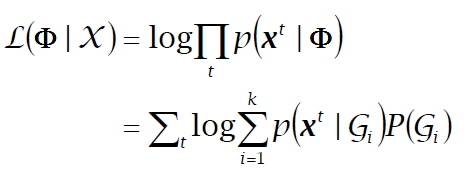
**Report for Homework 5**

**1.Introduction**

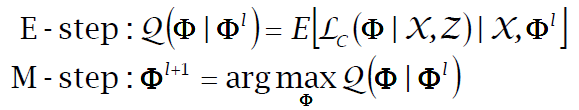
## In this homework, we deal with the problems about Gaussian mixtures using [Expectation Maximization.](http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction.html) First, we study the Expectation Maximization tutorial and use the function to implement Expectation Maximization algorithm on our family class problem. Then we write the training function gtrain.m based on the tutorial. Also we use the function grec.m to evaluate the likelihood for the data point. Moreover, we use the data from HW2 to apply this algorithm to the class of "family car" and "not family car". And we design the function gclass.m to create a binary classifier for the two classes. Finally we study the classification accuracy with different data points in the training set and different number of Gaussians in the mixture. We visualize the decision boundary for two classes and the generalization error based on the number of data points and clusters for Expectation Maximization.

**2.Methods**

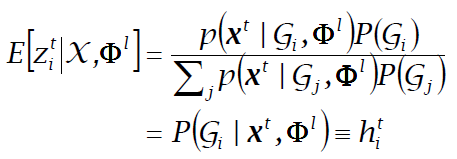
For Problem 1 and 2, we study the tutorial for Expectation Maximization and work on the function gtrain.m to implement EM clustering with Gaussian mixtures. EM is the abbreviation for Expectation Maximization. This algorithm applies the maximized likelihood method to the data set **X** to find the parameters **Φ** in theGaussian mixture model.



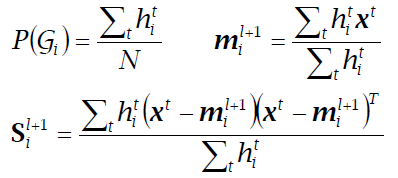
In order to make the optimization simpler ,we assume the hidden variables **Z** which introduces the complete likelihood in terms of **X** and **Z**.So in EM algorithm, there two iteration steps including E-step and M-step. In E-step, we estimate **Z** based on **X** and current parameters**Φ**. And in M-step, we find the new **Φ'** given Z and the old **Φ.**



We also know that an increase in **Ω** also increases the incomplete likelihood. So in order to maximize the likelihood ,we can do iterations until the parameter **Φ** converges. For E-step , the hidden variable if **Xt** belong to **Gi** , otherwise the variable equals 0.Assume the conditional variable follows the normal distribution **N(µi,Σi)**, then we can use the following formula in E-step for the hidden variable. We find that the expected value of hidden variable is the posterior probability based on the cluster Gi.



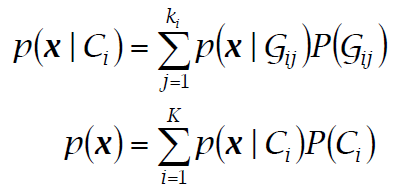
Therefore in M-step, we compute the parameters in each iteration based on the following formula.



Thus, based on EM algorithm ,we can obtain the Gaussian mixture after two iteration steps.

In Problem 3, we design the function grec.m to compute the likelihood for the data point **X** based on the Gaussian mixture from Problem 2. In this problem, because we assume the data point within each cluster follows the normal distribution, so we use the probability density function of normal distribution to compute the likelihood P(X|Gi) for each cluster given the parameters we get from EM algorithm.

As for Problem 4, we use the data from HW2 to fit the Gaussian mixture to the two class "family car" and "not family car". After EM algorithm in gtrain.m and likelihood in grec.m ,we design the function gclass.m to act as the binary classifier to assign the data point to one of the classes. In this function, based on the results from gtrain.m and grec.m ,we use the formula below to compute the posterior probability **P(Ci|X)** for the two classes. Then we compare the probabilities for the classes , the greater one indicate the assigned class for the data point. Thus the function gclass.m can act as a classifier to deal with the family car problem.



In Problem 5, we study the classification accuracy based on different number of data point and clusters. We use gclass.m function to deal with the family car problem in different scenarios. We compute the generalization error just like the previous homework. Finally we visualize the generalization error and decision boundary for different number of data points and clusters within each class.

**3. Results**

For the first two problems, we study the tutorial for EM and design our own function gtrain.m. We find that this function can output the parameters in Gaussian mixture including prior probability **P(Gi)** ,expectation **µi** and covariance matrix **Σi** for eachclusterbased on input cluster number and data points for each class in the training data set.

For the third problem, using prec.m ,we can obtain the output likelihood for the clusters based on the validation data and Gaussian mixture for each class in our model.

For Problem 4, using the function gclass.m ,we can assign every data point in the validation set to one of the class for family car problem based on the validation set and the Gaussian mixture in the clustering.

For Problem 5, we compute the generalization error in our model. In order to study the classification accuracy , we run our functions under different conditions including the number of training data points and clusters within each class. First we draw out the graph about the distribution of **N** data points in the training set such as Fig.1. The green "o" represents the class '0', and the blue "+" represents the class '1'. Also we present the graph about the validation data and the decision boundary from our classifier gclass.m for every data number **N** in the training set and cluster number **C** within each class as shown in Fig.2.However sometimes, because of the singularity of the covariance matrix **Σ**, we cannot see the decision boundary in the graph shown in Fig.3. The worst case is that in some extreme combination, such as **N** = 50 and C = **16** , there may be an error in our model, thus there is no graph for this combination ,we only get four graphs when **N** =50.

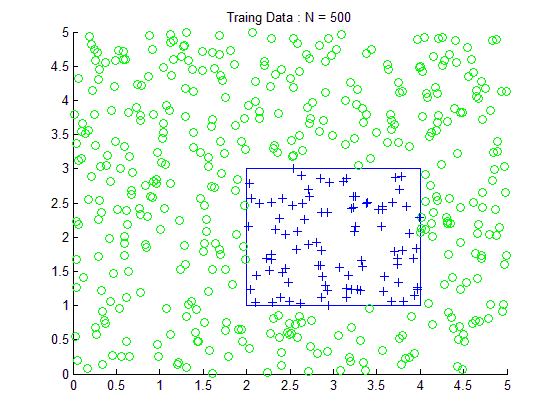


Fig.1 Distribution of data points in training set

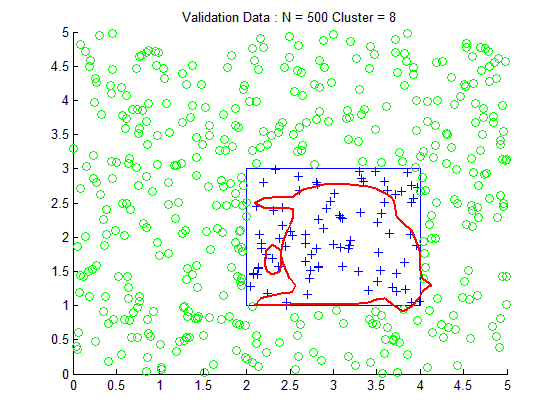


Fig.2 Validation data and decision boundary

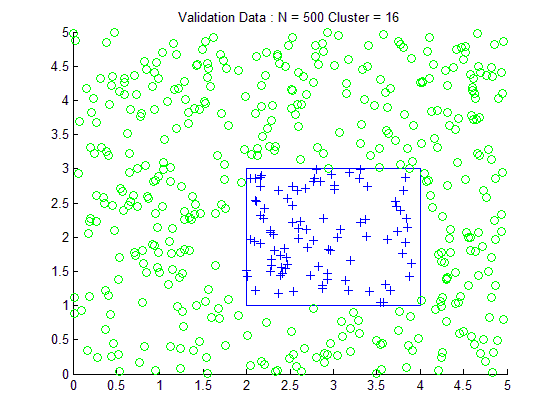


Fig.3 Validation data without decision boundary

Also we visualize the generalization error for each combination of **N** and **C** shown in Fig.4 below. For the extreme combination such as **N** = 50 and C = **16** , because of the error, we may not get the error in this case, so in the graph, the number of points may be different for every curve shown in Fig.4. We can see that for Cluster = 16 , there are only three stars in the cyan curve. There is no point for **N** = 50 .

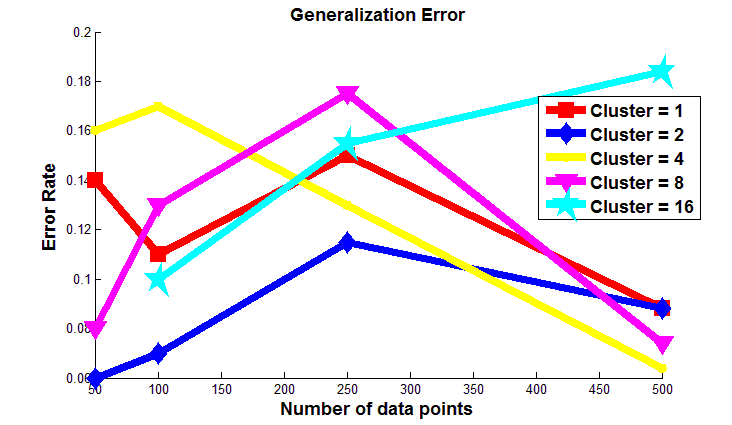


Fig.4 Generalization Error

**4. Discussion**

Based on the results in the section above, we can make a discussion on the classification accuracy for our model . We find that in some situation such as **N** = 50 and C = **16** , there may be error in our algorithm, because the number of generated training data point from one class may be smaller than the cluster number. Thus we have to use try and catch command to skip this loop in MATLAB. Therefore because of this error, we cannot get the generalization error in this case ,which leads to the absence of some point in the curve in the graph for generalization error Fig.4. Also sometimes, we may not see the decision boundary for the validation data set. This is because covariance matrix is singular, and the validation data points are all classified as a single label '0' or '1' in our model. Therefore the decision boundary for the two classes will not show up in our graph for validation data . Based on Fig.4, we can see that there is no obvious relationship between the generalization error and the conditions including data number **N** and cluster number **C**. We can see that the generalization error is normally below 0.2 for any data number and cluster number. In most cases, with **N** increasing, for a specific **C** except for **C** = 16, the error will first increase and then decrease to a level below 0.1. However for **C** = 16, the error keeps increasing with the increase of **N**. So we can see that **N** = 500 seems to be a good choice for training data because the errors are pretty low for most cluster numbers. Also we can see that when C = 2 , the generalization error keeps at low level below 0.12 for any number **N** in the training set.

**5. Software listing and executable software**

This program uses MATLAB including 4 files, gtrain.m, grec.m, gclass.m and P5.m.

Please click and run P5.m to get the graphs and results.