Medical Diagnosis

Harbin Institute of Technology (Shenzhen)

Intelligent Computation Research Center

Date: 11/01/2017

目录

[1. Introduction 1](#_Toc497338556)

[2. Expectation Maximization 1](#_Toc497338557)

[3 Data Structure 2](#_Toc497338558)

[4 Analysis and Implement 3](#_Toc497338559)

[4.1 Extend Data 5](#_Toc497338560)

[4.2 Store Combinations 5](#_Toc497338561)

[4.3 Create *θ* Table 7](#_Toc497338562)

[4.4 Calculate Weights (Estep) 8](#_Toc497338563)

[4.5 Calculate Probability (Mstep) 9](#_Toc497338564)

[4.6 Calculate Maximum Likelihood 11](#_Toc497338565)

[5. Conclusion 11](#_Toc497338566)

[5.1 Algorithm Thinking 11](#_Toc497338567)

[5.2 Experiences from Project 12](#_Toc497338568)

[6. Task Assignment 12](#_Toc497338569)

# 

# 1. Introduction

Some medical researchers plan to observe the interlocking of certain diseases and symptoms by creating a Bayesian network model. As a graduate student in computer science, we need to learn and improve the probability parameters in the bayesian network model by the patient's symptoms’ records. However, in real life, the patients’ symptoms’ records is often incomplete. What we need to do is to try to calculate the probability of Bayesian networks if the records have a certain amount of missing value

The project will take place in the Bayesian network, which medical researchers have created. The structure of Bayesian network is shown in figure 1-1.

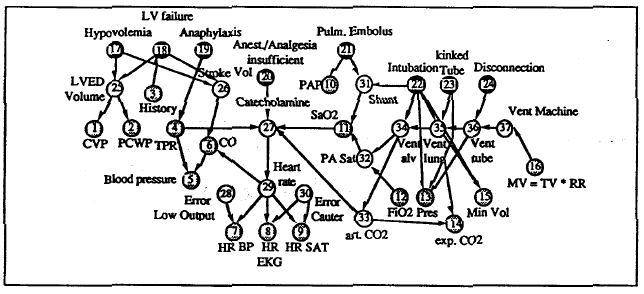


Figure 1‑1 The structure of Bayesian network

# 2. Expectation Maximization

In the project, we use the EM algorithm to calculate the probability of Bayesian networks.

The probabilistic models we discussed before only have the observable variables. Then we can use the method of maximum likelihood estimation or Bayesian estimation directly. But it is unfit for the models which have the latent variables. Precisely, the EM algorithm is just the algorithm for these models.

The process of EM algorithm is divided into two main parts. First, we should initialize the model parameters *θ* randomly. Then, we begin to iterate the algorithm which can be separated into two steps. The first step called E-step is to calculate the conditional probability of joint distribution. The formula is shown as follows.

 (2-1)

 (2-2)

In the formula,  is the joint distribution,  is the conditional distribution.

The second step called M-step is to maximize . The formula is shown as follows.

 (2-3)

After the two steps, if  is convergent, the algorithm will be ended. If not, the algorithm will go to E-step to continue iterating.

In the end, the algorithm will output model parameters *θ.*

# 3 Data Structure

To calculate the probability by EM algorithm, we mainly use three data structures to store different types of data.

The extended data is stored in re\_data. Because there are some incomplete data which labeled with “?” in alarm.bif , we must extend the original data into complete data. We use the ArrayList structure, which can be added when needed, to store the extended data. There is also a Arraylist in each element of the ArrayList re\_data. We set all possible values in the location of “?” , and add the line\_index in the first column, the weight of each sample data in the 39th column, the column of “?” is shown in the 40th column, the Total Probability of the every record is stored in the last column.

The thirty-seven columns data is stored after the first column. The data structure is shown as follows.

Table 3-1 The structure of re\_data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| line\_index | 1-37 data | weight | weight the column index of “?” | Total Probability |
| line\_index | 1-37 data | weight | weight the column index of “?” | Total Probability |
| …… | …… | …… | …… | …… |
| …… | …… | …… | …… | …… |
| …… | …… | …… | …… | …… |

The combination of parents and itself nodes is stored in combine. We use the ArrayList structure, which can be added when needed, to store the combination. Combine has thirty-seven pieces of data which are corresponding with thirty-seven variables. In each piece of data, there is a map. The key of each map is the combination of parents nodes and itself’s index. The value of each map is all of the combination of parents nodes and itself’s value. The data structure is shown as follows.

Table 3-2 The structure of combine

|  |
| --- |
| <map> [Arraylist (int)combine\_index]:[Arraylist (string)combine\_values] |
| <map> [Arraylist (int)combine\_index]:[Arraylist (string)combine\_values] |
| …… |
| …… |
| …… |

The probability of Bayesian network is stored in theta. We use the ArrayList structure to store the probability. Theta has thirty-seven pieces of data which are corresponding with thirty-seven variables. In each piece of data, there is a ArrayList which includes the probability of combinations corresponding with the order of combine\_values. The data structure is shown as follows.

Table 3-3 The structure of theta

|  |
| --- |
| [Arraylist (double) (double) (double) (double)] |
| [Arraylist (double) (double) (double) (double)] |
| …… |
| …… |
| …… |

# 4 Analysis and Implement

We implement the EM algorithm according to the pseudocode shown as follows.

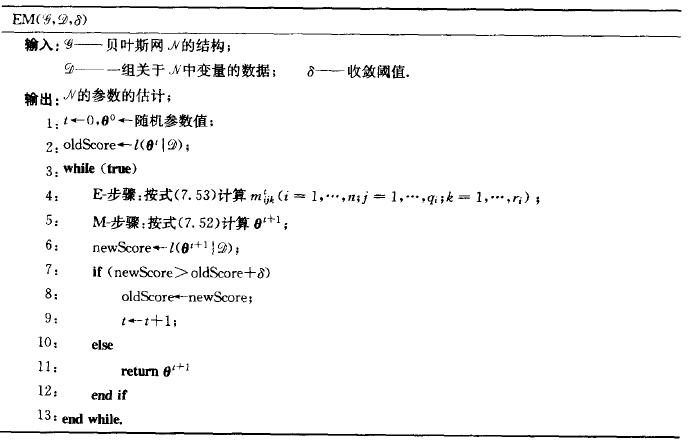


Figure 4‑1 The pseudocode of EM algorithm

The process of EM algorithm to calculate probability is shown as follows.

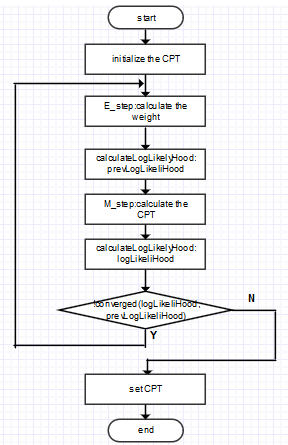


Figure 4‑2 The flow of EM algorithm

## 4.1 Extend Data

In each piece of data, there is no more than a missing value. We replace the missing value with all the possible values of this variable. So, the original data is extended into 31380 pieces of data. To get data more conveniently, we add the line\_index in the first column, add the weight of the row in the 39th, add the column of “?” in the 40th and add the total probability of the row in the last column.

The code is shown as follows.



Figure 4‑3 The code of extending data

## **4.2 Store Combinations**

In the alarm.bif, there are many tables that store the probabilities of different combinations. To get the combinations’ indexs and values intuitively, we use a map to store the corresponding indexs and values. We get the combinations’ indexs by traversing the parents nodes. We get different combinations of values which have the same order with tables in alarm.bif by iterating . The code is shown as follows.



Figure 4‑4 The code of storing combinations



Figure 4‑5 The code of storing combinations

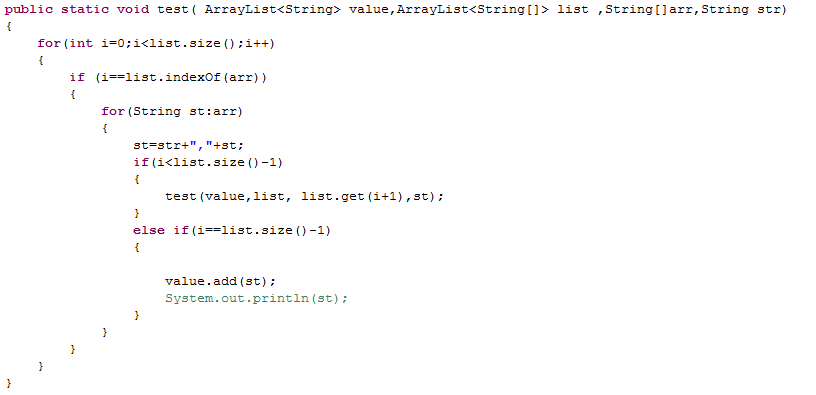


Figure 4‑6 The code of storing combinations

## 4.3 Create *θ* Table

We create*θ*table according to the order of combinations in combine’s map. First, we calculate the prior probability of none-parents nodes. Then, we set the average probability in having-parents nodes. The code is shown as follows.



Figure 4‑7 The code of creating *θ* table

## 4.4 Calculate Weights (Estep)

We calculate the weight of the row which has the “?” and set the weight of the row which does not have the “?” as 1.

First, we must calculate the total probability of every record. To calculate the total probability, we must calculate the conditional probability of every node. According to the initial CPT, we can get the initial probability of every node.

And we multiply the every node’s probability to get the total probability of every record.

Next, according to the first column of extended data, we can get the combinations which have the same index of original and calculate the every accumulation of them. For the weight of every record, we can use total probability of rows which had “?” divides the corresponding accumulation.

The code is shown as follows.



Figure 4‑8 The code of calculating weights

## 4.5 Calculate Probability (Mstep)

We calculate the probability according to weights we have gotten in the above work. The formula to calculate probability is shown as follows.

 (4-1)

First, we get the indexs and values of combinations. Then, we find the corresponding column and get weights of which have the same values with what we want. Similarly, we get weights of parents’ combinations. In the end, we can get *θ* values. The code is shown as follows.



Figure 4‑9 The code of calculating probability



Figure 4‑10 The code of calculating probability

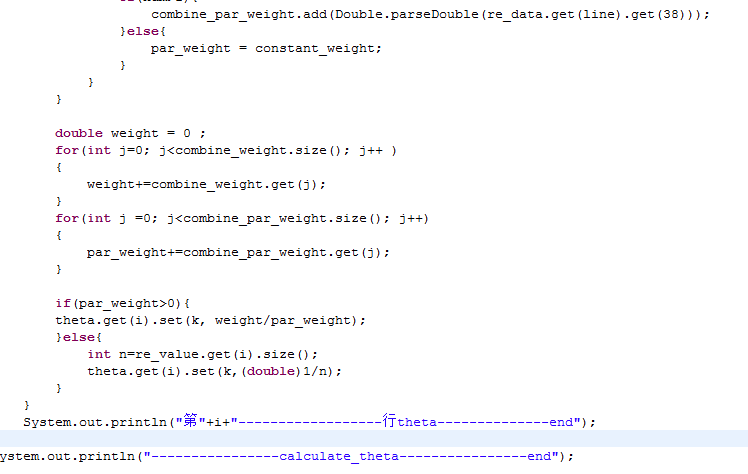


Figure 4‑11 The code of calculating probability

## 4.6 Calculate Maximum Likelihood

We calculate the maximum likelihood according to the weights and total probability we have gotten. We get log of product of the weight and total probability ,then get the sum of them. The code is shown as follows.

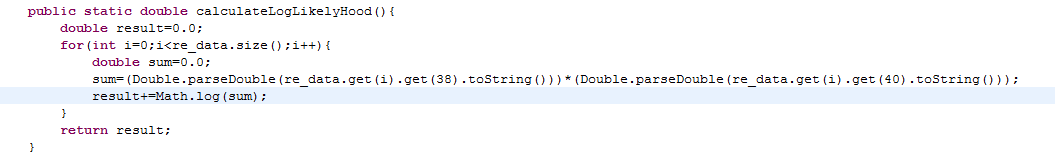


Figure 4‑12 The code of calculating maximum likelihood

# 5. Conclusion

## 5.1 Algorithm Thinking

Thinking about EM algorithm, we can see that the known data is observed data. The unknown data is incomplete data and model parameters. In E-step, we immobilize model parameters and optimize the distribution of incomplete data. In M-step, we immobilize the distribution of incomplete data and optimize model parameters.

## 5.2 Experiences from Project

In this experiment, we calculate the probability by the EM algorithm to finish the project. At first, we intended to delete the incomplete data to calculate the probability. But it is too unscientific. The EM algorithm can get the result iteratively to approximate value. So we choose EM algorithm to implement the project.

In the process of implement, we mainly use ArrayList and HashMap to store the data. And we have troubles in dealing with data because of the complicit data structures. But in the further work, we found that the structure is useful for getting data.

To be honest, We learned a lot from the project. Firstly, We have a deep comprehension of EM algorithms by implementing them. Then, we improve the programming level by several days’ coding. Thirdly, we can find solutions to solve bugs more efficiently.

# 6. Task Assignment

|  |  |
| --- | --- |
| Name | Task Assignment (code and report) |
| Leng Jia | Store combinations、 create θ table、 calculate probability. |
| Yan Hongyu | Extend data、 calculate weights、 calculate maximum likelihood |