**Report for Homework 3**

**1.Introduction**

## In this program, we deal with the problems using Parametric Learning and Bayesian Network. We install and do some practice with the Bayesian Network Toolbox for machine learning. And we work on maximum likelihood parameter estimation from complete data and missing values. For the problem in homework2, we apply parametric learning method with model selection to select out the best model to predict on family car. Also BNT toolbox is introduced to create a Bayesian network and build the hypothesis model about the family car problem. Finally we compare the empirical error and generalization capability for these two machine learning methods.

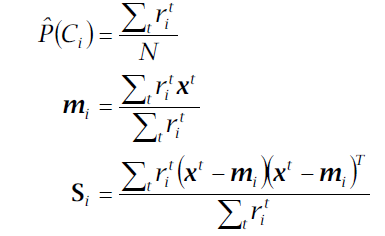
**2.Methods**

For the first part, we follow the tutorial of BNT online and work on the examples for maximum likelihood parameter estimation from complete data and with missing values about the sprinkler problem as shown in figure.1. We use the function learn\_params and learn\_params\_em respectively to training our data to get the conditional probabilities for node 4 WetGrass . Also we compare the conditional probabilities in the two scenarios.

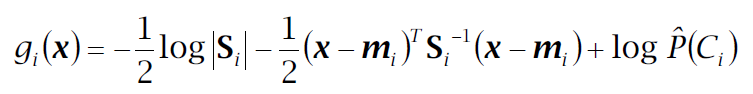
Figure.1 Sprinkler Problem

For the pregnancy problem, we use the BNT toolbox to create a Bayesian network based on the network structure. Then we learn the parameters based on the sample of 5 cases with missing values using learn\_params\_em to get the conditional probabilities required.

For the second part, we compare parametric learning and Bayesian network learning based on the data generated in homework2. For parametric learning, we first use the formula below to do parameters estimation for mean **m** and covariance matrix **S** based on the training data. After estimation of parameters, we start to do model selection from the most complex model, that is called Different, Hyper ellipsoidal. This means this model has different **Si** for class '0' and '1' and its shape looks like hyper ellipsoid. Then resorting to the discriminant function **gi(x)** below, we can determine the class for the data points. Therefore, in this way we can compute out the empirical errors for the training



data set and validation data set. In order to illustrate our model more clearly, we scan all the data points in the whole range from 0 to 5 for the special data points where the discriminant function g1(x) and g2(x) is equal. Then we use the Contour function in MATLAB to plot the boundary for the two classes. For the second model which has the shared **S**, We modify the discriminant function **gi(x)** with the new **S** determined by the prior probabilities for two classes. And with the same method above as the most complex model, We can obtain the decision boundary and empirical errors for this model with shared S. Also for the third model with diagonal S , we turn the shared S in the second model into a diagonal matrix and apply this new S into our discriminant function. Then the decision boundary and empirical errors can be obtained for this model. For the last model with diagonal **S** and equal variances, we just make S11 and S22 equal based on the prior probabilities and then get the decision boundary and the empirical errors for the training data and validation data. So , we compute out the boundaries and empirical errors for the four models in parametric learning.



After parametric learning ,we focus on the Bayesian network using BNT toolbox. Just as the example in the tutorial, we use the train data generated in Homework 2 as the sample for the parameters. We create a Bayesian network shown below with three nodes C,A and B. In this network C denotes the class and A,B represents X,Y for the data points. First we specify random parameters for node A ,B and C. Because A and B are continuous nodes, so we use gaussian\_CPD function to generate Gaussian distribution. And we use tabular\_CPD to generate random distribution for discrete node C. Then based on the training sample, we use learn\_params functions to learn the parameters for the Bayesian network. After learning the parameters , we can also scan the data points used as the evidence for node A and B in the whole range from 0 to 5 to obtain the special data points where the conditional probability of C for two classes are the same. Then we use the contour function to plot the decision boundary based on these special data points. Also for every data point in the validation set, we can take it as an evidence to get the conditional probability for C. Then we can determine which class these data points should be in to compute the empirical error for the validation data set. In the same way , the empirical error for the training data set can be easily obtained. Therefore, we get the decision boundary and empirical errors for the Bayesian network.

Figure.2 Bayesian Network for Family Car

**3. Results**

For the first part, in the sprinkler problem as shown below, we get the results based on complete data and missing values. We can find the learned parameters are pretty close to the true parameters for the conditional probabilities for node W. For example P(W=1 | S = 1,R =1) = 1 is exactly the same as the true parameter. And for the learned parameters with missing values, we get the conditional probabilities shown below. We find after 10 iterations , the conditional probabilities for node W are also close to the true parameters. For example, P(W=1 | S = 1,R =1) = 0.9692 , this is pretty close to the specified true value 1.

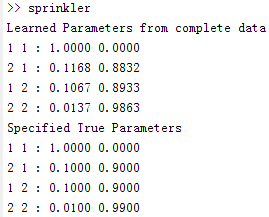


Fig 3. Learned Parameters from complete data and True Parameters

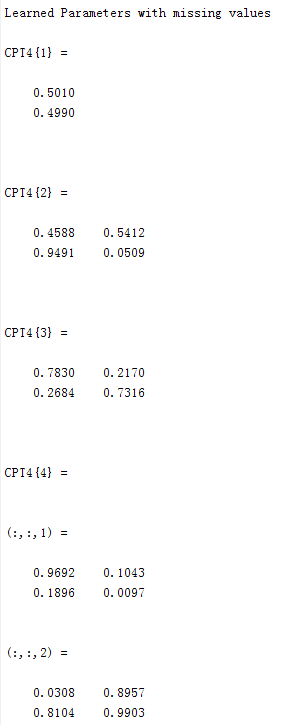


Figure 4. Learned Parameters with missing values

For the pregnancy problem, because of the missing values, we use learn\_params\_em to learn the parameters based on the 5 case. We compute out the probabilities for nodes Pr, Bt and Ut shown as follows. For example, P(Bt = Pos | Pr = Yes) = 0.6666 , P(Ut = Pos |Pr = Yes) = 0.9999. In our program , '1' represents 'Pos' or 'Yes' , '2' represents 'Neg' or 'No'.

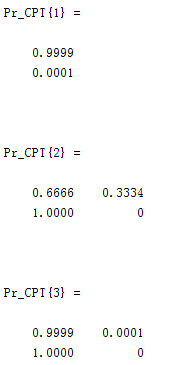


Figure 5. Learned Parameters for Pregnancy problem

For the 2rd part , we compute the decision boundaries and empirical errors for the 4 models for parametric learning shown below. We also use BNT toolbox to get the decision boundary and empirical error for Bayesian network below. We increase the size of the training data set from 50 to 400. And we set the number of the validation data set as 200. So we can compare the generalization capability for these model and Bayesian network. In the graphs below, the green 'o' and blue '+' are the training data points, and the black 'o' and '+' are our validation data points. The 5 colorful curves in the graph such as Figure 6 are the decision boundaries for different learning model and Bayesian network. Sometimes, there are maybe fewer than 5 boundaries in some graph such as Figure 8. This is because the boundaries are beyond the range of X-Y axis from 0 to 5, so they can not show up in the scope of our coordinate plane. We also obtain the empirical errors for the training set and validation set. And we find the most complex model and Bayesian network can achieve lower empirical errors compared with the other models. Therefore , for parametric learning, we should choose the most complex model ,that is the hyper ellipsoidal model with different covariance matrix **Si** .

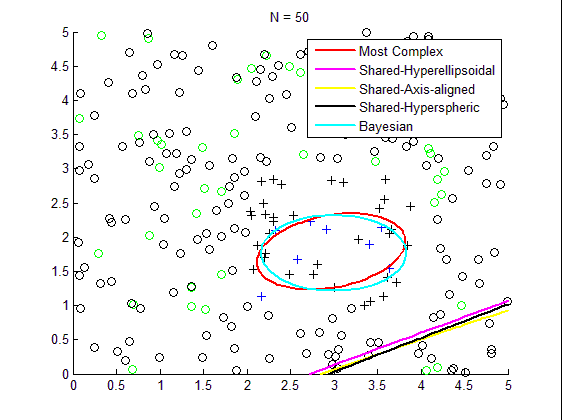


Figure 6. Decision Boundary for N = 50

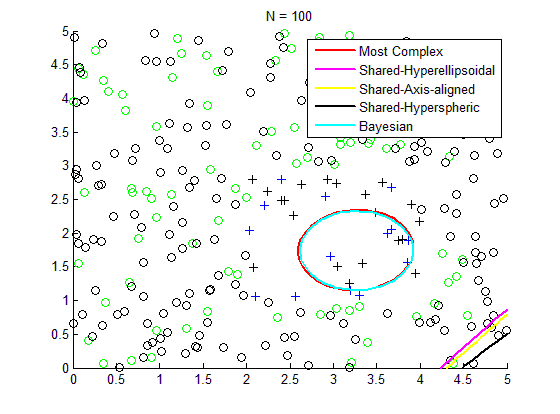


Figure 7. Decision Boundary for N = 100

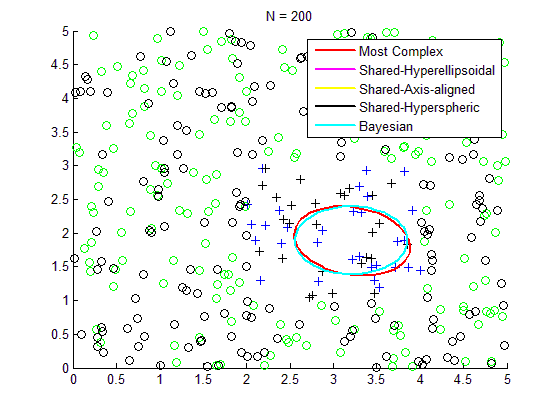


Figure 8. Decision Boundary for N = 200

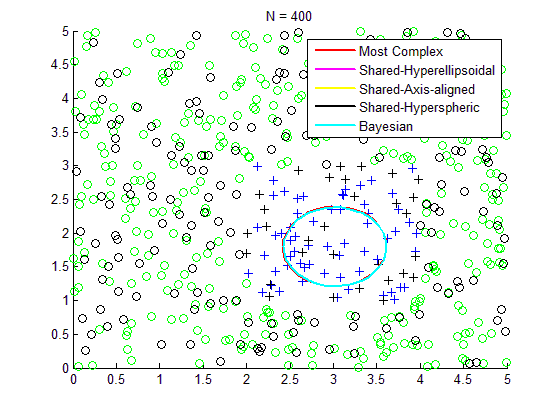


Figure 9. Decision Boundary for N = 400

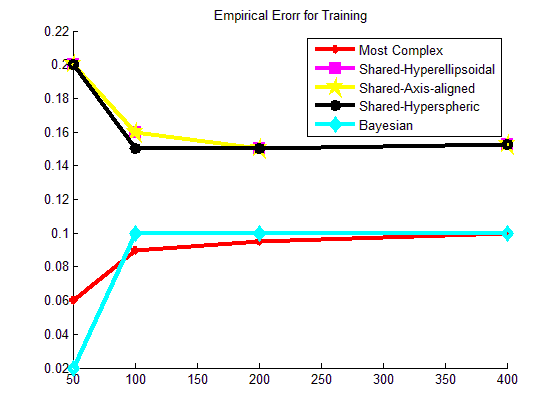


Figure 10. Empirical Errors for Training

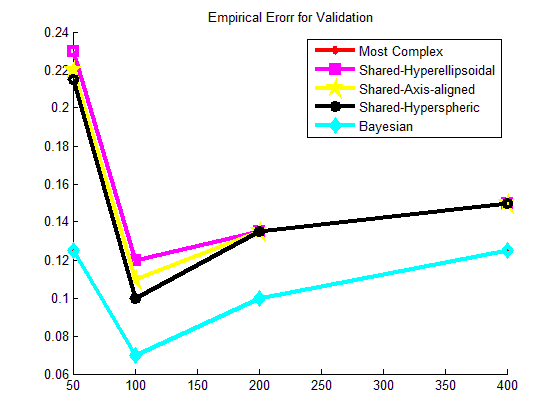


Figure 11. Empirical Errors for Validation

**4. Discussion**

Based on the results above, we can do some discussion on Parametric Learning and Bayesian Network. For the sprinkler problem in part 1, we compute the conditional probabilities for the nodes. Comparing the results from complete data and incomplete data, we find the results for complete data are more accurate and close to the specified true parameters than incomplete data. For example, with complete data, P(W=1 | S = 1,R =1) = 1 , but with incomplete data, P(W=1 | S = 1,R =1) = 0.9692. Also check other conditional probabilities of node W, we find the probabilities with complete data are more similar to the true parameters than that with incomplete data. Thus learned parameters with complete data are more accurate than that with incomplete data. For pregnancy problem, we compute P(Pr), P(Bt|Pr),P(Ut|Pr).We can see that the learned parameter based on incomplete data are not accurate.

In the 2rd part, for parametric learning, we can find the decision boundary for Most Complex model is an ellipsoid ,and for the other three models the boundary may be lines shown in Figure 6. And based on the errors in Figure 10 and 11 ,we can see that Most Complex model can get lower empirical errors than the other three models. Thus for model selection, Most Complex model is a better choice for parametric learning. For Most Complex model, we can see that the number of data points do not impact the accuracy for parametric classifier explicitly shown in Figure 11. And the error rate stays about 0.1. Also with the increase of the data points, the decision section for '+' seems to become smaller first, and then stays steady as an ellipsoid for Most Complex model shown in Figure 6,7,8 and 9.

For Bayesian Network, we use BNT tool to create the Bayesian and network and learn the parameters. Based on Figure 10 and 11, we can see the increase of data points seems not to impact the empirical error for Bayesian Network. And the decision boundary becomes smaller first, then stays steady just as that of Most Complex model shown in Figure 6,7,8 and 9.

Considering the results for Most Complex model and Bayesian Network, we find that with the increase of data points, the two decision boundaries trends to be more and more similar, then finally in Figure 9, the decision boundaries become identical when N =400. Also in Figure 11, we find the empirical error rates for the two classifiers become identical. The reason why these two classifiers obtain the same results is that both in Most Complex and Bayesian Network, we use Gaussian distribution for the training data points to learn the parameters. Because in Most Complex model, the covariance matrix **Si** is not compromised, so the discriminant function gi(x) can function as accurately as the conditional probability P(C|A,B) in Bayesian network. Thus these two classifiers can obtain almost the same results shown in Figure 9 and 11.

**5. Software listing and executable software**

This program uses MATLAB including three files, sprinkler.m, Preg.m and Part2.m.

For sprinkler problem , please run sprinkler.m .For pregnancy problem, please run , Preg.m. And for Part 2, please run Part2.m to get the results and graphs.

The m files are included in the file FullBNT-1.0.7\bnt .