

---

# Determining the Probabilities of Hand Writing Formations Using PGM'S

---

**Manish Reddy Challamala**  
PersonId: 50289714  
Department of Computer Science  
University at Buffalo  
Buffalo, NY 14214  
[manishre@buffalo.edu](mailto:manishre@buffalo.edu)

## Abstract

To develop a Probability Graphical Model (PGM's) to determine the best model for the given data and to infer the Maximum a-posterior probability estimate to obtain unobserved quantity for empirical data.

## 1 Introduction

The goal of the project is to develop a Probabilistic Graphical Model (PGM's) for the given handwriting patterns and to find the best Bayesian setting for the given features.

The features are obtained from two different sources:

1. Marginal and Conditional probability tables are provided for the bigram 'th'.
2. MNIST Dataset.

The code is implemented in five phases:

For bi-gram "th":

- We need to find the pair-wise correlations and in-dependencies that exists between six features. We can find independencies between  $x_i$  and  $x_j$  by using the following formula
$$\Sigma((abs P(x, y) - P(x) P(y)))$$
Here  $x$  is  $x_i$  and  $y$  is  $x_j$   
 $P(x, y)$  is Joint Probability.  
 $P(x)P(y)$  is the product of marginal probabilities.
- We need to construct several Bayesian networks and find the high probability and low probability Bayesian network among the networks by using the goodness score like K2 score
- Convert the high probability Bayesian network into Markov network using moralization and find the time taken by the Bayesian and Markov network to infer a query.

For MNIST Dataset:

- For MNIST the sample data for the features f1 to f9 are given and we need to construct several Bayesian networks to evaluate the goodness score of the network to determine high probability model.
- Infer the conditional probability distribution tables of the model.

## 2 Theory

### 2.1 Bayesian network

- Bayesian network is probabilistic graphical models for representing the multivariate probability distributions, in which nodes represent the random variables and the edges represent conditional probability distributions [CPD's] tables between random variables, which are used to calculate the probability dependencies between the variables.
- Bayesian networks are also called as belief or causal networks because they are directed acyclic graphs [DAG], even with a change of CPD at one node can affect the whole networks performance.
- The probability distribution is defined in the form of factors:

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^N P(X_i | \Pi X_i)$$

Where  $\prod_{i=1}^N P(X_i | \Pi X_i)$  is P (node | parent(node))

### 2.1 Markov network

- Markov network is undirected acyclic graphs which are similar to the Bayesian network. As the graphs are undirected, instead of edges they have cliques which connect each node with their neighboring nodes.
- The probability distribution is defined in the form of factors of potential functions. Which can be written in the log-linear model.
- As there is no topological order for the Markov network, we don't use the chain rule instead of potential functions for each clique in the graph.
- Joint distribution in the Markov network is proportional to the product of clique potentials
- The conditional probability is defined as

$$P(y | \theta) = \frac{1}{Z(\theta)} \prod_c \phi(y_c | \theta_c)$$

Where  $Z(\theta)$  is the partition function derives as summation of products of all potential factors.

### 2.2 Moralization

- Moralization is the method to get the moral graphs, which finds the equivalent undirected graph form a directed graph.
- Moralization is adding the edges between the unlinked nodes of a parent which have the common child.

### 3 Experimental Setup:

#### 3.1 Task1: Finding Pair-wise Correlations and Independencies

We are calculating the pair-wise correlations and independencies for all possible combinations of  $x_i$  and  $x_j$ .

$$\text{Correlation} = \Sigma((\text{abs } P(x_i, x_j) - P(x_i)P(x_j)))$$

Where  $P(x_i, x_j)$  is the joint probability distribution.

$P(x_i)P(x_j)$  is the product of marginal distributions.

The correlation values are stated below:

Table 1: Pair-wise Correlation values and Independencies.

		GIVEN				
	X1	X2	X3	X4	X5	X6
X1				0.119		0.1666
X2	0.159		0.2982	0.115	0.36	0.1749
X3		0.218			0.115	0.0948
X4	0.119					0.1432
X5		0.129	0.1156			
X6	0.160		0.094	0.094		

- By placing the threshold value as 0.15 we can find strong and weak correlation relationship between the nodes. Lower the value the higher the correlation.
- Strong Correlations:  $(x_4 | x_1)$ ,  $(x_5 | x_2)$ ,  $(x_6 | x_3)$ ,  $(x_6 | x_4)$ ,  $(x_3 | x_5)$ ,  $(x_3 | x_6)$
- Weak Correlations:  $(x_6 | x_1)$ ,  $(x_3 | x_2)$ ,  $(x_2 | x_3)$ ,  $(x_1 | x_4)$ ,  $(x_2 | x_5)$ ,  $(x_2 | x_6)$
- From the above values we can infer that  $(x_6, x_3)$  are strongly correlated in bi directional way and these values are dependent on each other.
- Similarly, for the weak correlations the  $x_2$  and  $x_3$  values are independent to each other and we can also observe that node  $x_2$  is appearing as a weak correlation node for most of the other nodes and can conclude that  $x_2$  node as good correlation with  $x_1$  node when compared to other nodes.

#### 3.2 Task2: Constructing the Bayesian networks and finding the goodness score

- In this task, we are constructing the five Bayesian networks using the threshold value calculated from the above task1.
- Creating a Bayesian network by using 'pgmpy. models' – Bayesian Model function and adding the nodes and corresponding edges to the model.
- Adding the CPD tables of the edges using 'Tabularcpd' -add\_cpds function.
- After instantiating the model object, we are performing the Ancestral or forward sampling.

### 3.2.1 Ancestral or Forward Sampling

- Given a Bayesian network with probability distributions of nodes  $x_1$  to  $x_6$ , we sample each variable in topological order.
  - Start sampling the nodes with no parents and take the index of the highest value of that node and sample the next child nodes based on the sampled values at the first step.
  - To sample a discrete distribution we split  $[0,1]$  interval into bins whose sizes are determined by the probabilities  $P(x^i)$ ,  $i = 1, 2, 3, \dots, k$
  - Generate a sample  $s$  uniformly from the interval
  - If  $S$  is the  $i^{\text{th}}$  interval then sampled value is  $x^i$
- 
- After generating the sample data for the five models we are combining the five data sets and calculating the goodness score for by using the k2 function.

Table2: K2 Scores for Bayesian Models

K2 Score	Values
Model 1	-31949.32
Model 2	-32018.68
Model 3	-32090.66 [LOW PROBABILITY MODEL]
Model 4	-31947.80 [HIGH PROBABILITY MODEL]
Model 5	-32038.93

- By comparing the goodness score of the five Bayesian models we can observe that model 4 is the High probability model where the parent node is  $x_1$  dependent on  $x_4$  and  $x_6$ , the child node  $x_4$  is dependent on its descendant  $x_2$ , which is dependent on  $x_5$  and the node  $x_5$  is connected with  $x_3$ .
- Edges of high probability model:  $(x_4, x_1), (x_1, x_2), (x_1, x_6), (x_2, x_3), (x_3, x_5)$
- Edges of low probability model :  $(x_1, x_4), (x_1, x_6), (x_4, x_2), (x_2, x_5), (x_5, x_3)$

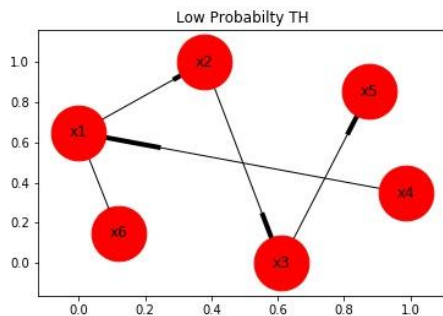


Figure1: Low Probability 'th' model

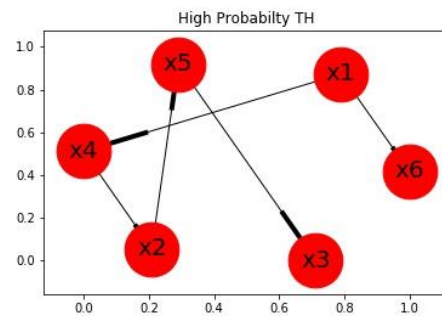


Figure2: High Probability 'th' model

### 3 Task 3: Converting Bayesian network to Markov network

#### 3.1 Converting to Markov network

- From the above task consider the high probability 'th' model and convert it into a Markov network using the moralization.
- `Model_name.to_markov_model()` predefined function is used to convert the Bayesian network into a Markov network.

#### 3.2 Inferring the query

- Some conditional probability inference algorithms are stated below:
  1. Variable Elimination
  2. Belief Propagation
  3. Variational Approximations
  4. Markov chain Monte Carlo (MCMC)
- What is meant by inference in Bayesian network?  
Given a query (joint probability distribution) we are getting the probabilities for all other specified nodes whose sum should be 1.
- In this process belief propagation and Variable elimination, algorithms have been used to infer a single query from both Bayesian and Markov networks and time taken to infer the query is calculated.
- The same query is inferred multiple times and time is compared for both networks and count will generate each time
- From the multiple results, I can infer that time taken for Bayesian network is greater than Markov network for both variable and belief propagation methods
- The average time taken for Bayesian network to infer the query is 0.046 sec and for Markov network is 0.062 sec.
- The query inferred for belief propagation is  
variables=['x1','x3','x4','x5'], evidence= {'x2': 2, 'x6': 0}
- The model returned the following probabilities  
The query is: {'x1': 0, 'x3': 1, 'x4': 0, 'x5': 0}
- For the given evidence parameters  
 $x_2^2$ : shape of loop of h is curved left side and straight right side  
 $x_6^0$ : shape of t is tented.
- The model returned the best settings as :  
 $x_1^0$ : Height relationship of t to h where t is shorter than h  
 $x_3^1$ : Shape of Arch of h is pointed  
 $x_4^0$ : Height of cross is on upper half of staff  
 $x_5^0$ : Base line of h is slanting upwards

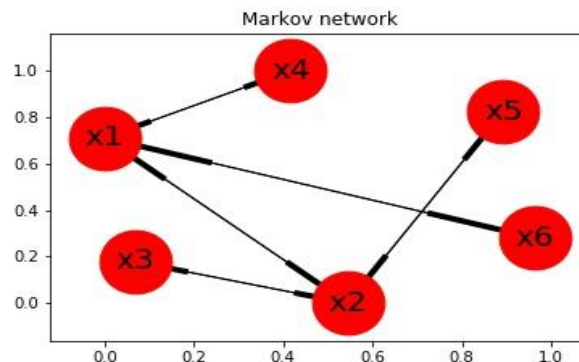


Figure3: Markov model of best probability 'th' mode

## 4 Task 4: Constructing Bayesian networks for MNIST dataset

### 4.1 Constructing the Bayesian network

- In this task, we need to construct the Bayesian network, by using the data given. One model is built by using the hill climb search method and the second model is built by using the constraint-based estimator which gives us a partial network and remaining nodes are constructed arbitrarily.

### 4.2 Evaluating the goodness Score of the models

- The goodness score of all models are calculated using the k2 score and the results are compared to find the high probability 'AND' model and low probability 'AND' model which are displayed below.

Table3: K2 Score for 'AND' models

K2 SCORE	VALUES
Model 1	-9462.70 [HIGH PROBABILITY AND MODEL]
Model 2	-9651.20 [LOW PROBABILITY AND MODEL]
Model 3	-9650.59
Model 4	-9638.59
Model 5	-9708.69

- Edges of high probability model: ('f3', 'f4'), ('f3', 'f9'), ('f3', 'f8'), ('f5', 'f9'), ('f5', 'f3'), ('f9', 'f8'), ('f9', 'f7'), ('f9', 'f1'), ('f9', 'f6'), ('f9', 'f2'), ('f9', 'f4')
- Edges of low probability model : ('f6', 'f4'), ('f6', 'f1'), ('f4', 'f3'), ('f4', 'f5'), ('f3', 'f7'), ('f5', 'f7'), ('f5', 'f8'), ('f7', 'f9'), ('f8', 'f9')

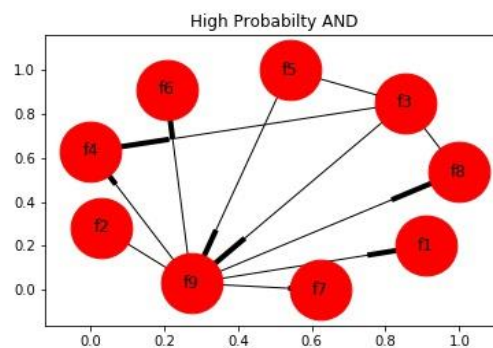


Figure 4: High probability AND model

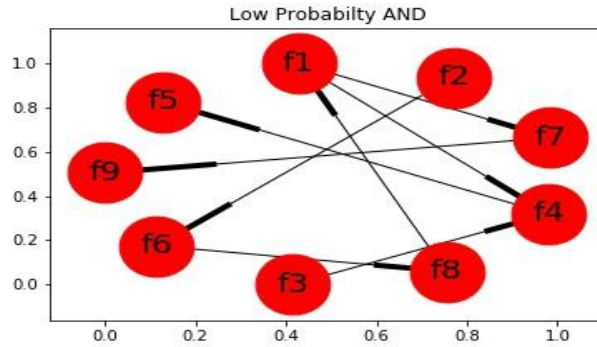


Figure5: Low probability AND mode

### 4.3 Inferring CPD from the model

In this process, Bayesian estimator has been used to infer a CPD's from Bayesian networks.

So, when we infer the query, the Bayesian network already has all the precomputed values and it will be easy for the model to retrieve the query CPD tables.

f5	f5 (0)	f5 (1)	f5 (2)	f5 (3)
f3 (0)	0.20967741935483872	0.024390243902439025	0.11823899371069183	0.125
f3 (1)	0.7419354838709677	0.9512195121951219	0.8528301886792453	0.375
f3 (2)	0.04838709677419355	0.024390243902439025	0.028930817610062894	0.5

### References

- [1] [www.pgmpy.org/estimators.html](http://www.pgmpy.org/estimators.html)
- [2] <https://ermongroup.github.io/cs228-notes/inference/sampling/>
- [3] [https://frnsys.com/ai\\_notes/foundations/probabilistic\\_graphical\\_models.html](https://frnsys.com/ai_notes/foundations/probabilistic_graphical_models.html)
- [4] [www.cedar.buffalo.edu/~srihari/CSE674/Chap12/12.1-ForwardSampling.pdf](http://www.cedar.buffalo.edu/~srihari/CSE674/Chap12/12.1-ForwardSampling.pdf)