62.88%. To improve it, we redesigned our model by combining Hidden Markov Model with the Naïve Bayes model, where the relationships among letters of one word are taken into consideration. With this ensemble model, we improved the accuracy rate to 98.86%.

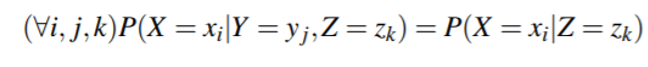
# 3. Background

## 3.1. Naïve Bayes

Naïve Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

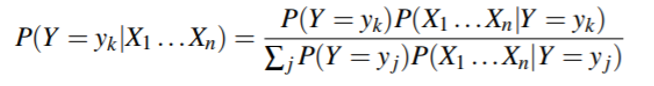
Conditional Independence assumption in NB is given as below:

Definition: Given random variables X, Y and Z, we say X is conditionally independent of Y given Z, if and only if the probability distribution governing X is independent of the value of Y given Z; that is

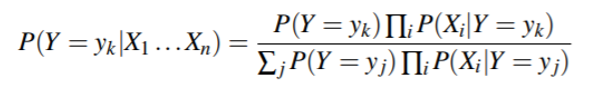


Naïve Bayes is derived as below:

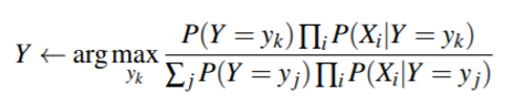
Assuming Y is any discrete-valued variable, and the attributes X1 ...Xn are any discrete or real-valued attributes. Our goal is to train a classifier that will output the probability distribution over possible values of Y, for each new instance X that we ask it to classify.



When we assume Xi are conditionally independent given Y, the above equation can be rewritten as follows:

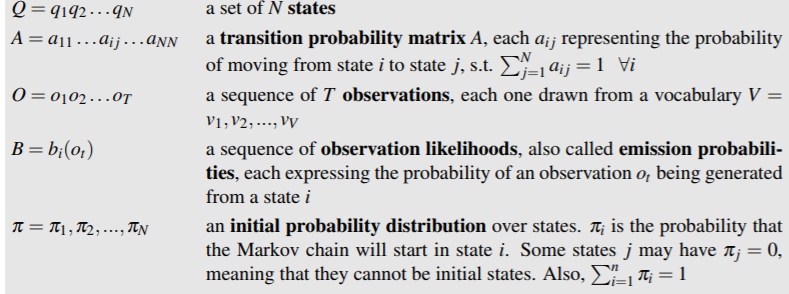


If we are interested only in the most probable value of Y, then we have the Naïve Bayes classification rule:



## 3.2. Hidden Markov Model

A hidden Markov model (HMM) allows us to talk about both observed events Hidden Markov model and hidden events that we think of as causal factors in our probabilistic model. An HMM is specified by the following components:



In addition, a first order HMM instantiates two simplifying assumptions:

1. Markov Assumption: the probability of a particular state depends only on the previous state



2. Output Independence: the probability of an output observation oi depends only on the state that produced the observation qi and not on any other states or any other observations.



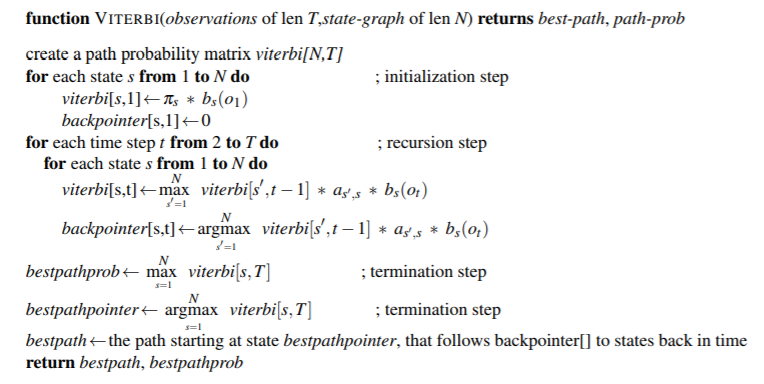
## 3.3. The Viterbi Algorithm in HMM

Viterbi Algorithm is targeted for the decoding problem. Given A, B (where A is the transition matrix and B is the emission probability), and a sequence of observations O = o1, o2, o3, …, oT, find the most probable sequence of states Q = q1q2q3…qT.

The idea is to process the observation sequence left to right, filling out the trellis. Each cell of the trellis, vt(j), represents the probability that the HMM is in state j after seeing the first t observations and passing through the most probable state sequence q1,...,qt−1, given the automaton λ.



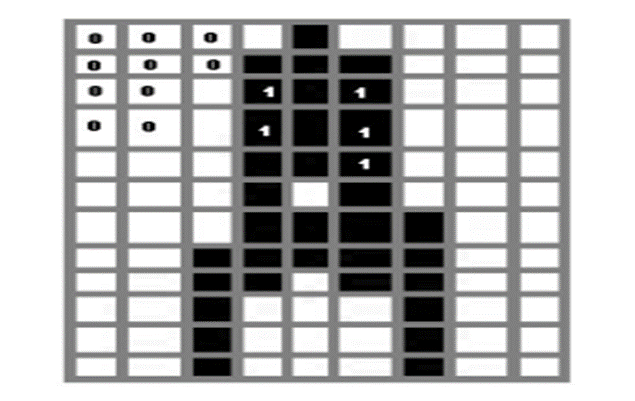
Viterbi uses dynamic programming to fills each cell recursively. Algorithm is given below:



# 3. Dataset

We used the dataset collected by Rob Kassel at MIT Spoken Language Systems Group. The dataset was cleaned and normalized properly. The first letter of each word was capitalized and rest were lowercase, therefore, the first letter was removed since it is out of the scope of the study.

In the dataset, each letter is well prepared and represented as a 16 x 8 array of pixels. In the array, 1 denotes black, and 0 denotes white.



Each data point represents a lowercase letter, with the following features:

1. Id: each letter is assigned with a unique integer id

2. Letter: a\_z

3. next\_id: id for next letter in the word, if it is the last letter, -1 was filled.

4. word\_id: each word is assigned with a unique integer id (not used)

5. position: position of letter in the word (not used)

6. fold: 0-9 cross-validation fold

7. p\_i\_j: 0/1 -- value of pixel in row i, column j

Dataset link:

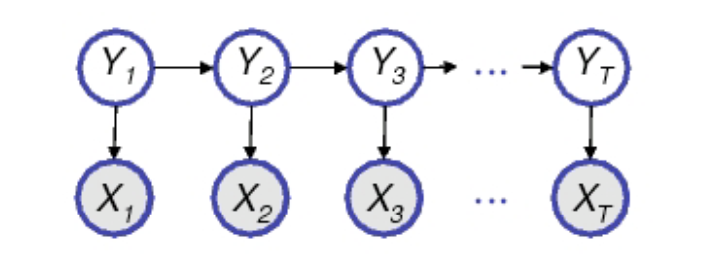
<http://ai.stanford.edu/~btaskar/ocr/>

# 4. Proposed Model

## 4.1.Naïve Bayes

Each letter image is represented as a 16 x 8 array of pixels. In the array, 1 denotes black, and 0 denotes white. We assume that each pixel in a letter is conditional independent given the letter, therefore the model for the Naïve Bayes can be represented as the following equations:

## 4.2. Markov Hidden Model Combined with Naïve Bayes



Hidden variable: Yt  is the final predicted letter, which makes up 26 possible states in our model.

Observed variable: Xt  is the predictions from Naïve Bayes. There are 26 possible predictions from Naïve Bayes.

Transition matrix

it is a 26 x 26 matrix to denote the relationship between letter and letter. It is calculated by the following equation:

where P(Yt) =

Emission probability

p(Xt|Yt) denotes the probability of each predicted letter by naïve Bayes given that the true value is Zt. It is calculated by the following equation:

# 5. Experiment

Viterbi Algorithm is used to predict the most probable letter sequence of Y, which is the most likely word. 10-fold cross validation is used to calculate the accuracy. In details, the total data set is divided into 10 folds, among which 9 folds are used for training, leaving one out for testing. This procedure is repeated 10 times such that we will get 10 models with 10 accuracy rate, and the final accuracy rate is calculated by taking the average of the 10.

# 6. Result

Following is the 10-fold cross validation

## 6.1. Result from Naïve Bayes model

|  |  |
| --- | --- |
| Fold used for testing | Accuracy |
| 1 | 62.72% |
| 2 | 61.99% |
| 3 | 63.70% |
| 4 | 62.69% |
| 5 | 62.07% |
| 6 | 63.03% |
| 7 | 61.15% |
| 8 | 64.58% |
| 9 | 63.53% |
| 10 | 61.30% |
| Average | 62.68% |

## 6.2. Result from the combination of HMM and Naïve Bayes model

|  |  |
| --- | --- |
| Fold used for testing | Accuracy |
| 1 | 98.85% |
| 2 | 99.01% |
| 3 | 98.83% |
| 4 | 98.77% |
| 5 | 98.63% |
| 6 | 98.63% |
| 7 | 99.07% |
| 8 | 98.73% |
| 9 | 98.84% |
| 10 | 98.70% |
| Average | 98.78% |

# 7. Conclusion

As we can tell from the result, through combining the HMM model into NB model, we raised up the accuracy from 62.68% to 98.78%. The result also confirms our conjecture that there is a strong relationship between letter and letter within one word.