Recap

- We have been considering the linear least squares method for learning classifiers and regression functions.

Recap

- We have been considering the linear least squares method for learning classifiers and regression functions.
- The least squares method is based on the criterion of minimizing mean squared error.

Recap

- We have been considering the linear least squares method for learning classifiers and regression functions.
- The least squares method is based on the criterion of minimizing mean squared error.
- This is a good way to fit linear models to given data.

Recap

- We have been considering the linear least squares method for learning classifiers and regression functions.
- The least squares method is based on the criterion of minimizing mean squared error.
- This is a good way to fit linear models to given data.
- As mentioned in the previous lecture, Fisher Linear Discriminant is another way of constructing linear classifiers.

Fisher Linear Discriminant

- A linear discriminant function based classifier is:

  Decide $X \in$ C-1 if $W^T X + w_0 > 0$

Fisher Linear Discriminant

- A linear discriminant function based classifier is:

  Decide $X \in$ C-1 if $W^T X + w_0 > 0$

- Hence One can think of the best W as the direction along which the two classes are well separated.

Fisher Linear Discriminant

- A linear discriminant function based classifier is:

  Decide $X \in$ C-1 if $W^T X + w_0 > 0$

- Hence One can think of the best W as the direction along which the two classes are well separated.
- We project the data along the direction W. Separation between points of different classes in the projected data is a good way to rate how good is W.

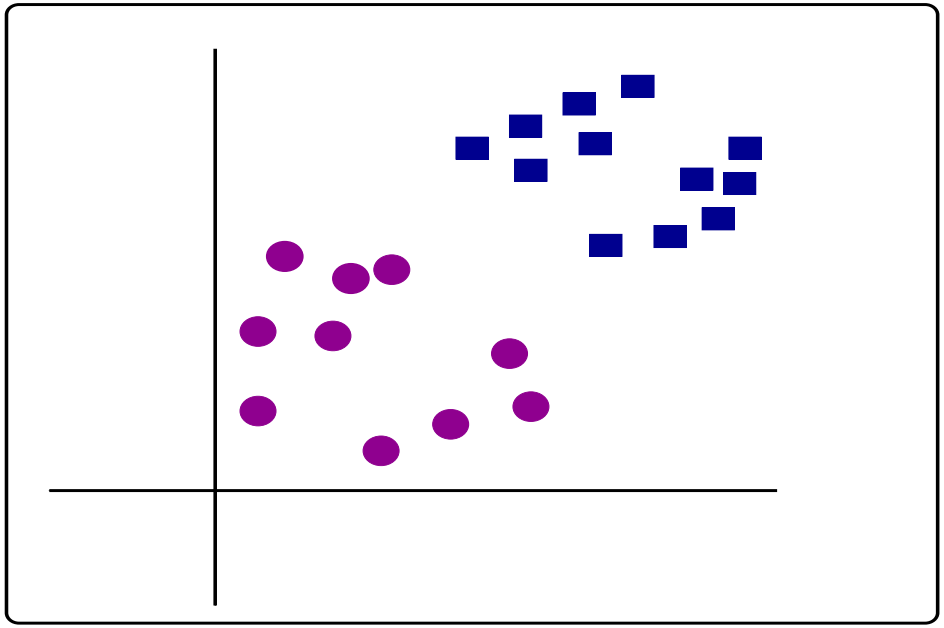
Fisher Linear Discriminant

- A linear discriminant function based classifier is:

  Decide $X \in$ C-1 if $W^T X + w_0 > 0$

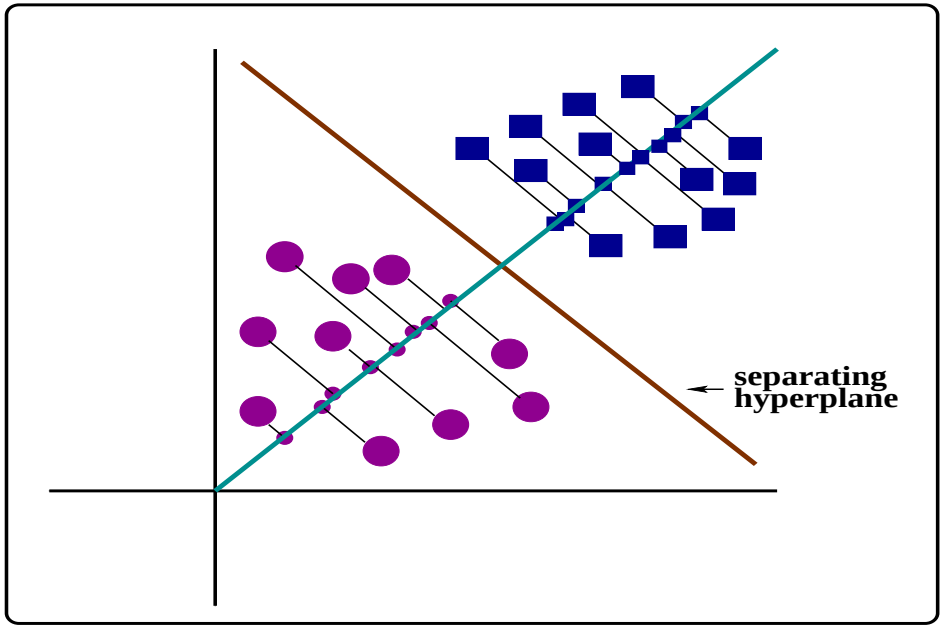
- Hence One can think of the best W as the direction along which the two classes are well separated.
- We project the data along the direction W. Separation between points of different classes in the projected data is a good way to rate how good is W.
- Such a method is called Fisher Linear Discriminant.

- Consider the following 2-class example

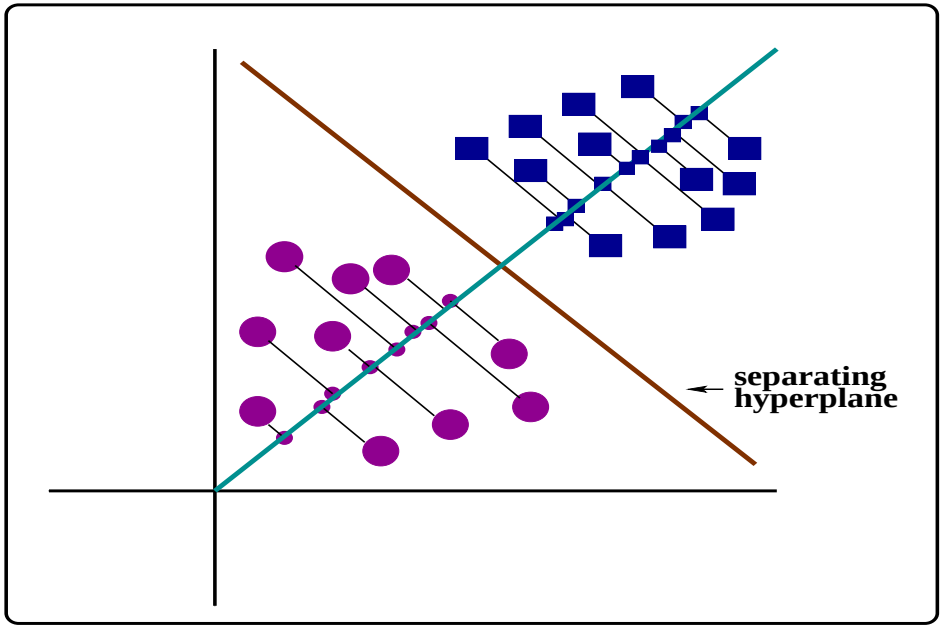


- A good direction to project the data here is as shown.

- A good direction to project the data here is as shown.



- A good direction to project the data here is as shown.



- Fisher Linear Discriminant is based on formalizing this notion

Fisher Linear Discriminant

- The idea is to find a direction W such that the training data of the two classes are well-separated if projected onto this direction.

Fisher Linear Discriminant

- The idea is to find a direction W such that the training data of the two classes are well-separated if projected onto this direction.
- We need some figure of merit for each W to characterize how well the W results in such a separation.

Fisher Linear Discriminant

- The idea is to find a direction W such that the training data of the two classes are well-separated if projected onto this direction.
- We need some figure of merit for each W to characterize how well the W results in such a separation.
- This is what we do now. We first consider the 2-class case.

- Let $\{(X_i,\ y_i),\ i = 1, \cdots, n\}$ be the data.

- Let $\{(X_i,\ y_i),\ i = 1, \cdots, n\}$ be the data.
- Let $y_i \in \{0,\ 1\}$.

- Let $\{(X_i,\ y_i),\ i = 1, \cdots, n\}$ be the data.
- Let $y_i \in \{0,\ 1\}$.
- Let C_0 and C_1 denote the two classes. Thus, if $y_i = 0$ then $X_i \in C_0$ and if $y_i = 1$ then $X_i \in C_1$.

- Let $\{(X_i,\ y_i),\ i = 1, \cdots, n\}$ be the data.
- Let $y_i \in \{0,\ 1\}$.
- Let C_0 and C_1 denote the two classes. Thus, if $y_i = 0$ then $X_i \in C_0$ and if $y_i = 1$ then $X_i \in C_1$.
- Let n_0 and n_1 denote the number of examples of each class. ($n = n_0 + n_1$)

- Let $\{(X_i,\ y_i),\ i = 1, \cdots, n\}$ be the data.
- Let $y_i \in \{0,\ 1\}$.
- Let C_0 and C_1 denote the two classes. Thus, if $y_i = 0$ then $X_i \in C_0$ and if $y_i = 1$ then $X_i \in C_1$.
- Let n_0 and n_1 denote the number of examples of each class. ($n = n_0 + n_1$)
- For any W, let $z_i = W^T X_i$.

- Let $\{(X_i,\ y_i),\ i = 1, \cdots, n\}$ be the data.
- Let $y_i \in \{0,\ 1\}$.
- Let C_0 and C_1 denote the two classes. Thus, if $y_i = 0$ then $X_i \in C_0$ and if $y_i = 1$ then $X_i \in C_1$.
- Let n_0 and n_1 denote the number of examples of each class. ($n = n_0 + n_1$)
- For any W, let $z_i = W^T X_i$.
- z_i are the one dimensional data that we get after projection.

- Let M_0 and M_1 be the means of data from the two classes:

$$M_0 = \frac{1}{n_0} \sum_{X_i \in C_0} X_i; \quad M_1 = \frac{1}{n_1} \sum_{X_i \in C_1} X_i$$

- Let M_0 and M_1 be the means of data from the two classes:

$$M_0 = \frac{1}{n_0} \sum_{X_i \in C_0} X_i; \quad M_1 = \frac{1}{n_1} \sum_{X_i \in C_1} X_i$$

The corresponding means of the projected data would be

$$m_0 = W^T M_0 \quad \text{and} \quad m_1 = W^T M_1$$

- The difference $(m_0 - m_1)$ gives us an idea of the separation between samples of the two classes after projecting the data onto the direction W.

- The difference $(m_0 - m_1)$ gives us an idea of the separation between samples of the two classes after projecting the data onto the direction W.
- Hence, we may want a W that maximizes $(m_0 - m_1)^2$.

- The difference $(m_0 - m_1)$ gives us an idea of the separation between samples of the two classes after projecting the data onto the direction W.
- Hence, we may want a W that maximizes $(m_0 - m_1)^2$.
- However, we have to make this scale independent.

- The difference $(m_0 - m_1)$ gives us an idea of the separation between samples of the two classes after projecting the data onto the direction W.
- Hence, we may want a W that maximizes $(m_0 - m_1)^2$.
- However, we have to make this scale independent.
- Also, the distance between means should be viewed relative to the variances.

- Define

$$s_0^2 = \sum_{X_i \in C_0} (W^T X_i - m_0)^2; \quad s_1^2 = \sum_{X_i \in C_1} (W^T X_i - m_1)^2$$

These give us the variances (upto a factor) of the two classes in the projected data.

- Define

$$s_0^2 = \sum_{X_i \in C_0} (W^T X_i - m_0)^2; \quad s_1^2 = \sum_{X_i \in C_1} (W^T X_i - m_1)^2$$

  These give us the variances (upto a factor) of the two classes in the projected data.

- We want large separation between m_0 and m_1 relative to the variances.

- Hence we can take our objective to be to maximize

$$J(W) = \frac{(m_1 - m_0)^2}{s_0^2 + s_1^2}$$

- Hence we can take our objective to be to maximize

$$J(W) = \frac{(m_1 - m_0)^2}{s_0^2 + s_1^2}$$

- We now rewrite J into a more convenient form.

- Hence we can take our objective to be to maximize

$$J(W) = \frac{(m_1 - m_0)^2}{s_0^2 + s_1^2}$$

- We now rewrite J into a more convenient form.
- We have

$$(m_1 - m_0)^2 = (W^T M_1 - W^T M_0)^2$$

- Hence we can take our objective to be to maximize

$$J(W) = \frac{(m_1 - m_0)^2}{s_0^2 + s_1^2}$$

- We now rewrite J into a more convenient form.
- We have

$$\begin{aligned}(m_1 - m_0)^2 &= (W^T M_1 - W^T M_0)^2 \\ &= W^T (M_1 - M_0)(M_1 - M_0)^T W\end{aligned}$$

- Thus we have $(m_1 - m_0)^2 = W^T S_B W$ where

$$S_B = (M_1 - M_0)(M_1 - M_0)^T.$$

- Thus we have $(m_1 - m_0)^2 = W^T S_B W$ where

$$S_B = (M_1 - M_0)(M_1 - M_0)^T.$$

- Here, S_B is a $d \times d$ matrix (note that $X_i \in \Re^d$).

- Thus we have $(m_1 - m_0)^2 = W^T S_B W$ where

$$S_B = (M_1 - M_0)(M_1 - M_0)^T.$$

- Here, S_B is a $d \times d$ matrix (note that $X_i \in \Re^d$).
- It is called *between class* scatter matrix.

- Thus we have $(m_1 - m_0)^2 = W^T S_B W$ where

$$S_B = (M_1 - M_0)(M_1 - M_0)^T.$$

- Here, S_B is a $d \times d$ matrix (note that $X_i \in \Re^d$).
- It is called *between class* scatter matrix.
- We can similarly write s_0^2 and s_1^2 also as quadratic forms.

We have

$$s_0^2 = \sum_{X_i \in C_0} (W^T X_i - W^T M_0)^2$$

We have

$$
\begin{aligned}
s_0^2 &= \sum_{X_i \in C_0} (W^T X_i - W^T M_0)^2 \\
&= \sum_{X_i \in C_0} [W^T (X_i - M_0)]^2
\end{aligned}
$$

We have

$$
\begin{aligned}
s_0^2 &= \sum_{X_i \in C_0} (W^T X_i - W^T M_0)^2 \\
&= \sum_{X_i \in C_0} [W^T (X_i - M_0)]^2 \\
&= \sum_{X_i \in C_0} W^T (X_i - M_0)(X_i - M_0)^T W
\end{aligned}
$$

We have

$$
\begin{aligned}
s_0^2 &= \sum_{X_i \in C_0} (W^T X_i - W^T M_0)^2 \\
&= \sum_{X_i \in C_0} [W^T (X_i - M_0)]^2 \\
&= \sum_{X_i \in C_0} W^T (X_i - M_0)(X_i - M_0)^T W \\
&= W^T \left[\sum_{X_i \in C_0} (X_i - M_0)(X_i - M_0)^T \right] W
\end{aligned}
$$

- Similarly, we get

$$s_1^2 = W^T \left[\sum_{X_i \in C_1} (X_i - M_1)(X_i - M_1)^T \right] W$$

- Similarly, we get

$$s_1^2 = W^T \left[\sum_{X_i \in C_1} (X_i - M_1)(X_i - M_1)^T \right] W$$

- Thus we can write $s_0^2 + s_1^2 = W^T S_w W$, where

$$S_w = \sum_{X_i \in C_0} (X_i - M_0)(X_i - M_0)^T + \sum_{X_i \in C_1} (X_i - M_1)(X_i - M_1)^T$$

- Similarly, we get

$$s_1^2 = W^T \left[\sum_{X_i \in C_1} (X_i - M_1)(X_i - M_1)^T \right] W$$

- Thus we can write $s_0^2 + s_1^2 = W^T S_w W$, where

$$S_w = \sum_{X_i \in C_0} (X_i - M_0)(X_i - M_0)^T + \sum_{X_i \in C_1} (X_i - M_1)(X_i - M_1)^T$$

- S_w is also $d \times d$ matrix and is called *within class* scatter matrix.

- Hence we can now write J as

$$J(W) = \frac{W^T S_B W}{W^T S_w W}$$

- Hence we can now write J as

$$J(W) = \frac{W^T S_B W}{W^T S_w W}$$

- We want to find a W that maximizes $J(W)$.

- Hence we can now write J as

$$J(W) = \frac{W^T S_B W}{W^T S_w W}$$

- We want to find a W that maximizes $J(W)$.
- Note that $J(W)$ is not affected by scaling of W.

- Hence we can now write J as

$$J(W) = \frac{W^T S_B W}{W^T S_w W}$$

- We want to find a W that maximizes $J(W)$.
- Note that $J(W)$ is not affected by scaling of W.
- Given the data we can calculate the S_B and S_w.

- Hence we can now write J as

$$J(W) = \frac{W^T S_B W}{W^T S_w W}$$

- We want to find a W that maximizes $J(W)$.
- Note that $J(W)$ is not affected by scaling of W.
- Given the data we can calculate the S_B and S_w.
- Maximizing ratio of quadratic forms is a standard optimization problem.

- We need to maximize

$$J(W) = \frac{W^T S_B W}{W^T S_w W}$$

- We need to maximize

$$J(W) = \frac{W^T S_B W}{W^T S_w W}$$

- Differentiating w.r.t. W and equating to zero, we get

$$\frac{2S_B W}{W^T S_w W} - \frac{W^T S_B W}{(W^T S_w W)^2} 2S_w W = 0$$

- We need to maximize

$$J(W) = \frac{W^T S_B W}{W^T S_w W}$$

- Differentiating w.r.t. W and equating to zero, we get

$$\frac{2S_B W}{W^T S_w W} - \frac{W^T S_B W}{(W^T S_w W)^2} 2S_w W = 0$$

- Implies, $S_B W$ is in the same direction as $S_w W$.

- Thus, any maximizer of $J(W)$ has to satisfy

$$S_w W \;=\; \lambda\, S_B W$$

for some constant λ.

- Thus, any maximizer of $J(W)$ has to satisfy

$$S_w W \;=\; \lambda\, S_B W$$

  for some constant λ.
- This is known as the generalized eigen value problem.

- Thus, any maximizer of $J(W)$ has to satisfy

$$S_w W \;=\; \lambda\, S_B W$$

  for some constant λ.
- This is known as the generalized eigen value problem.
- There are standard methods to solve this problem using, e.g., LU decomposition.

- Thus, any maximizer of $J(W)$ has to satisfy

$$S_w W \;=\; \lambda\, S_B W$$

  for some constant λ.
- This is known as the generalized eigen value problem.
- There are standard methods to solve this problem using, e.g., LU decomposition.
- By solving the generalized eigen value problem we can find the best direction W.

- Often, the real symmetric matrix S_w would be invertible.

- Often, the real symmetric matrix S_w would be invertible.
- Recall that

$$S_w = \sum_{X_i \in C_0} (X_i - M_0)(X_i - M_0)^T + \sum_{X_i \in C_1} (X_i - M_1)(X_i - M_1)^T$$

- Often, the real symmetric matrix S_w would be invertible.
- Recall that

$$S_w = \sum_{X_i \in C_0} (X_i - M_0)(X_i - M_0)^T + \sum_{X_i \in C_1} (X_i - M_1)(X_i - M_1)^T$$

- This is a sum of large number of rank 1 matrices.

- Often, the real symmetric matrix S_w would be invertible.
- Recall that

$$S_w = \sum_{X_i \in C_0} (X_i - M_0)(X_i - M_0)^T + \sum_{X_i \in C_1} (X_i - M_1)(X_i - M_1)^T$$

- This is a sum of large number of rank 1 matrices.
- Also each term in S_w is proportional to the sample-mean-estimate of the covariance matrix of one of the class conditional densities.

- If S_w is invertible, then we can write

$$W = \lambda S_w^{-1} S_B W$$

- If S_w is invertible, then we can write

$$W \;=\; \lambda\, S_w^{-1} S_B W$$

- We have

$$S_B W \;=\; (M_1 - M_0)(M_1 - M_0)^T W = k(M_1 - M_0)$$

where k is some constant. (note $k = (m_1 - m_0)$)

- If S_w is invertible, then we can write

$$W \;=\; \lambda\, S_w^{-1} S_B W$$

- We have

$$S_B W \;=\; (M_1 - M_0)(M_1 - M_0)^T W = k(M_1 - M_0)$$

where k is some constant. (note $k = (m_1 - m_0)$)

- Now we get (since scale factor in W is not relevant)

$$W \;=\; S_w^{-1}(M_1 - M_0)$$

Obtaining Fisher Linear Discriminant

- We can sum-up the process as follows.

Obtaining Fisher Linear Discriminant

- We can sum-up the process as follows.
- Given the training data, we first form the scatter matrix S_w and also calculate the means M_0 and M_1.

Obtaining Fisher Linear Discriminant

- We can sum-up the process as follows.
- Given the training data, we first form the scatter matrix S_w and also calculate the means M_0 and M_1.
- If S_w is invertible, we calculate W by $W = S_w^{-1}(M_1 - M_0)$.

Obtaining Fisher Linear Discriminant

- We can sum-up the process as follows.
- Given the training data, we first form the scatter matrix S_w and also calculate the means M_0 and M_1.
- If S_w is invertible, we calculate W by $W = S_w^{-1}(M_1 - M_0)$.
- Even if S_w is not invertible, there are techniques to find the maximizer of $J(W)$ by solving the generalized eigen value problem.

Obtaining Fisher Linear Discriminant

- We can sum-up the process as follows.
- Given the training data, we first form the scatter matrix S_w and also calculate the means M_0 and M_1.
- If S_w is invertible, we calculate W by $W = S_w^{-1}(M_1 - M_0)$.
- Even if S_w is not invertible, there are techniques to find the maximizer of $J(W)$ by solving the generalized eigen value problem.
- Thus we can find the best direction W.

- The final discriminant function based classifier is sign$(W^T X + b)$.

- The final discriminant function based classifier is sign$(W^T X + b)$.
- So far, we have seen how to obtain best W.

- The final discriminant function based classifier is sign$(W^T X + b)$.
- So far, we have seen how to obtain best W.
- We still have to learn the best b also. But this is a simple threshold learning problem.

- The final discriminant function based classifier is sign$(W^T X + b)$.
- So far, we have seen how to obtain best W.
- We still have to learn the best b also. But this is a simple threshold learning problem.
- We can do a simple line search to find the threshold b to maximize probability of correct classification.

- The final discriminant function based classifier is sign$(W^T X + b)$.
- So far, we have seen how to obtain best W.
- We still have to learn the best b also. But this is a simple threshold learning problem.
- We can do a simple line search to find the threshold b to maximize probability of correct classification.
- Or, we can take the one dimensional projected data and learn the best classifier by, e.g., modelling the class conditional densities as normal.

- The final discriminant function based classifier is sign$(W^T X + b)$.
- So far, we have seen how to obtain best W.
- We still have to learn the best b also. But this is a simple threshold learning problem.
- We can do a simple line search to find the threshold b to maximize probability of correct classification.
- Or, we can take the one dimensional projected data and learn the best classifier by, e.g., modelling the class conditional densities as normal.
- This gives us the final Fisher Linear Discriminant (FLD) classifier.

- Fisher Linear Discriminant is also a popular classifier.

- Fisher Linear Discriminant is also a popular classifier.
- Though the method looks quite different from that of linear least squares there are close connections between the two.

- Fisher Linear Discriminant is also a popular classifier.
- Though the method looks quite different from that of linear least squares there are close connections between the two.
- We explain this connection next.

- Given the original training data $\{(X_i, y_i)\}$ we form new training data $\{(X_i, y_i')\}$ as follows.

- Given the original training data $\{(X_i, y_i)\}$ we form new training data $\{(X_i, y_i')\}$ as follows.
- We take $y_i' = n/n_0$ if $y_i = 0$ and $y_i' = -n/n_1$ if $y_i = 1$.

- Given the original training data $\{(X_i, y_i)\}$ we form new training data $\{(X_i, y_i')\}$ as follows.
- We take $y_i' = n/n_0$ if $y_i = 0$ and $y_i' = -n/n_1$ if $y_i = 1$.
- We now treat this as a data for a regression problem and learn a model $\hat{y} = W^T X + b$ using linear least squares.

- Given the original training data $\{(X_i, y_i)\}$ we form new training data $\{(X_i, y_i')\}$ as follows.
- We take $y_i' = n/n_0$ if $y_i = 0$ and $y_i' = -n/n_1$ if $y_i = 1$.
- We now treat this as a data for a regression problem and learn a model $\hat{y} = W^T X + b$ using linear least squares.
- It can be shown that the least squares solution for W would be same as that of FLD.

- Given the original training data $\{(X_i, y_i)\}$ we form new training data $\{(X_i, y_i')\}$ as follows.
- We take $y_i' = n/n_0$ if $y_i = 0$ and $y_i' = -n/n_1$ if $y_i = 1$.
- We now treat this as a data for a regression problem and learn a model $\hat{y} = W^T X + b$ using linear least squares.
- It can be shown that the least squares solution for W would be same as that of FLD.
- Thus FLD can be viewed as a special case of linear least squares.

- Consider a two class problem with class conditional densities to be normal with same covariance matrix.

- Consider a two class problem with class conditional densities to be normal with same covariance matrix.
- Let μ_0 and μ_1 be the two means and let Σ be the common covarinace matrix.

- Consider a two class problem with class conditional densities to be normal with same covariance matrix.
- Let μ_0 and μ_1 be the two means and let Σ be the common covarinace matrix.
- Suppose we we estimate the class conditional densities using ML method.

- Consider a two class problem with class conditional densities to be normal with same covariance matrix.
- Let μ_0 and μ_1 be the two means and let Σ be the common covarinace matrix.
- Suppose we we estimate the class conditional densities using ML method.
  Then the M_0 and M_1 would be the sample mean estimates for μ_0 and μ_1.

- Consider a two class problem with class conditional densities to be normal with same covariance matrix.
- Let μ_0 and μ_1 be the two means and let Σ be the common covarinace matrix.
- Suppose we we estimate the class conditional densities using ML method.
  Then the M_0 and M_1 would be the sample mean estimates for μ_0 and μ_1.

  Let $\hat{\Sigma}$ be the sample mean estimate for covariance matrix.

- Then the Bayes classifier, implemented with estimated class conditional densities, would be a linear classifier sign$(W^T X + b)$ where W is given by

$$W = \hat{\Sigma}^{-1}(M_1 - M_0)$$

- Then the Bayes classifier, implemented with estimated class conditional densities, would be a linear classifier sign$(W^T X + b)$ where W is given by

$$W = \hat{\Sigma}^{-1}(M_1 - M_0)$$

- It is easily seen that this would be essentially the same as the W given by FLD.

- Recall that the matrix S_w is defined by

$$S_w = \sum_{X_i \in C_0} (X_i - M_0)(X_i - M_0)^T + \sum_{X_i \in C_1} (X_i - M_1)(X_i - M_1)^T$$

- Recall that the matrix S_w is defined by

$$S_w = \sum_{X_i \in C_0} (X_i - M_0)(X_i - M_0)^T + \sum_{X_i \in C_1} (X_i - M_1)(X_i - M_1)^T$$

Since the two classes have the same covariance matrix, each of the two terms above would be proportional to sample mean estimator for Σ.

- Recall that the matrix S_w is defined by

$$S_w = \sum_{X_i \in C_0} (X_i - M_0)(X_i - M_0)^T + \sum_{X_i \in C_1} (X_i - M_1)(X_i - M_1)^T$$

  Since the two classes have the same covariance matrix, each of the two terms above would be proportional to sample mean estimator for Σ.
- Thus, S_w is proportional to sample mean estimator of covariance matrix.

- Recall that the matrix S_w is defined by

$$S_w = \sum_{X_i \in C_0} (X_i - M_0)(X_i - M_0)^T + \sum_{X_i \in C_1} (X_i - M_1)(X_i - M_1)^T$$

  Since the two classes have the same covariance matrix, each of the two terms above would be proportional to sample mean estimator for Σ.
- Thus, S_w is proportional to sample mean estimator of covariance matrix.
- Thus, the FLD given by $W = S_w^{-1}(M_1 - M_0)$ would be essentially same as the Bayes optimal classifier.

- We have considered various methods of learning linear classifiers and regression functions.

- We have considered various methods of learning linear classifiers and regression functions.
- In the case of regression, we have considered only estimating real-valued functions. We can generalize this to vector-valued functions.

- We have considered various methods of learning linear classifiers and regression functions.
- In the case of regression, we have considered only estimating real-valued functions. We can generalize this to vector-valued functions.
- In classification we considered only 2-class problems. This can also be generalized to multi-class case.

- First consider estimating vector-valued functions.

- First consider estimating vector-valued functions.
- Now the training data is $\{(X_i, y_i),\ i = 1, \cdots, n\}$ where $X_i \in \Re^d$ and $y_i = (y_i^1, \cdots, y_i^m) \in \Re^m$.

- First consider estimating vector-valued functions.
- Now the training data is $\{(X_i, y_i),\ i = 1, \cdots, n\}$ where $X_i \in \Re^d$ and $y_i = (y_i^1, \cdots, y_i^m) \in \Re^m$.
- For any given X we want to predict the target $y = (y^1, \cdots, y^m)$.

- First consider estimating vector-valued functions.
- Now the training data is $\{(X_i, y_i),\ i = 1, \cdots, n\}$ where $X_i \in \Re^d$ and $y_i = (y_i^1, \cdots, y_i^m) \in \Re^m$.
- For any given X we want to predict the target $y = (y^1, \cdots, y^m)$.
- Thus we want to learn W_j, b_j, $j = 1, \cdots, m$ so that

$$\hat{y}^j \ = \ W_j^T X + b_j,\ j = 1, \cdots, m$$

- First consider estimating vector-valued functions.
- Now the training data is $\{(X_i, y_i),\ i = 1, \cdots, n\}$ where $X_i \in \Re^d$ and $y_i = (y_i^1, \cdots, y_i^m) \in \Re^m$.
- For any given X we want to predict the target $y = (y^1, \cdots, y^m)$.
- Thus we want to learn W_j, b_j, $j = 1, \cdots, m$ so that

$$\hat{y}^j \ = \ W_j^T X + b_j,\ j = 1, \cdots, m$$

- We can obtain these by simply solving m number of linear least squares regression problems.

- Now let us consider the multi-class problem.

- Now let us consider the multi-class problem.
- Suppose we have K classes: $C_1, \cdots, C_K$.

- Now let us consider the multi-class problem.
- Suppose we have K classes: $C_1, \cdots, C_K$.
- We can solve the multi-class problem by learning a number of 2-class classifiers.

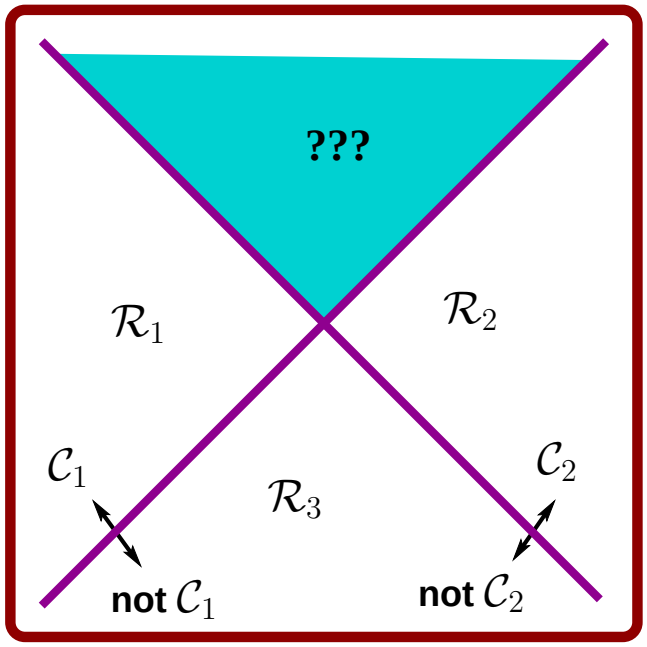
- Now let us consider the multi-class problem.
- Suppose we have K classes: $C_1, \cdots, C_K$.
- We can solve the multi-class problem by learning a number of 2-class classifiers.
- For example, we can learn K two class classifiers: 'C_i Vs not-C_i'. (Called one versus rest).

- Now let us consider the multi-class problem.
- Suppose we have K classes: $C_1, \cdots, C_K$.
- We can solve the multi-class problem by learning a number of 2-class classifiers.
- For example, we can learn K two class classifiers: 'C_i Vs not-C_i'. (Called one versus rest).
- Or we can learn $K(K-1)/2$ number of 2-class classifiers: 'C_i Vs C_j'

- Now let us consider the multi-class problem.
- Suppose we have K classes: $C_1, \cdots, C_K$.
- We can solve the multi-class problem by learning a number of 2-class classifiers.
- For example, we can learn K two class classifiers: 'C_i Vs not-C_i'. (Called one versus rest).
- Or we can learn $K(K-1)/2$ number of 2-class classifiers: 'C_i Vs C_j'
- But neither of these approaches are really satisfactory for generalizing linear discriminant functions.

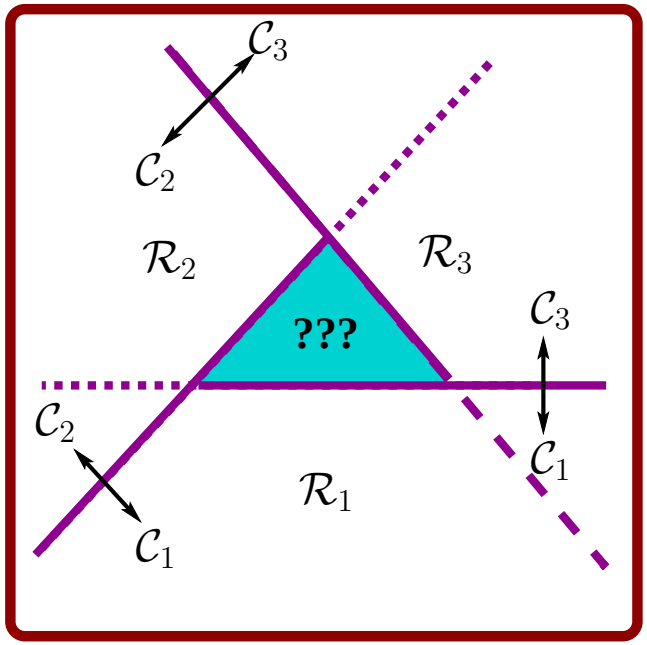
- Suppose we try ‘one versus rest’ approach. Then there may be regions of feature space where classification is ambiguous.

- Suppose we try ‘one versus rest’ approach. Then there may be regions of feature space where classification is ambiguous.



- Similar problem exists when we try 'C_i Vs C_j' approach

- Similar problem exists when we try 'C_i Vs C_j' approach



- A better way to formulate a linear discriminant function based classifier for the multi-class case is as follows.

- A better way to formulate a linear discriminant function based classifier for the multi-class case is as follows.
- We will have K functions, g_s, $s = 1, \cdots, K$, given by

$$g_s(X) = W_s^T X + b_s$$

- A better way to formulate a linear discriminant function based classifier for the multi-class case is as follows.
- We will have K functions, g_s, $s = 1, \cdots, K$, given by

$$g_s(X) = W_s^T X + b_s$$

- Now the classifier would assign class C_j to X if

$$g_j(X) \geq g_s(X), \ \forall s$$

- A better way to formulate a linear discriminant function based classifier for the multi-class case is as follows.
- We will have K functions, g_s, $s = 1, \cdots, K$, given by

$$g_s(X) = W_s^T X + b_s$$

- Now the classifier would assign class C_j to X if

$$g_j(X) \geq g_s(X), \ \forall s$$

- In the above we would have a fixed (may be arbitrary) rule for breaking ties.

- A better way to formulate a linear discriminant function based classifier for the multi-class case is as follows.
- We will have K functions, g_s, $s = 1, \cdots, K$, given by

$$g_s(X) = W_s^T X + b_s$$

- Now the classifier would assign class C_j to X if

$$g_j(X) \geq g_s(X), \ \forall s$$

- In the above we would have a fixed (may be arbitrary) rule for breaking ties.
- Recall that this is the way we generalized Bayes classifier to multi-class case.

- Now, to learn a linear classifier for the K-class case, we need to learn the K functions g_s.

- Now, to learn a linear classifier for the K-class case, we need to learn the K functions g_s.
- The simplest way to do this is to make the class label to be a vector of K components.

- Now, to learn a linear classifier for the K-class case, we need to learn the K functions g_s.
- The simplest way to do this is to make the class label to be a vector of K components.
- If $X_i \in C_j$ then y_i would be a K-vector with j^{th} component one and all others zero.

- Now, to learn a linear classifier for the K-class case, we need to learn the K functions g_s.
- The simplest way to do this is to make the class label to be a vector of K components.
- If $X_i \in C_j$ then y_i would be a K-vector with j^{th} component one and all others zero.
- Now learning the K functions is same as linear regression with vector valued targets.

- Now, to learn a linear classifier for the K-class case, we need to learn the K functions g_s.
- The simplest way to do this is to make the class label to be a vector of K components.
- If $X_i \in C_j$ then y_i would be a K-vector with j^{th} component one and all others zero.
- Now learning the K functions is same as linear regression with vector valued targets.
- We can similarly generalize logistic regression and FLD also for the K-class case.