1 Mixture of Gaussians and Bayes Classification

A) Assume that you have two classes of datapoints, each of which has been fitted with a single Gauss function. The two classes have the same number of points and the Gaussians are not normalized. One can determine to which of the two classes each group of datapoints belongs by comparing the likelihoods of the data under each Gauss function. The Gauss function corresponding to the highest likelihood wins. Draw the classification boundary for each of the three examples illustrated in fig. 1.

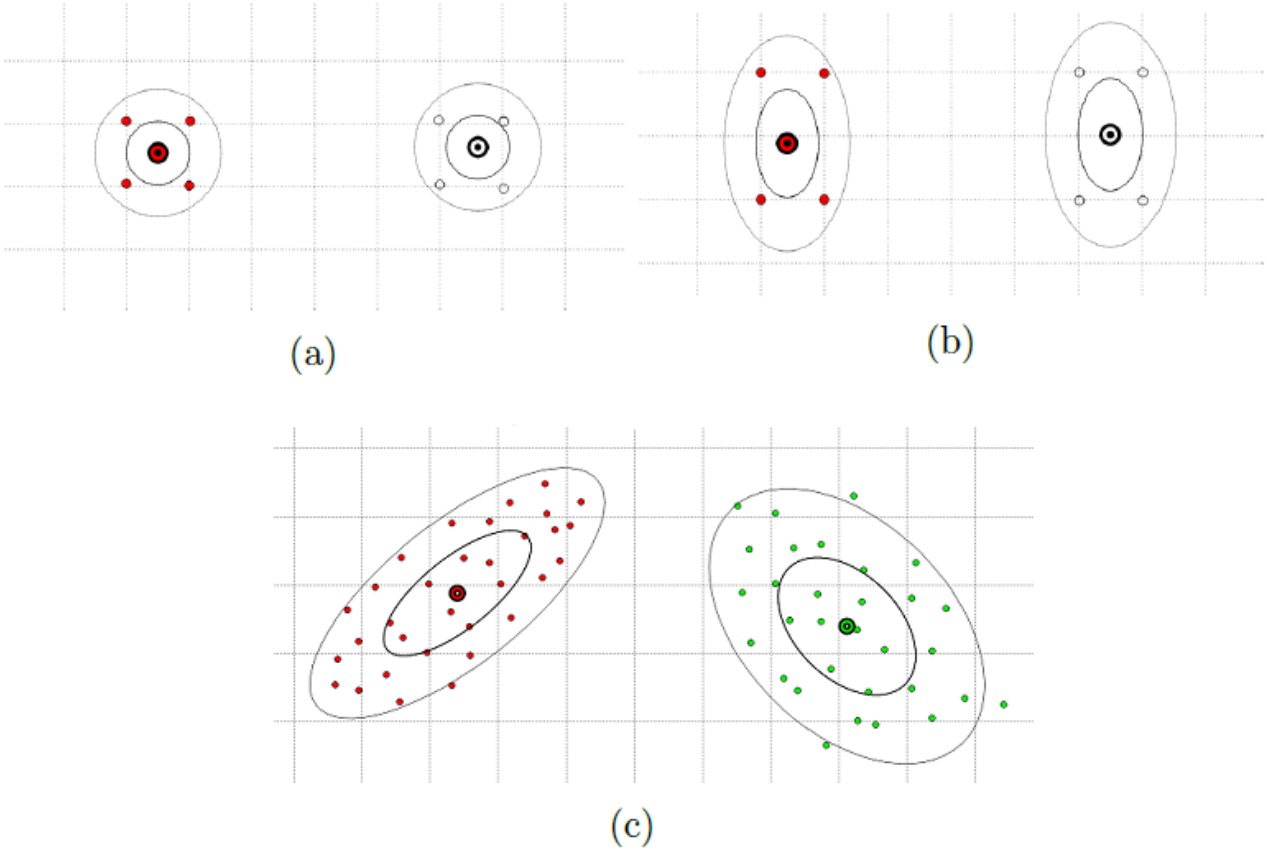


Figure 1: Three datasets comprised of two classes of datapoints. Each class is fitted with a single Gauss function. The isolines correspond to one and two standard deviations. In this question, we assume that the isolines have equal values for all Gauss functions.

Note: In MLDemos and in this question, the Gaussians are not normalized; meaning that the isolines corresponding to one and two standard deviations give the same probabilities for each class. Since the number of points is similar for both classes, the boundary can be obtained through the intersection of isolines of the same probabilities.

In a Bayesian framework, Gaussians are usually normalized, i.e., the value of a Gaussian at point x is computed as

$$\frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right],$$

where the normalization factor $\frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}}$ ensures that the function integrates to 1, defining a probability density function. Consequently, the values corresponding to each isoline may differ across Gaussian functions. In this question, we assume that the isolines have identical values for all Gaussian functions.

B) Revisit the (a) and (b) classification problems above where each class is modeled with a single Gauss function aligned with the X-Y axes. Assume that the two covariance matrices for the two classes are the same. Show how the classification boundary moves as a function of the number of datapoints in each class. Consider a ratio of 1:1, 2:1, and 4:1, respectively, for the number of datapoints in class 1 versus the number of datapoints in class 2. Assume that the datapoints in each case have been generated following a Gauss distribution in each class.

Hint: Recall that the classification boundary is obtained by setting the likelihood ratio of the two classes to 1, i.e.,

$$\frac{p(y=1|x)}{p(y=2|x)} = 1.$$

Using Bayes rule, this is equivalent to

$$\frac{p(x|y=1)}{p(x|y=2)} \times \frac{p(y=1)}{p(y=2)} = 1,$$

which can be transformed using the logarithm into

$$-\frac{1}{2}(x-\mu^1)^\top \Sigma^{-1}(x-\mu^1) + \ln p(y=1) = -\frac{1}{2}(x-\mu^2)^\top \Sigma^{-1}(x-\mu^2) + \ln p(y=2).$$

Note: In MLDemos, the class ratio is not taken into consideration and you will not be able to observe the behavior in this question.

C) Now, we consider the case where the Gaussians are normalized and their covariance matrix is different. What would be the boundary for the example shown in fig. 2?

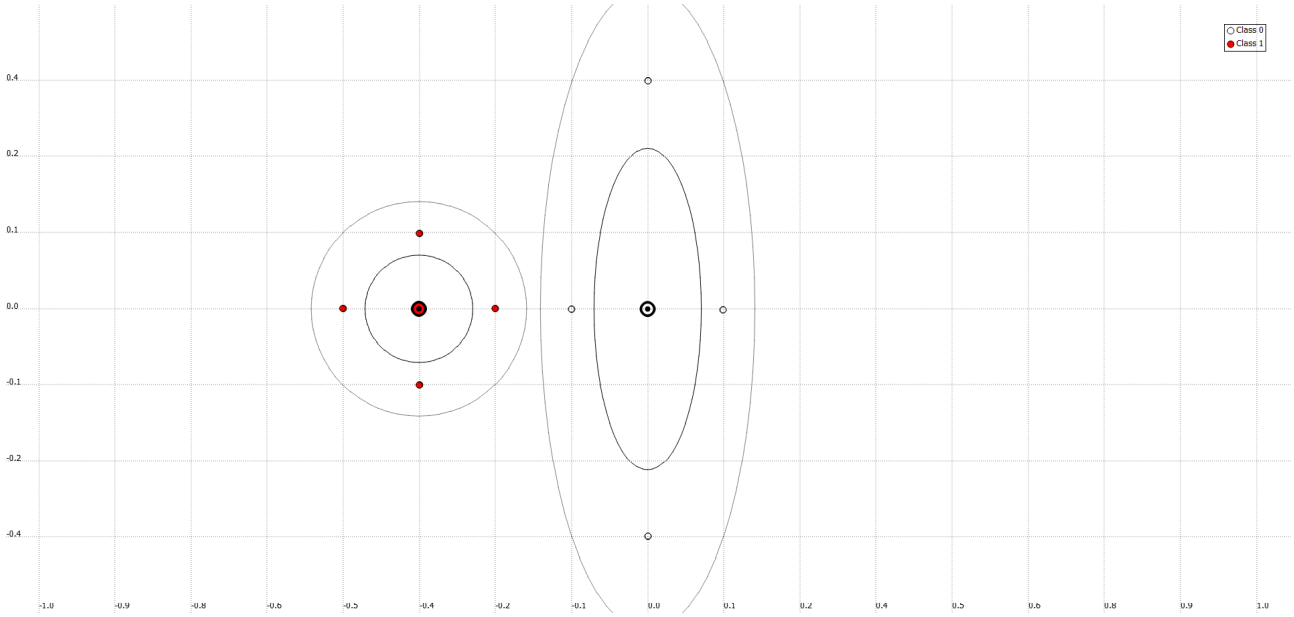
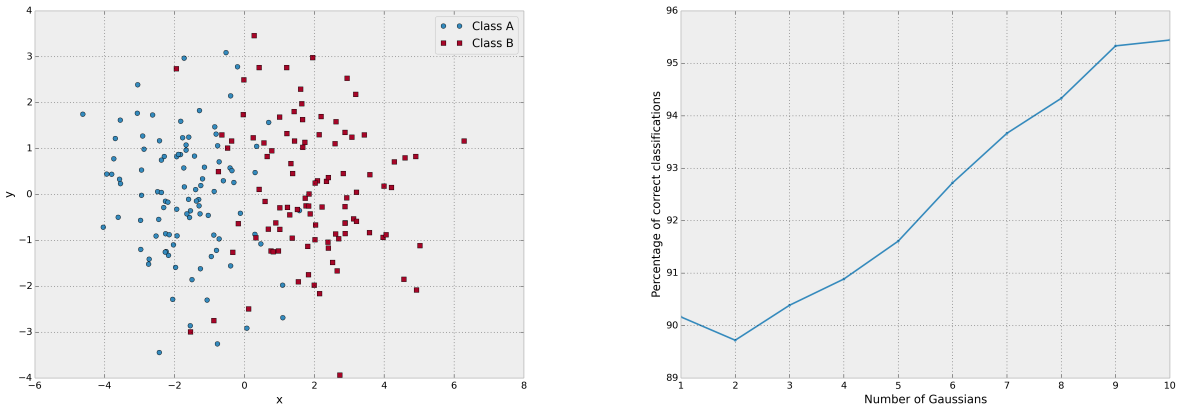


Figure 2: Binary classification task. Each class is fitted with a single Gauss function. The isolines correspond to one and two standard deviations. In this question, we assume the isolines do not have equal values for all Gauss functions. The variance along the X-axis is the same for both Gaussian distributions.

2 Overfitting, Generalization, and Computational Costs

A) Joe was given the labeled dataset represented on fig. 3a. With the goal of classifying the future unlabeled data, Joe trains separately a GMM for each class. He does not know the number of Gaussians to use in his GMM; thus, he tries different values and tests the performance of the classifier on the training dataset, obtaining the plot presented in fig. 3b. He decides to use mixtures of 10 Gaussians. Do you agree with Joe's decision? What would you do in his place?



(a) (b)

Figure 3: (a) Labeled dataset. (b) Percentage of correct classifications for GMM with different number of Gaussians.

B) Draw one example dataset and choice of model for GMM that would lead to overfitting.

C) What is the number of parameters to be estimated for GMM? What is the computational cost per iteration of the update step for GMM? Discuss the effect of choosing spherical, diagonal, or full covariance matrices.

D) On another dataset, Joe decides to use the KNN classification method. In order to find the best hyperparameter K, he decides to perform cross-validation and plot the mean train and test F-measure for different K (see fig. 4). What value of K should he select for his classification problem?

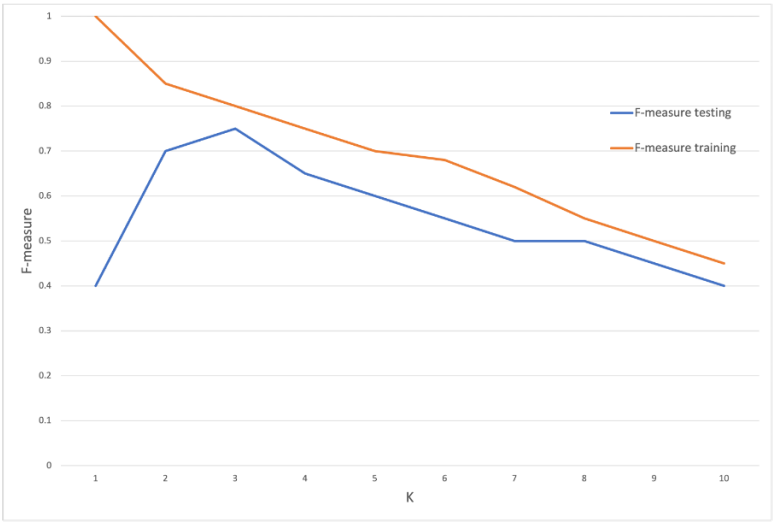


Figure 4: Mean train and test F-measure for different K.