Project 1:
Supervised Learning Using Bayesian Decision Rule - Two Category Classification

September 8, 2021

Yangsong Gu
Course COSC 522

---

1 Objective

The objective of this project is, first of all, to learn how to implement supervised learning algorithms based on Bayesian decision theory. The second objective is to get you familiar with the design flow when applying machine learning algorithms to solve real-world problems. Some practical considerations include, for example, 1) the selection of the right pdf model to characterize the data distribution in the training set, 2) the selection of the right ratio of prior probability, 3) the different ways to evaluate the performance of the learning algorithm, and 4) how differently the same ML algorithm performs when applied to different datasets.

2 Data Sets

Two datasets will be used from Ripley's Pattern Recognition and Neural Networks, both are 2-category classification problems. The first is a synthetic dataset with 2 features where there are about equal number of samples in each category. The second is a dataset for diabetes in Pima Indians with 7 features where the number of diabetic patients is much less than that of the normal patients.

- The synthetic dataset: synth.tr (the training set) and synth.te (the test set)

  Preprocessing you need to do: remove the first row with any text editor. No need to write Python code for that. The labels are 0 and 1.

- The Pima dataset: pima.tr (the training set) and pima.te (the test set)

  Preprocessing you need to do: 1) Remove the first row with any text editor; 2) Change the labels from "Yes" and "No" to "0" and "1" respectively indicating 'with disease' and 'without disease' - you can do this using the same text editor; and 3) Normalize the data set to make the features comparable (or with the same scale). Suppose x is a data sample, m_i is the mean of each feature i, $sigma_i$ is the

standard deviation of each feature i, then normalization is conducted by $\frac{(x-m_i)}{sigma_i}$. Keep in mind that you also need to normalize the samples in the test set. Be careful which mean and standard deviation you should use. (For each sample in the test set, use the same m_i and $sigma_i$ you derived from the training set.)

3 Algorithm

You need to implement the three cases of the discriminant function based on Baysian decision theory, 1) minimum Euclidean distance classifier (linear machine), 2) minimum Mahalanobis distance classifier (linear machine), and 3) the generic form of Baysian decision rule (quadratic machine), where Gaussian pdf is assumed.

3.1 Discriminant Function

Assume the data points follow the Gaussian distribution, the minimum Euclidean distance classifier (linear machine) can be written as:

$$g_i(x) = -\frac{\|x-\mu_i\|^2}{2\sigma^2} + \ln P(w_i) \tag{1}$$

For minimum Mahalanobis distance classifier (linear machine), the discriminant function can be written as:

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^T \sum\nolimits^{-1} (x-\mu_i) + \ln P(w_i) \tag{2}$$

For the generic form of Bayesian decision rule (quadratic machine), the discriminant function is:

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^T \sum\nolimits_i^{-1} (x-\mu_i) - \ln |\sum\nolimits_i| + \ln P(w_i) \tag{3}$$

where subscript i denotes the class, x is the feature vector (1-d), μ_i is the mean value of features of category i. $\sum_i$ is the covariance matrices (d-d), w_i denotes the class.

4 Performance Metrics

Three metrics are used to evaluate the performance of the ML algorithms, including 1) overall classification accuracy (Equation 4), 2) classwise classification accuracy (Equation 5), and 3) run time. For two category classification, we have:

$$overall\ acc. = \frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

$$
\begin{aligned}
class\ P\ acc. &= \frac{TP}{TP+FP} \\
class\ N\ acc. &= \frac{TN}{TN+FN}
\end{aligned}
\tag{5}
$$

where TP the cases when the actual class of the data point was 1(True) and the predicted is also 1(True); TN are the cases when the actual class of the data point was 0(False) and the predicted is also 0(False); FP are the cases when the actual class of the data point was 0(False) and the predicted is 1(True). FN are the cases when the actual class of the data point was 1(True) and the predicted is 0(False).

5 Tasks:

5.1 Visualization of training sample

Figure 1 presents the distribution of training sample in which the purple dots represents the class 0 while yellow dots represents the class 1. From visual inspection, there are two apparent "components" or "clusters" of each class, with different mean value. Thus, I think single-modal Gaussian is not a reasonable model for the probability density function. Maybe the mixture Gaussian model is a preferable model.

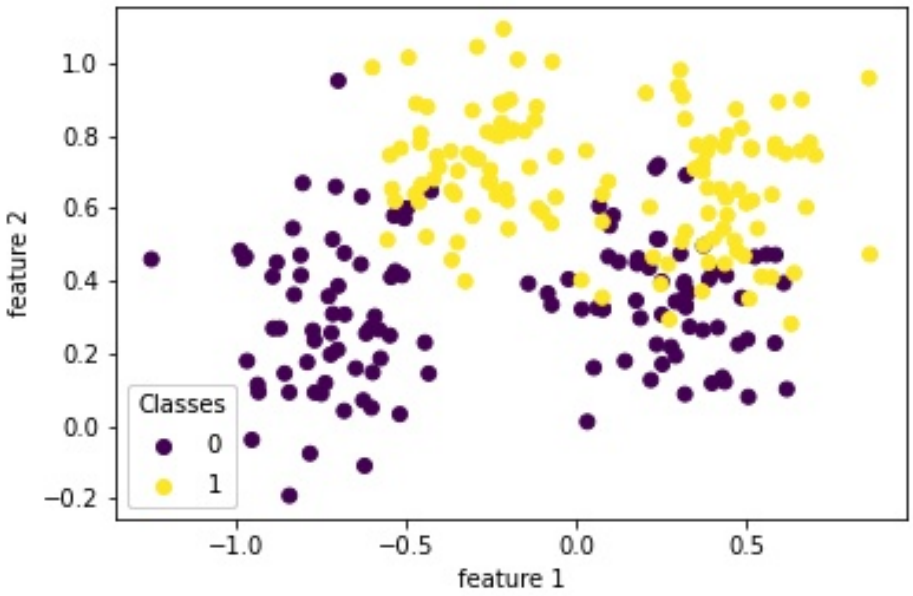


Figure 1: Two classes of training set from synthetic data set

5.2 Performance matrices

Given equal priority of class 0 and 1, **Table** 1 shows the results of three different classifiers performed on synthetic test set. **Table** 2 presents the results of classifiers applied on Pima test set.

Table 1: Model performance (synthetic data set & Gaussian model)

	Overall acc.	Acc. of class 0	Acc. of class 1	Run time (seconds)
Euclidean dist.	71.30%	72.81%	69.98%	0.0170
Mahalanobis dist.	89.20%	88.58%	89.84%	0.0818
Quadratic	89.80%	89.02%	90.61%	0.1054

Table 2: Model performance (Pima data set & Gaussian model)

	Overall acc.	Acc. of class 0	Acc. of class 1	Run time (seconds)
Euclidean dist.	74.10%	58.78%	84.08%	0.0060
Mahalanobis dist.	76.81%	62.70%	85.44%	0.0479
Quadratic	74.10%	60.36%	81.00%	0.0677

5.3 Comprehensive Discussion

(accuracy) As shown in table 1, the minimum Euclidean distance classifier generates the lowest accuracy for all of accuracy performance and generic Gaussian classifier (quadratic machine) outperforms rest models in accuracy. In addition, Minimum Mahalanobis distance classifier and generic Gaussian classifier significantly overcome the minimum Euclidean distance classifier, with over 15% accuracy enhancement. From the table 2, we can see that minimum Mahalanobis distance obtained the best accuracy . It is also worth noting that all classifiers generate much better accuracy of class 1 than class 0. Looking at the synthetic dataset, the accuracy of class 0 and 1 is close. It might be because of the assumption of equal prior probability. Specifically, the equal prior probability is suitable for synthetic dataset with equal number of of samples in each category. while it is not reasonable for pima dataset where class 0 (yes) cases is much less than class 1 cases (no).

(run time) Run time is not only associated with the algorithm itself but also the size of test set(i.e., number of samples and features). Comparing the run time of each individual dataset, we found that the complexity of algorithm is in sequence of minimum Euclidean distance, minimum Mahalanobis distance and generic Gaussian classifier. Besides, compared to minimum Euclidean distance classifier, the generic Gaussian classifier takes exponential time to run on the same data set. From the dataset perspective, we know that synthetic dataset contains 250 test samples and 2 features while pima dataset contains 200 test samples with 7 features. Nonetheless, the results show that the run

time of each classifier on pima dataset is less than on synthetic dataset, indicating that the number of test samples potentially caused more run time than the features brought.

(effect assumption of covariance matrices) Recap that minimum Euclidean distance classifier assume the equal variance of each feature and no correlation between features. This classifier generates awful accuracy in synthetic dataset when there are only 2 features. In contrast,minimum mahalanobis distance classifier and generic Gaussian classifier generate the similar and relative good results. This indicates that the covariance and dependence between features should be considered in this dataset. In addition, the assumption of covariance matrices did make the difference to the results among different dataset. we noticed that the generic Gaussian form which distinguishes the covaraince of each category would **not** always generate the best results. For instance, when applied on Pima dataset, the generic Gaussian classifier is not as well performed as simple minimum Euclidean distance classifier regarding class 1 accuracy. In that case, the assumption of equal variance and Independence between features could cancel out the bias in training dataset, and it coincidentally brings benefits to final performance. Furthermore, the covariance matrices is important in that it records the correlation of features, especially as the number of features increased. The highly correlated features could not gain the benefit of performance due to potential over fitting issues, on the contrary, it will boost the computation time.

5.4 Decision boundaries

To figure out the decision boundary, we let the discriminant function of two classes equal, which is $g_1(x) = g_2(x)$. Then, by solving the equations (detailed derivation is given in Appendix 5.7), we can get the corresponding classifier boundaries on synthetic dataset, which are:

Minimum Euclidean distance classifier.

$$x_2 = -0.6209x_1 + 0.4459 \tag{6}$$

Minimum Mahalanobis distance classifier.

$$x_2 = -0.1033x_1 + 0.4844 \tag{7}$$

Quadratic classifier.

$$1.4677x_1^2 + 4.5656x_1x_2 + -3.7909x_1 + 3.5610x_2^2 - 16.3241.x_2 + 6.9487 = 0 \tag{8}$$

where x_1 and x_2 is feature 1 and feature 2, respectively.

Figure 2 demonstrates the three different decision boundaries upon test data points. Regarding two linear machine, the minimum Euclidean distance classifier (green dash line) differs from the minimum Mahalanobis distance (blue dash line) in the slope and intercept. The slope of minimum Mahalanobis distance classifier is smaller which

correctly classified more category 1. The non-linear machine (red line) has the similar decision boundary on the points dense area with the minimum mahalanobis distance classifier, as a long stretch of them is overlapped or very close to each other. Combining table 1, we can claim that the generic form gets more samples classified correctly at the boundary of data clusters than mahalanobis distance classifier, leading to a higher accuracy.

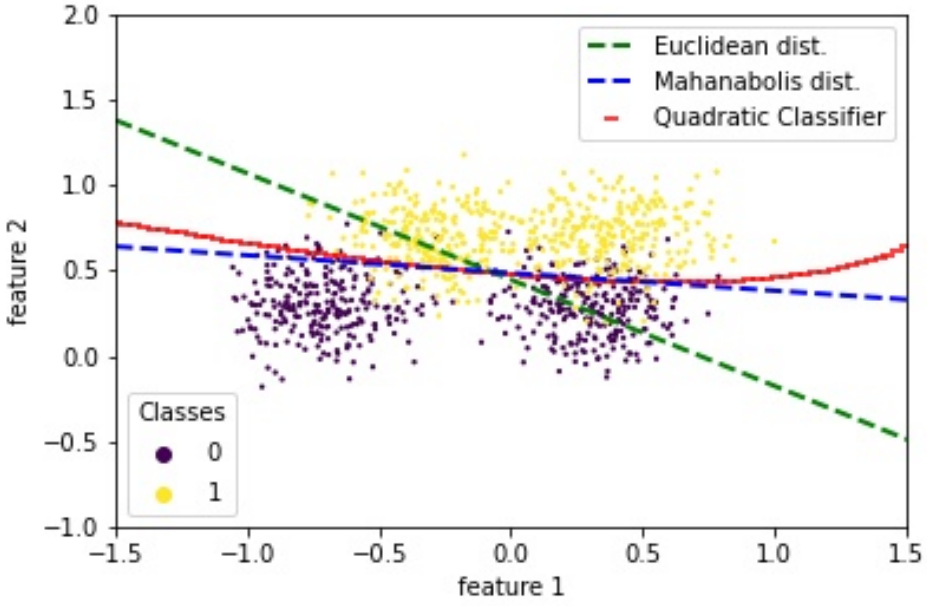


Figure 2: Two classes of test set from synthetic data set

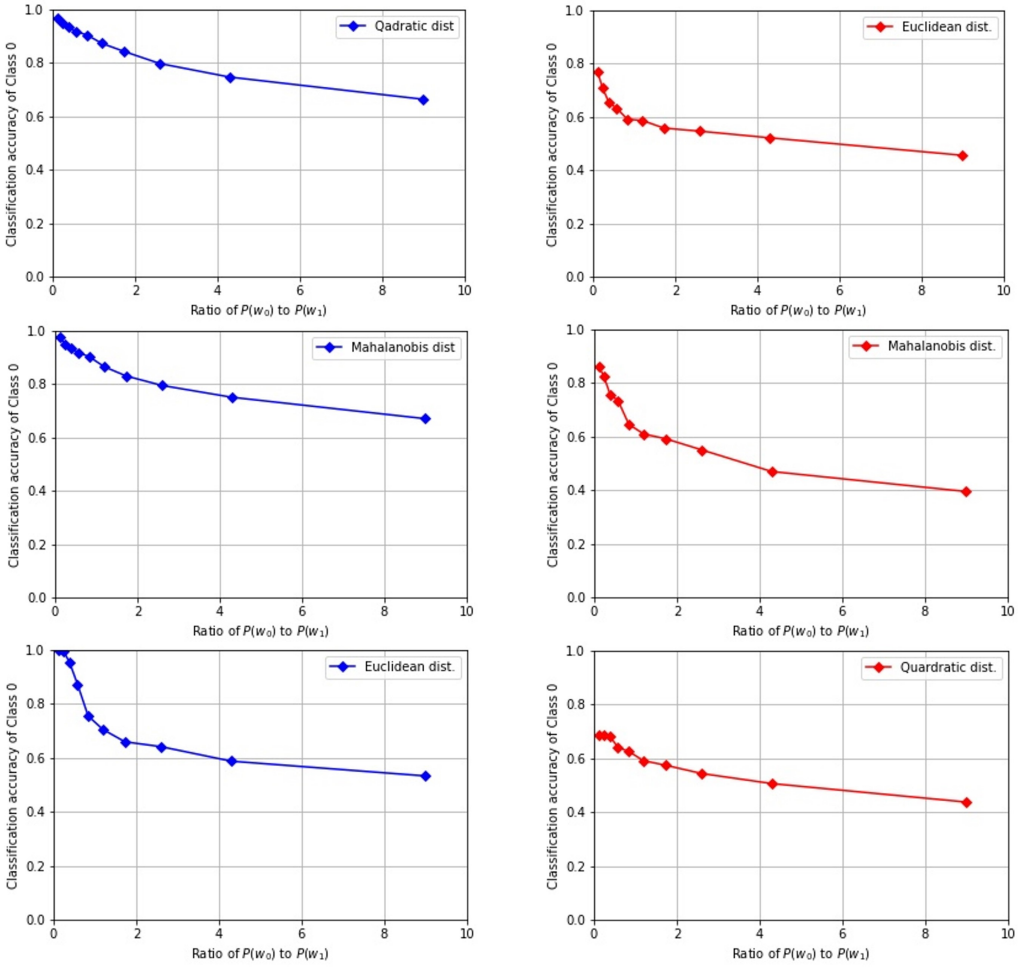
5.5 Class 0 accuracy v.s. ratio of prior probability

Figure 3 presents the class-0 accuracy v.s. ratio of prior probability, with left column being synthetic data and right column being pima data. For each subplot, the x-axis is the ratio of prior probability of class 0 $P(w_0)$ to the prior probability of class 1 $P(w_1)$ and the y-axis is the accuracy of class 0. The physical meaning of ratio denotes the scale of $P(w_1)$ relative to $P(w_1)$. The larger the ratio, the larger frequency of category 0. For instance, if the ratio is 2, the class 0 cases is twice of class 1. To compute the ratio, $P(w_0)$ is sampled every 0.1 between 0.1 and 0.9. All of these subplots show a negative exponential shape but with different slope.

Regarding synthetic dataset, the class-0 accuracy decreases gradually as the ratio increased on two linear machine while we can see a sharp dip on non-linear machine as the ratio is within 0 and 2. Besides, the range of accuracy change of non-linear machine

(i.e., (0.55 - 1))is larger than two linear machine (0.65-1), indicating that the performance of non-linear machine is more sensitive to prior probability than linear machine.

Regarding pima dataset, the minimum Euclidean distance classifier and generic gaussian classifier are more insert than the minimum Mahalanobis distance classifier as the ratio increased. The minimum Mahalanobis distance classifier could generate the higher class-0 accuracy than rest two models when the ratio is less than 2.



(a) synthetic data (b) pima data

Figure 3: class-0 accuracy vs. ratio of prior probability

5.6 Discussion

Through applying three different classifiers, that is, minimum Euclidean distance, minimum Mahalanobis distance and generic classifiers, on two different datasets, this report evaluates the performance of algorithms and discussed the potential reasons causing the

accuracy difference. For the same dataset, the root difference of three algorithms lies in the covariance metrices. The assumption of equal covariance and non-correlation between features would lead to awful classification especially when only few features involved. In practice, when the limited features are used for classification, we should distinguish the covariance and correlation between features. When different covariance is applied (i.e., generic form), the classifier could classify more outliers of each category correctly than linear machines. Comparing the different datasets, the performance of algorithm differs because of the test sample size including the number of samples and features. It is worth noting that the generic Gaussian classifier would not always produce the best results because of the potential sample bias and multicollinearity between features in training set. When their covaricance matrices are distinguished, the over fitting issues might arise, and the accuracy might be affected. Thus, in practice, the training sample should be finely sampled and pre-processed before training. What's more, the run time results indicate that the test sample size could require much more computational time than the extra time brought from the complexity of features.

5.7 Bonus task

The distribution of points in training set (Figure 1) suggests there are two clusters. Hence, a Gaussian Mixture Model with two components is considered. To find out the mean and standard deviation of two clusters, training set is manually separated by a vertical line x = -0.3. Denote the points left to this boundary is component 1, points right to this boundary is component 2. Referring to [1], the p.d.f of a two components Gaussian Mixture Model (GMM2) on 2-D dimension is:

$$\begin{aligned} p(x|\mu_1, \sum\nolimits_1, \mu_2, \sum\nolimits_2) = \lambda_1 \cdot \frac{1}{\sqrt{(2\pi)^2 \det(\sum_1)}} exp(-\frac{1}{2}(x-\mu_1)^T \sum\nolimits_1^{-1}(x-\mu_1)) + \\ \lambda_2 \cdot \frac{1}{\sqrt{(2\pi)^2 \det(\sum_2)}} exp(-\frac{1}{2}(x-\mu_2)^T \sum\nolimits_2^{-1}(x-\mu_2)) \end{aligned} \tag{9}$$

where $\lambda_1 + \lambda_2 = 1$ and μ_1 and μ_2 denote the mean vector of component 1 and 2 in training set of which the sub-item is the mean of each feature. $\sum_1$ and $\sum_2$ represents the covaraince matrices of component 1 and 2, respectively.

It is worth noting that the equation 9 shows the GMM2 is just the summation of weighted Gaussian Model. Hence, we can rewrite three types of discriminant functions by equations (10,11, 12).

Minimum Euclidean distance classifier:

$$g_i(x) = \ln\left(A \cdot \exp(-\frac{\|x-\mu_{i1}\|^2}{2\sigma_{i1}^2} + B \cdot \exp(\frac{\|x-\mu_{i2}\|^2}{2\sigma_{i2}^2})\right) + \ln P(w_i)$$
$$A = \lambda_1 * \frac{1}{\sqrt{(2\pi)^2 \sum_{i1}}} \tag{10}$$
$$B = \lambda_2 * \frac{1}{\sqrt{(2\pi)^2 \sum_{i2}}}$$

In this case, covariance matrices is same across categories, and the off-diagonal of covariance matrices is 0. $\sum_i$ can be simplified as $\sigma_i \cdot I$. In the end, we take the equal weight for GMM components, that is, $\lambda_1 = \lambda_2 = 0.5$.

Minimum Mahalanobis distance classifier:

$$g_i(x) = \ln\left(A \cdot \exp(-\frac{1}{2}(x-\mu_{i1})^T \sum\nolimits_{i1}^{-1}(x-\mu_{i1})) + \right.$$
$$\left. B \cdot \exp(-\frac{1}{2}(x-\mu_{i2})^T \sum\nolimits_{i2}^{-1}(x-\mu_{i2}))\right) + \ln P(w_i) \tag{11}$$

We set $\sum_{01} = \sum_{11}$,that is, the covariance matrices of class 0 in component 1 is equal to covariance matrices of class 1 in component 1. Likewise, we set $\sum_{02} = \sum_{12}$, meaning that the covariance matrices of class 0 in component 2 is equal to covariance matrices of class 1 in component 2.

Generic Gaussian classifier:

$$g_i(x) = \ln\left(A \cdot \exp(-\frac{1}{2}(x-\mu_{i1})^T \sum\nolimits_{i1}^{-1}(x-\mu_{i1})) + \right.$$
$$\left. B \cdot \exp(-\frac{1}{2}(x-\mu_{i2})^T \sum\nolimits_{i2}^{-1}(x-\mu_{i2}))\right) + \ln P(w_i) \tag{12}$$

Equation 12 is same as equation 11. The only difference is that the generic Gaussian classifier applies raw covaraiance matrices for each component and each category.

Table 3 summarizes the performance of three discriminant functions performed on synthetic data set where GMM2 p.d.f function is assumed. Compared with single-modal Gaussian model, all types of accuracy of three discriminant functions have improved. The minimum Euclidean distance classifier produced largest improvement in three discriminant functions, as Figure 4 shows. At the same time, the run time also increased a lot compared to the single-modal model. For class-wise accuracy, the Minimum Mahalanobis distance classifier outperforms other two classifiers regarding overall accuracy and class 0 accuracy. The generic Gaussian classifier obtained highest accuracy in terms of classifying category 1.

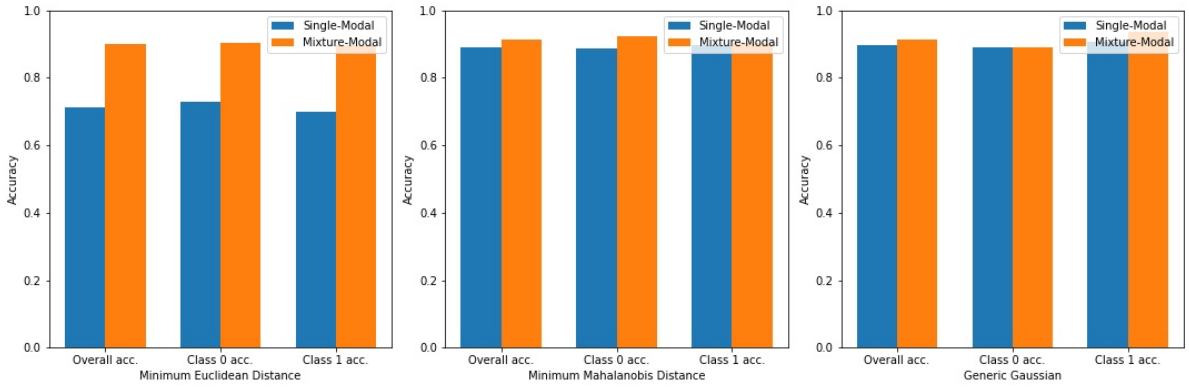


Figure 4: Performance Comparison

Table 3: Model performance (synthetic data set & p.d.f is GMM2)

	Overall acc.	Acc. of class 0	Acc. of class 1	Run time (seconds)
Euclidean dist.	90.10%	90.34%	89.86%	3.1130
Mahalanobis dist.	91.40%	92.42%	90.43%	3.4272
Quardratic	91.30%	89.18%	93.66%	3.3444

A Appendix:Classifier boundary derivation

References

[1] Nicholas Ruozzi. *Mixture Models EM*. URL: https://personal.utdallas.edu/~nrr150130/cs7301/2016fa/lects/Lecture_17_GMM.pdf. (accessed: 09.05.2021).

Figure 5: Minimum Euclidean distance classifier

two category

decision boundary $g_1(\vec{x}) = g_2(\vec{x})$ $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ features.

$\mu_1 = \begin{bmatrix} \mu_{11} \\ \mu_{12} \end{bmatrix}$ $\mu_2 = \begin{bmatrix} \mu_{21} \\ \mu_{22} \end{bmatrix}$

$$\frac{\vec{\mu}_1^T}{\sigma^2}\vec{x} - \frac{\vec{\mu}_1^T\vec{\mu}_1}{2\sigma^2} + \ln P(w_1)$$

$$= \frac{\mu_{11}\cdot x_1 + \mu_{12}\cdot x_2}{\sigma^2} - \frac{\mu_{11}^2+\mu_{12}^2}{2\sigma^2} + \ln P(w_1) + \frac{x_1^2+x_2^2}{2\sigma^2}$$

$$= \frac{x_1^2+x_2^2+2(\mu_{11}x_1+\mu_{12}x_2)-(\mu_{11}^2+\mu_{12}^2)}{2\sigma^2} + \ln P(w_1)$$

$$g(w_2) \quad \frac{x_1^2+x_2^2+2(\mu_{21}x_1+\mu_{22}x_2)-(\mu_{21}^2+\mu_{22}^2)}{2\sigma^2} + \ln P(w_2)$$

$P(w_1) = P(w_2)$ $\quad 2(\mu_{11}x_1+\mu_{12}x_2) - (\mu_{11}^2+\mu_{12}^2) = 2(\mu_{21}x_1+\mu_{22}x_2) - (\mu_{21}^2+\mu_{22}^2)$

$$\Rightarrow 2(\mu_{11}-\mu_{21})x_1 + 2(\mu_{12}-\mu_{22})x_2 - \left[(\mu_{11}^2+\mu_{12}^2)-(\mu_{21}^2+\mu_{22}^2)\right] \Rightarrow$$

plug in $\mu_{11} = -0.2009$

$\mu_{12} = 0.2885$

$\mu_{21} = 0.0519$

$\mu_{22} = 0.6958$

$$\Rightarrow -0.5058x_1 - 0.8146x_2 + 0.3632 \Rightarrow$$

$$\Rightarrow x_2 = -0.6207x_1 + 0.4459$$

Figure 6: Minimum Mahalanobis distance classifier

Minimum Mahalanobis distance

$g_i(\vec{x}) = -\frac{1}{2}(\vec{x}-\vec{\mu}_i)^T \Sigma_i^{-1}(\vec{x}-\vec{\mu}_i) + \ln P(w_i)$

$g_1(x) = \mu_1^T (\Sigma^{-1})^T \vec{x} - \frac{1}{2}\mu_1^T \Sigma^{-1}\vec{\mu}_1 + \ln(P(w_1))$

$\mu_1 = \begin{bmatrix}\mu_{11}\\ \mu_{12}\end{bmatrix}$ $\mu_2 = \begin{bmatrix}\mu_{21}\\ \mu_{22}\end{bmatrix}$

assume $\Sigma^{-1} = \begin{bmatrix}a & b\\ c & d\end{bmatrix}$

$= \begin{bmatrix}\mu_{11}\\ \mu_{12}\end{bmatrix}^T \cdot (\Sigma^{-1})^T \begin{bmatrix}x_1\\ x_2\end{bmatrix} - \frac{1}{2}\begin{bmatrix}\mu_{11}\\ \mu_{12}\end{bmatrix} \Sigma^{-1} \cdot [\mu_{11}, \mu_{12}] + \ln P(w_1)$

$= \begin{bmatrix}\mu_{11}\\ \mu_{12}\end{bmatrix}^T \begin{bmatrix}a & b\\ c & d\end{bmatrix}^T \begin{bmatrix}x_1\\ x_2\end{bmatrix} - \frac{1}{2}\begin{bmatrix}\mu_{11}\\ \mu_{12}\end{bmatrix} \begin{bmatrix}a & b\\ c & d\end{bmatrix} [\mu_{11}, \mu_{12}] + \ln P(w_1)$

$= [\mu_{11}(a+b), \mu_{12}(c+d)] \begin{bmatrix}x_1\\ x_2\end{bmatrix} - \frac{1}{2}[\mu_{11}(a+b), \mu_{12}(b+d)] \begin{bmatrix}\mu_{11}\\ \mu_{12}\end{bmatrix} + \ln P(w_1)$

$= \mu_{11}(a+b)x_1 + \mu_{12}(c+d)x_2 - \frac{1}{2}(\mu_{11}^2(a+b) + \mu_{12}^2(b+d)) + \ln P(w_1)$

likewise, we can get $g_2(x)$

$= \mu_{21}(a+b)x_1 + \mu_{22}(c+d)x_2 - \frac{1}{2}(\mu_{21}^2(a+b) + \mu_{22}^2(b+d)) + \ln P(w_2)$

set $g_1(x) = g_2(x)$

$\Rightarrow (a+b)(\mu_{11}-\mu_{21})x_1 + (c+d)(\mu_{12}-\mu_{22})x_2 - \frac{1}{2}\left((a+b)(\mu_{11}^2-\mu_{21}^2) + (b+d)(\mu_{12}^2-\mu_{22}^2)\right) = 0$

plug in $a, b, c, d, \mu_{11}, \mu_{12}, \mu_{21}, \mu_{22}$

$\Rightarrow$ $a = 4.584$ $\mu_{11} = -0.2059$
$b = 0.3015$ $\mu_{12} = 0.2885$
$c = 0.3015$ $\mu_{21} = 0.0519$
$d = 30.287$ $\mu_{22} = 0.6858$

$\Rightarrow 4.8855 \cdot (-0.2518)x_1 + 30.585 \times (-0.4073)x_2 - \frac{1}{2}[4.8855 \times (-0.0183) + 30.585(-0.4009)]$

$\Rightarrow -1.2298x_1 - 12.457x_2 + 6.2360 = 0$

$1.2351x_1 + 12.457x_2 - 6.2184 \Rightarrow$ $\Rightarrow x_2 = -0.1x_1 + 0.50$