Classification

Lecture 16

Classifiers

Covered so far

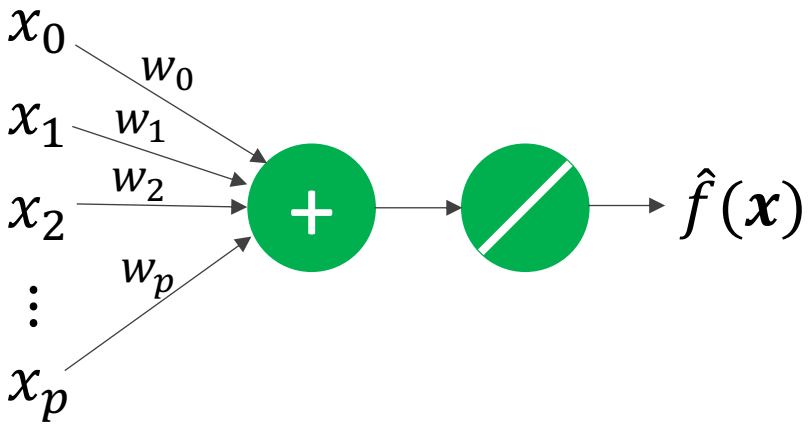
- K-Nearest Neighbors
- Linear regression (can be used, but not recommended)
- Perceptron
- Logistic Regression
- Fisher's Linear Discriminant / Linear Discriminant Analysis
- Quadratic Discriminant Analysis
- Naïve Bayes

Rely on a linear combination of weights and features: $\boldsymbol{w}^T \boldsymbol{x}$

Remember linear models?

Linear Regression

$$\hat{f}(\boldsymbol{x}) = \sum_{i=0}^{N} w_i x_i$$

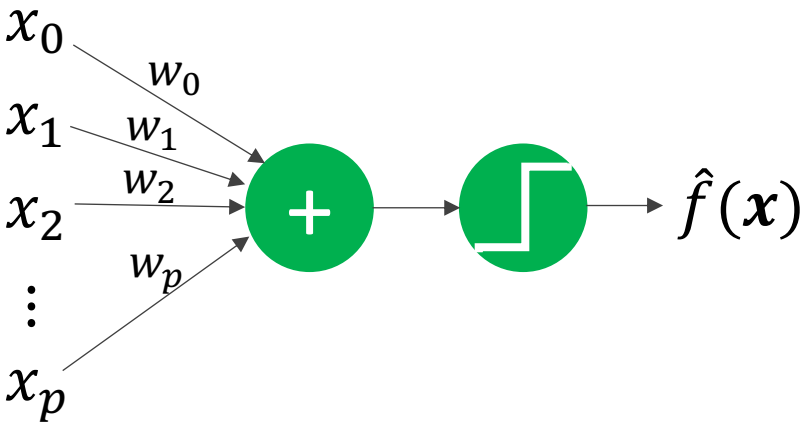


Linear Classification

Perceptron

$$\hat{f}(\boldsymbol{x}) = sign\left(\sum_{i=0}^{N} w_i x_i\right)$$

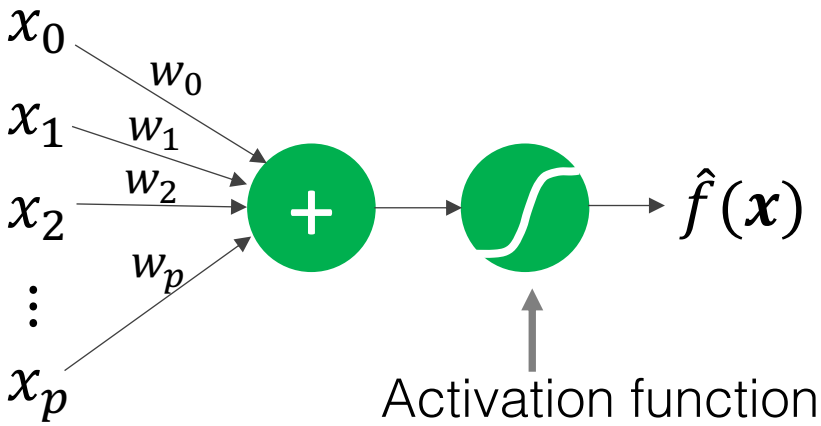
$$sign(x) = \begin{cases} 1 & x > 0 \\ -1 & \text{else} \end{cases}$$



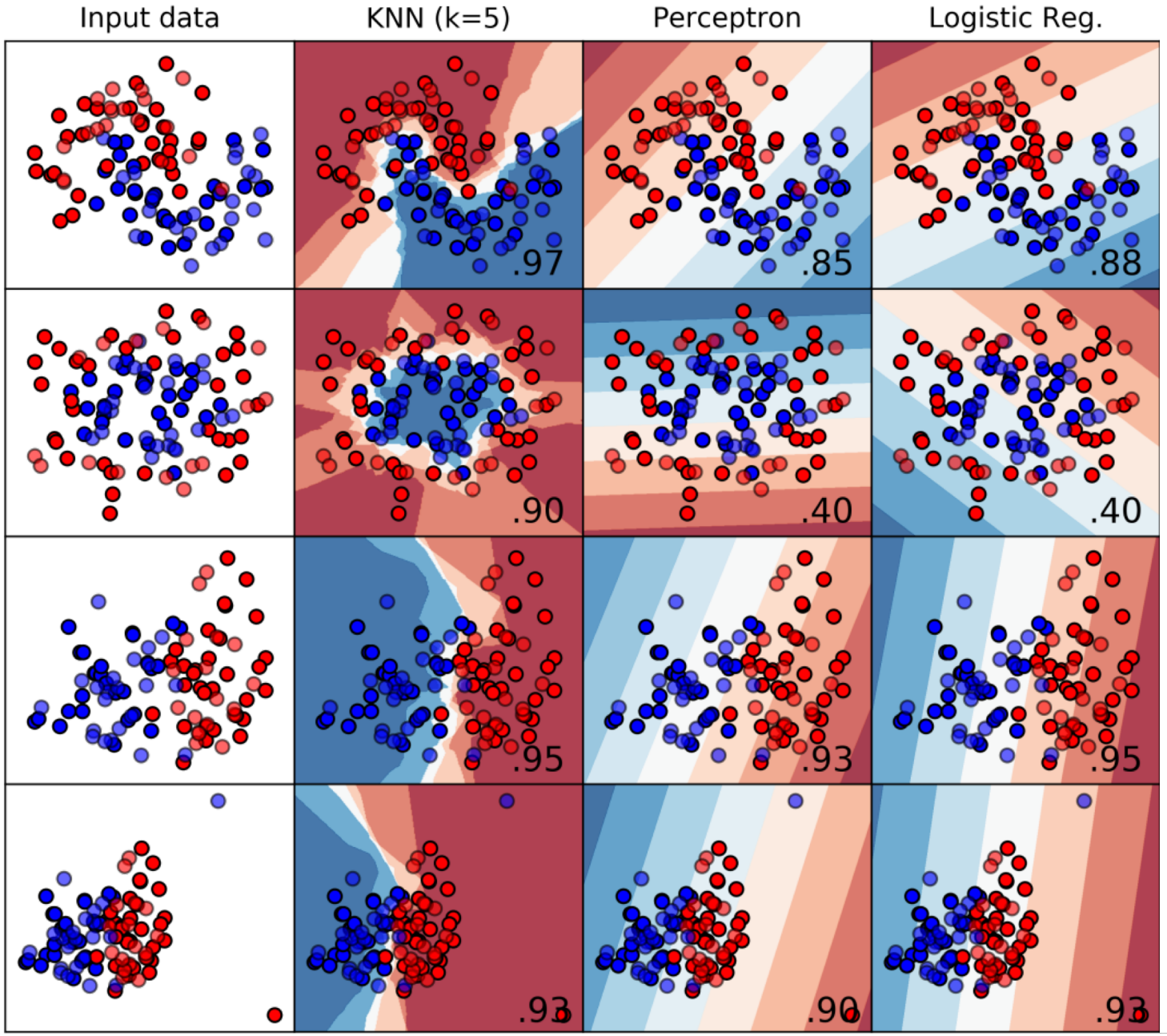
Logistic Regression

$$\hat{f}(\boldsymbol{x}) = \sigma\left(\sum_{i=0}^{N} w_i x_i\right)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



Source: Abu-Mostafa, Learning from Data, Caltech

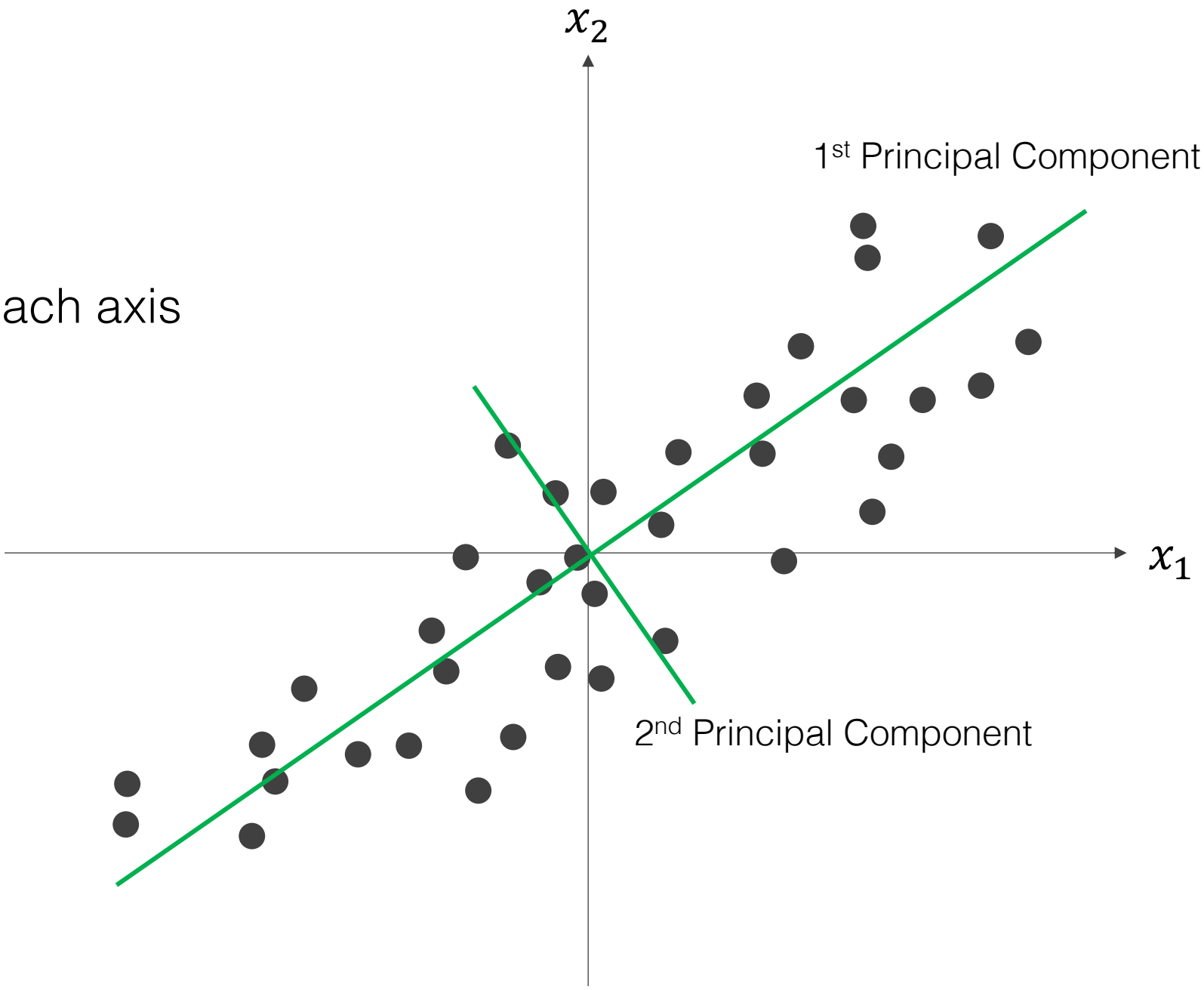


Comparison of classifiers we've seen so far

Principal Components

Maximize the variance along each axis

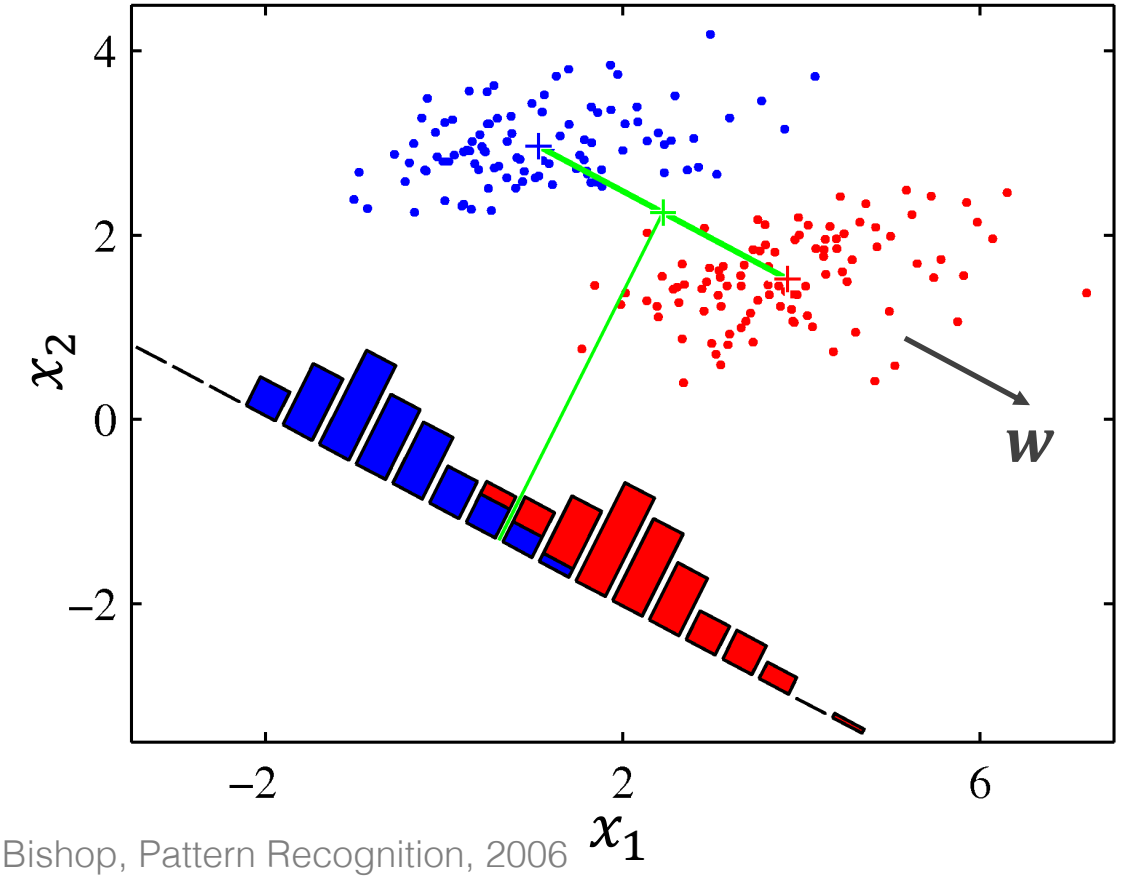
We want to maximize the **separability** of classes instead



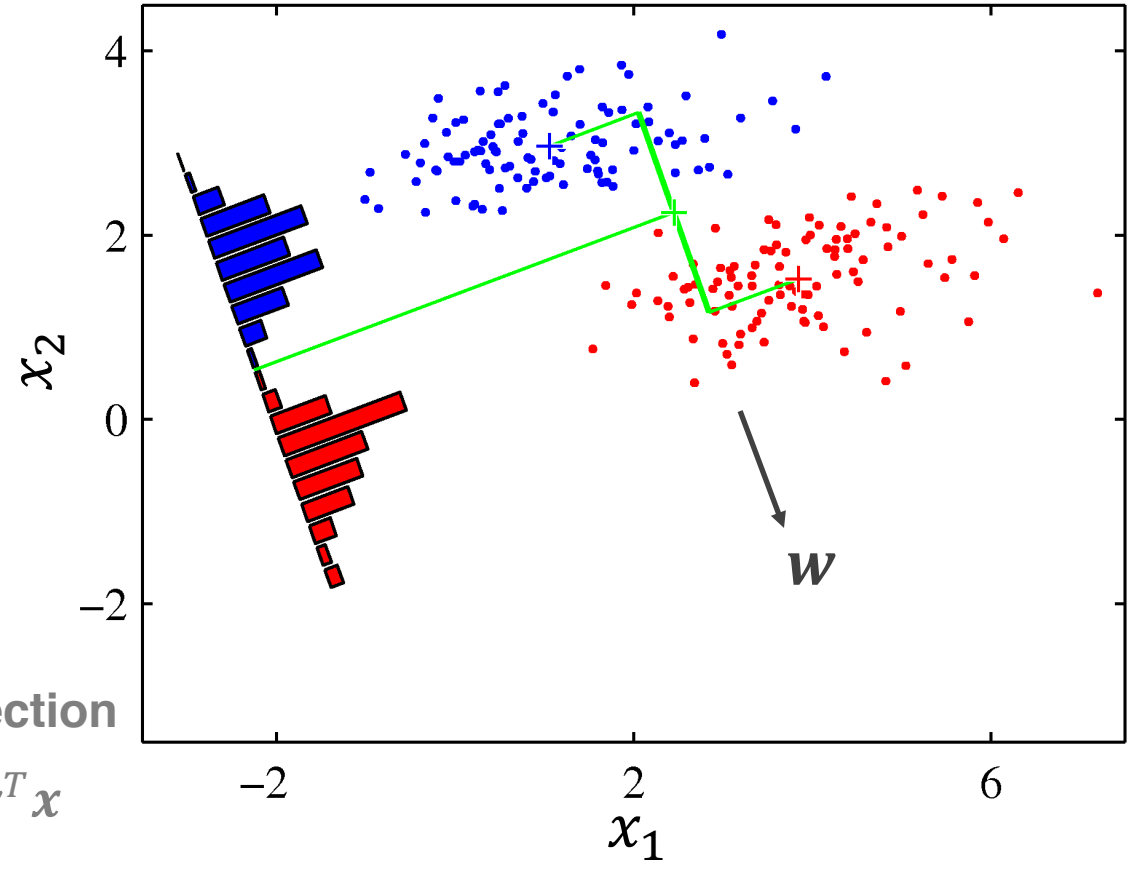
Fisher's Linear Discriminant

Looks for the projection into the one dimension that "best" separates the classes

Projection onto line connecting the means



Projection onto a line providing improved class separation



Linear projection

$$\hat{f}(x) = w^T x$$

Bishop, Pattern Recognition, 2006

Fisher's Linear Discriminant (FLD)

1. Finds a projection into a lower dimension that "best" separates the classes

$$\hat{f}(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$$

2. We then classify the data in this space

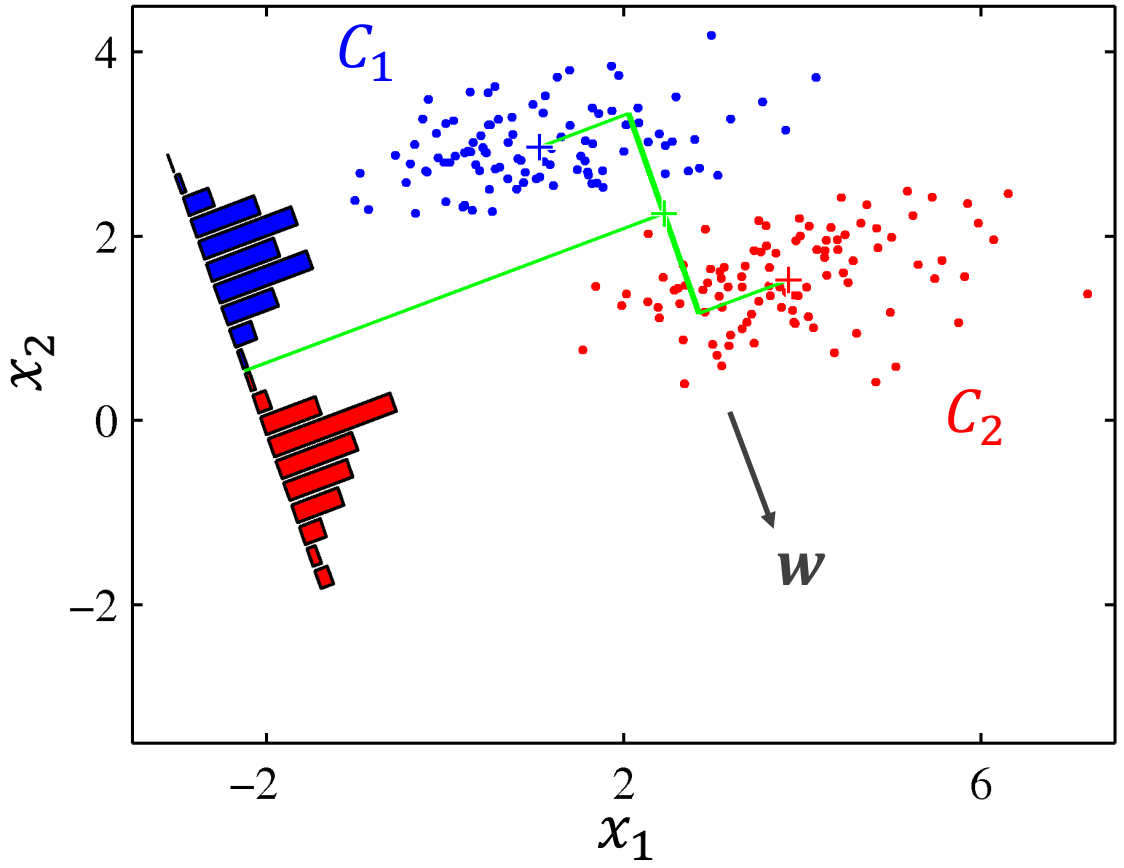
Similar to PCA, but accounts for class separability

Our decision rule becomes:

$$\text{if} \quad \hat{f}(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} > \lambda_{thresh} \qquad \text{Class 1}$$

$$\text{else} \qquad \text{Class 2}$$

FLD: how do we choose the vector w?

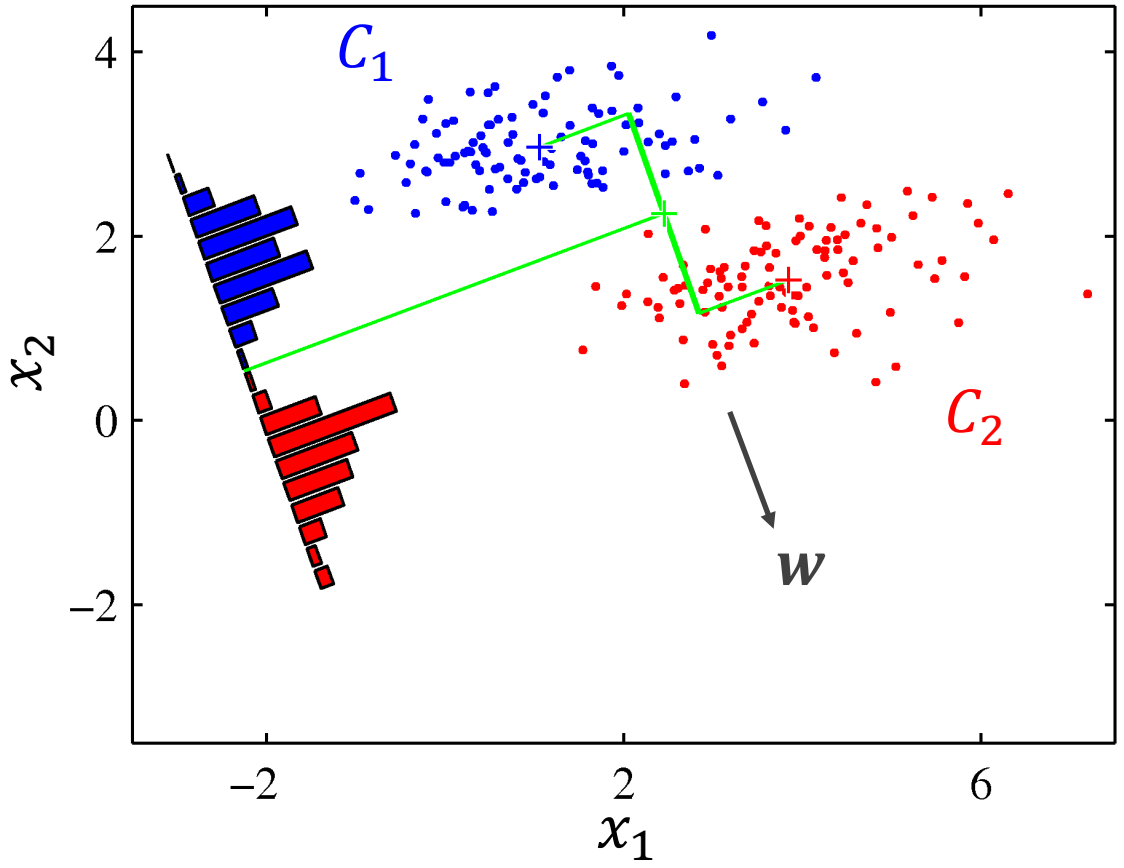


$$y = \hat{f}(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$$

Increase the distance between the **means**

Decrease the **variance** within each class

FLD: how do we choose the vector w?



$$y = \hat{f}(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$$

Increase the distance between the **means**

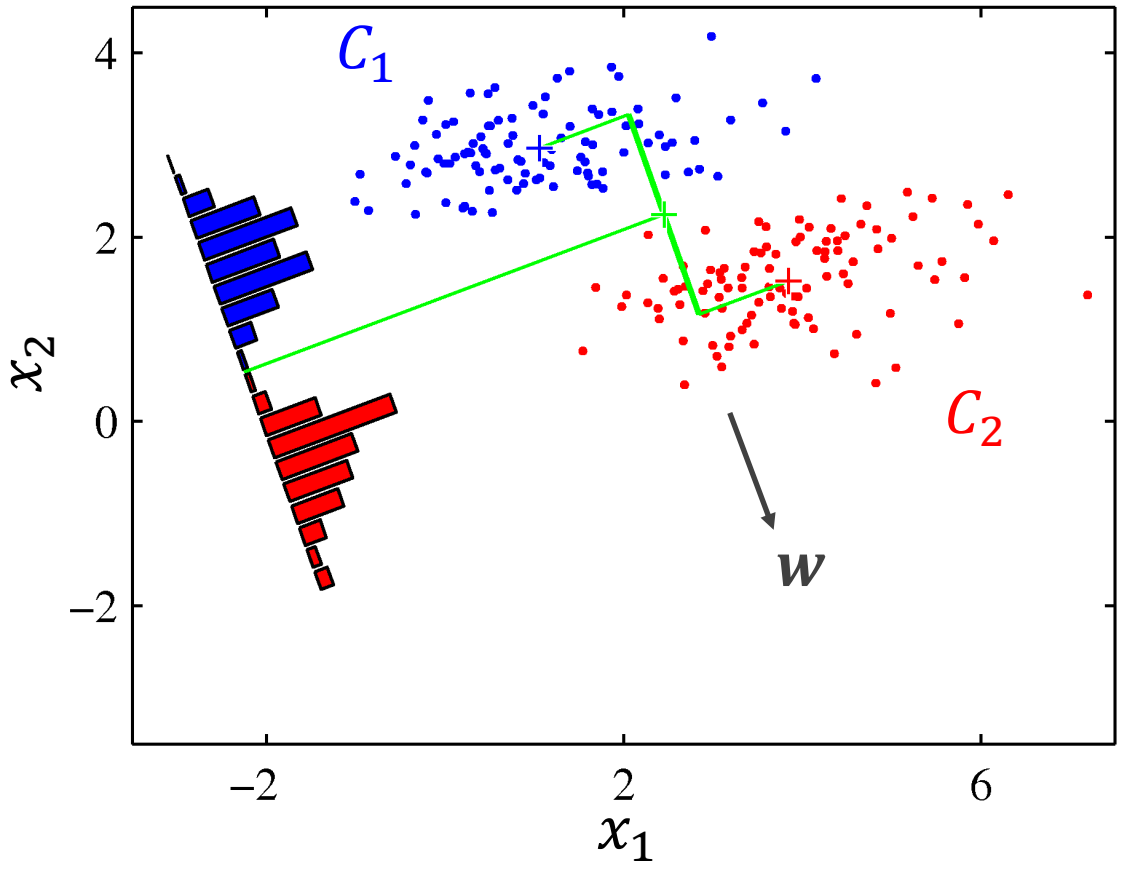
$$\boldsymbol{m}_1 = \frac{1}{N_1} \sum_{i \in C_1} \boldsymbol{x}_i \qquad \boldsymbol{m}_2 = \frac{1}{N_2} \sum_{i \in C_2} \boldsymbol{x}_i$$

The means projected onto $\boldsymbol{w}$: $m_k = \boldsymbol{w}^T \boldsymbol{m}_k$

The distance between the means:

$$m_2 - m_1 = \boldsymbol{w}^T (\boldsymbol{m}_2 - \boldsymbol{m}_1)$$

FLD: how do we choose the vector w?



$$y = \hat{f}(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$$

Decrease the **variance** within each class

The "scatter" of the projected data:

$$s_k^2 = \sum_{i \in C_k} (y_i - m_k)^2$$

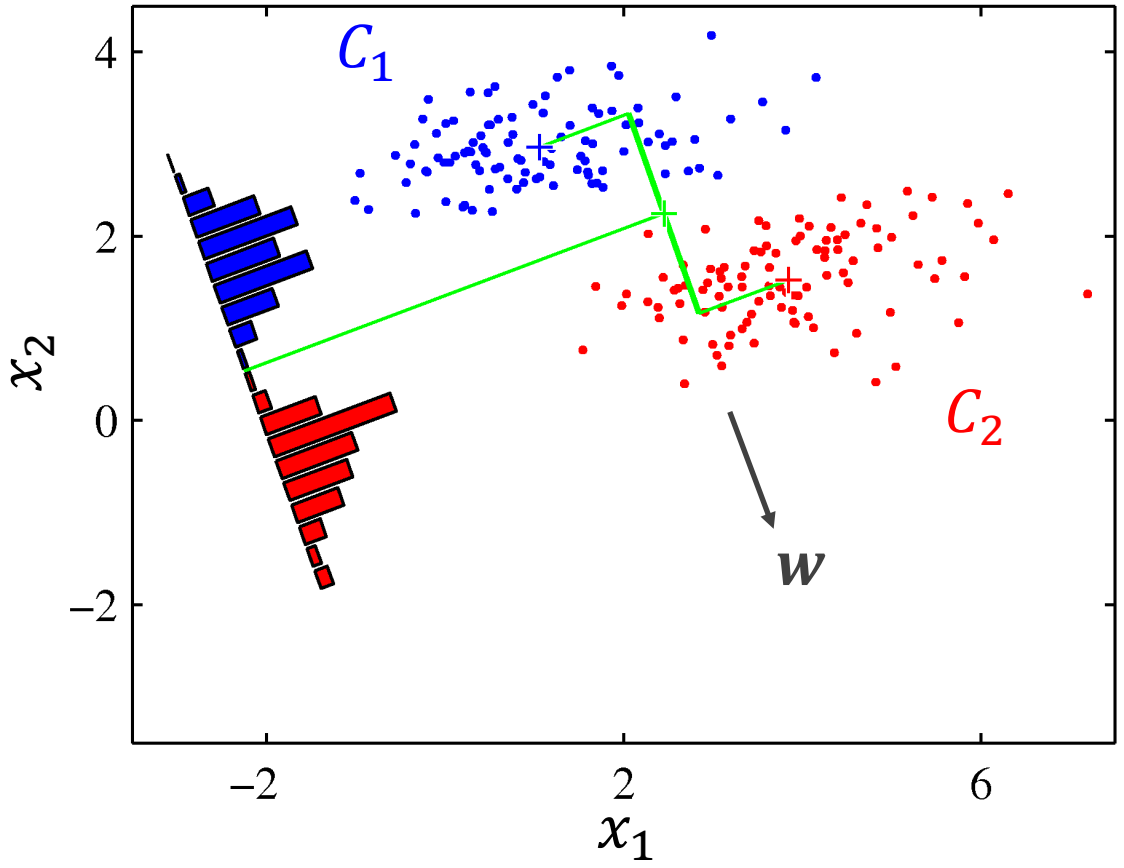
where $m_k = \boldsymbol{w}^T \boldsymbol{m}_k$

$$y_i = \boldsymbol{w}^T \boldsymbol{x}_i$$

Therefore the total within-class scatter:

$$S = s_1^2 + s_2^2$$

FLD: how do we choose the vector w?



$$y = \hat{f}(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$$

Increase the distance between the **means**

$$m_2 - m_1 = \boldsymbol{w}^T(\boldsymbol{m}_2 - \boldsymbol{m}_1)$$

Decrease the **variance** within each class

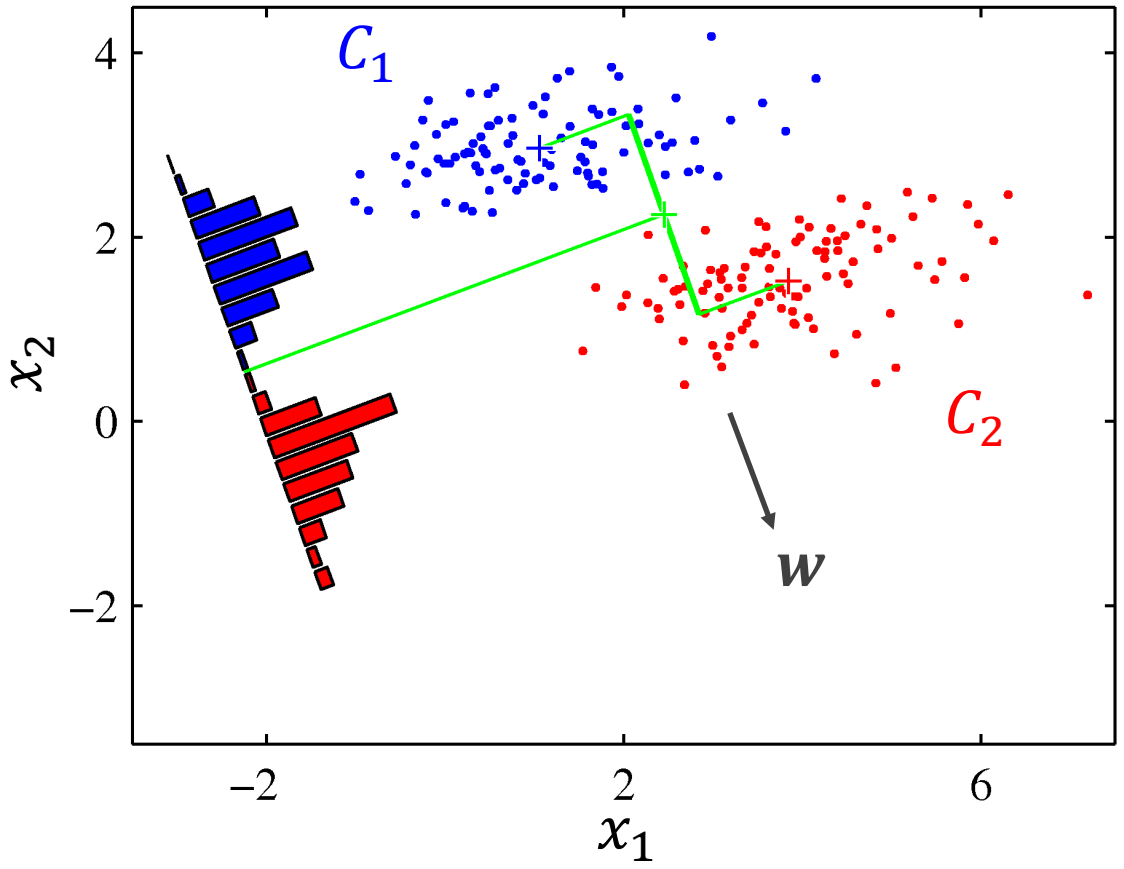
$$S = s_1^2 + s_2^2$$

The Fisher criterion is then:

$$J(\boldsymbol{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

We want to maximize this and solve for $\boldsymbol{w}$

FLD: how do we choose the vector $\boldsymbol{w}$?



$$y = \hat{f}(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$$

We want to maximize this and solve for $\boldsymbol{w}$

$$J(\boldsymbol{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

$$= \frac{\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_W \boldsymbol{w}}$$

(see appendix slides for full derivation)

Take the derivative (gradient), set it equal to zero, solve for $\boldsymbol{w}$

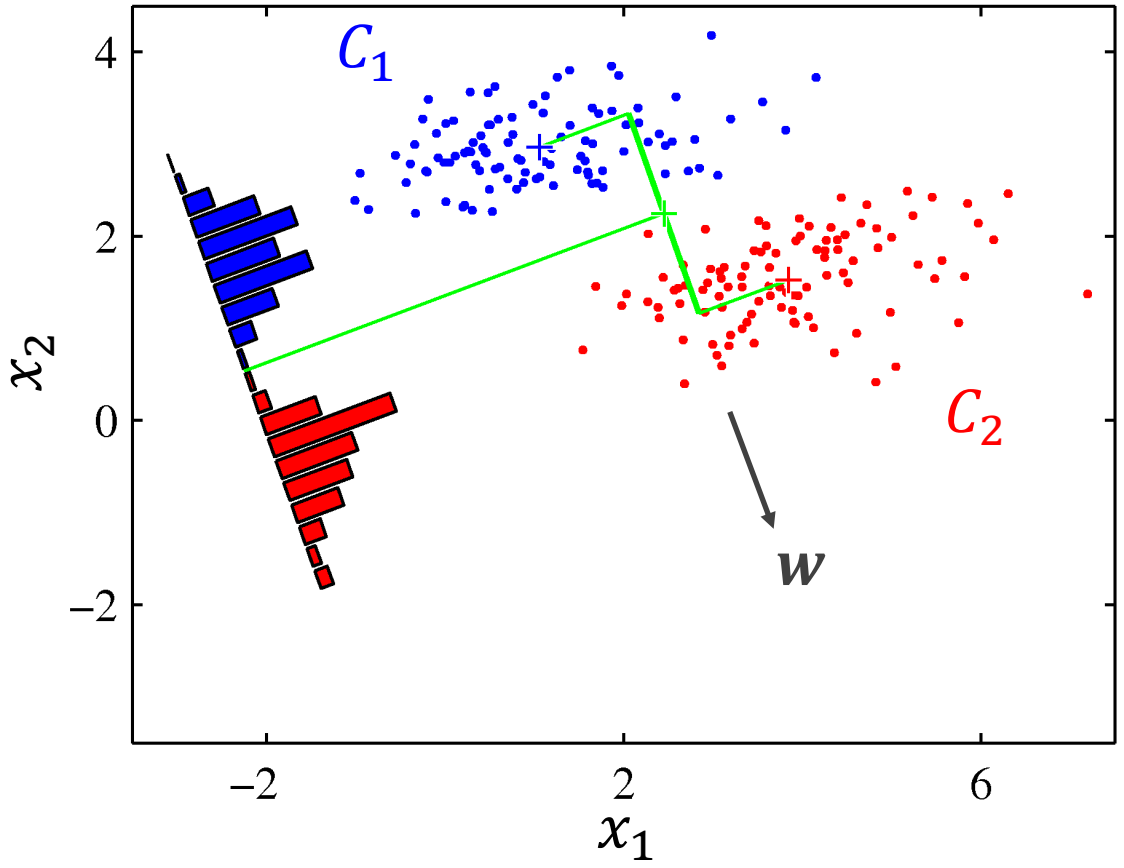
(see appendix slides for full derivation)

$$\boldsymbol{w} \propto \boldsymbol{S}_W^{-1}(\boldsymbol{m}_2 - \boldsymbol{m}_1)$$

$$\boldsymbol{w} \propto (\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)^{-1}(\boldsymbol{m}_2 - \boldsymbol{m}_1)$$

We use this to project the features into one dimension for classification, $\boldsymbol{w}^T \boldsymbol{x}$

Fisher's Linear Discriminant



$$y = \hat{f}(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$$

Makes no assumptions about the distribution of the data, and allows for different covariance matrices

This is a **projection** into one dimension that can be used to construct a discriminant

$$\boldsymbol{w} \propto \boldsymbol{S}_W^{-1}(\boldsymbol{m}_2 - \boldsymbol{m}_1)$$

$$\boldsymbol{w} \propto (\Sigma_1 + \Sigma_2)^{-1}(\boldsymbol{m}_2 - \boldsymbol{m}_1)$$

Discriminant Functions

If we have c different classes, we define a discriminant function, $d_i(\boldsymbol{x})$ for $i = 1, \ldots, c$

If $d_i(\boldsymbol{x}) > d_j(\boldsymbol{x})$ for $i \neq j$, then we assign feature $\boldsymbol{x}$ to class i

$$d_i(x) = P(y = i|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y = i)P(y = i)}{P(\boldsymbol{x})}$$

Bayes' Rule: $P(Y|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|Y)P(Y)}{P(\boldsymbol{X})}$ (Posterior, Likelihood, Prior, Evidence)

$$= \frac{P(\boldsymbol{x}|y = i)P(y = i)}{\sum_{i=1}^{c} P(\boldsymbol{x}|y = i)P(y = i)}$$ → This denominator will be the same for all i

We can simply write $d_i(x) = P(\boldsymbol{x}|y = i)P(y = i)$

Or in log form: $\ln d_i(x) = \ln P(\boldsymbol{x}|y = i) + \ln P(y = i)$

Discriminant Functions

$$d_i(x) = P(\boldsymbol{x}|y = i)P(y = i)$$

$$\ln d_i(x) = \ln P(\boldsymbol{x}|y = i) + \ln P(y = i)$$

Equivalent discriminant functions since log is monotonic

1. Assume a form for $P(\boldsymbol{x}|y = i)$

   Gaussian for Linear and Quadratic Discriminant Analysis
   Gaussian mixture models
   Nonparametric density estimates
   Naïve Bayes models

2. Assign the class, i, for which $d_i(x)$ is largest

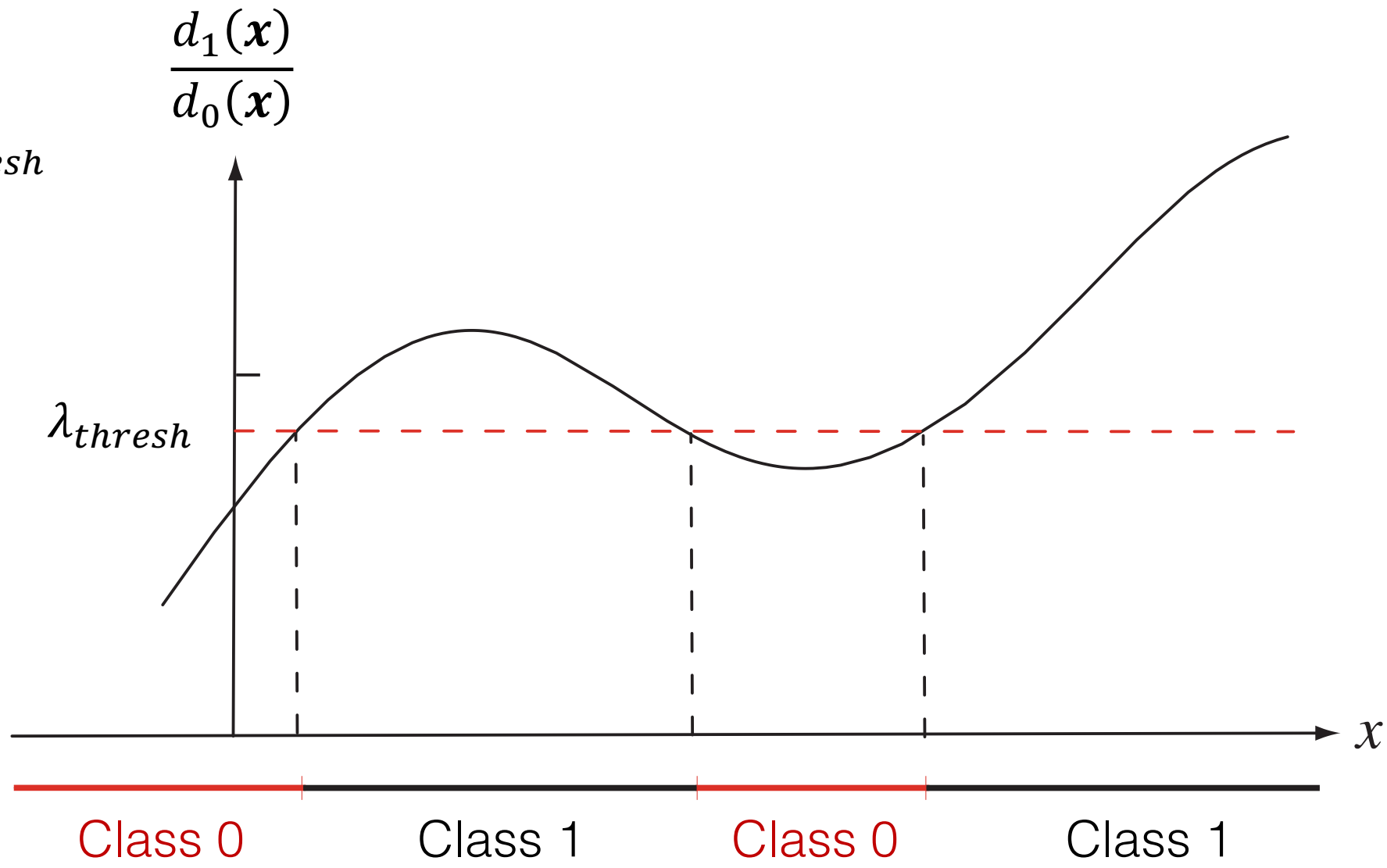
   Applies to both binary and multiclass problems

Discriminant Functions: 2 classes

Decision rule:

Class 1 if: $\frac{d_1(\boldsymbol{x})}{d_0(\boldsymbol{x})} > \lambda_{thresh}$

Otherwise, class 0



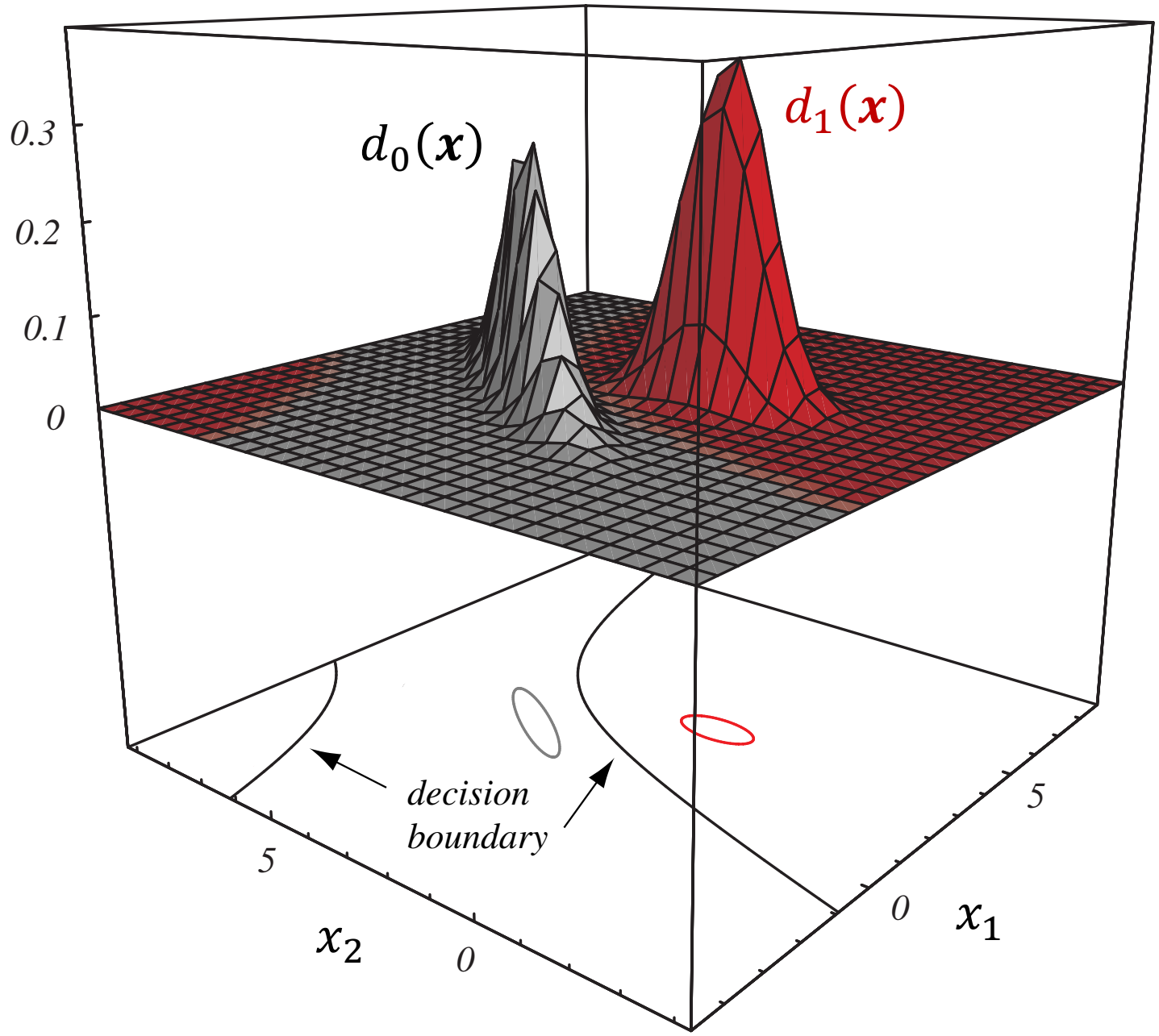
Duda, Hart, and Stork, Pattern Classification, 2006

Discriminant Functions: 2 classes, 2 dimensions

Decision rule:

Class 1 if: $\frac{d_1(\boldsymbol{x})}{d_0(\boldsymbol{x})} > \lambda_{thresh}$

Otherwise, class 0



Duda, Hart, and Stork, Pattern Classification, 2006

Discriminant Function: 2 classes

We build a classifier that assigns the class with the higher posterior probability:

If $$\frac{P(y=1|\boldsymbol{x})}{P(y=0|\boldsymbol{x})} = \frac{P(\boldsymbol{x}|y=1)P(y=1)}{P(\boldsymbol{x}|y=0)P(y=0)} > 1$$ Assign class 1, else class 0

Assumes these likelihoods are normal

$$= \frac{P(\boldsymbol{x}|y=1)}{P(\boldsymbol{x}|y=0)} > \frac{P(y=0)}{P(y=1)}$$

$N(\mu_1, \Sigma_1)$

$N(\mu_0, \Sigma_0)$

Estimate the class-conditional mean and covariance matrix from the data

Discriminant Function: 2 classes

We build a classifier that assigns the class with the higher posterior probability:

Likelihood ratio:

$$\frac{P(\boldsymbol{x}|y=1)}{P(\boldsymbol{x}|y=0)} > \frac{P(y=0)}{P(y=1)}$$

$P(\boldsymbol{x}|y=1) \sim N(\mu_1, \Sigma_1)$

$P(\boldsymbol{x}|y=0) \sim N(\mu_0, \Sigma_0)$

If we assume the class conditional distributions are Gaussian, this represents

Quadratic Discriminant Analysis

If we further assume the covariance matrices are the same, $\Sigma_0 = \Sigma_1$, this represents

Linear Discriminant Analysis

Comparison

	Fisher's Linear Discriminant (FLD)	**Quadratic Discriminant Analysis (QDA)**	**Linear Discriminant Analysis (LDA)**
Assumes Gaussian Likelihood $P(\boldsymbol{x}\|y)$ (class conditional density)	No	Yes	Yes
Assumes equivalent covariance matrices $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_0$	No	No	Yes

FLD and LDA reduce the dimensionality of the data to make them more separable

FLD and LDA are often presented as the same, but there are differences

Linear Discriminant Analysis ($\Sigma_0 = \Sigma_1$)

We build a classifier that assigns the class with the higher posterior probability:

$$\frac{P(\boldsymbol{x}|y=1)}{P(\boldsymbol{x}|y=0)} = \frac{\frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)\right]}{\frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_0)\right]}$$

$$\ln\left|\frac{P(\boldsymbol{x}|y=1)}{P(\boldsymbol{x}|y=0)}\right| = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_0)$$

$$\ln\left|\frac{P(\boldsymbol{x}|y=1)}{P(\boldsymbol{x}|y=0)}\right| = \boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0) - \frac{1}{2}(\boldsymbol{\mu}_1+\boldsymbol{\mu}_0)\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)$$

(see appendix slides for full derivation)

Linear Discriminant Analysis ($\Sigma_0 = \Sigma_1$)

$$\ln \left| \frac{P(\boldsymbol{x}|y=1)}{P(\boldsymbol{x}|y=0)} \right| = \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0)\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$$

Since our decision rule is to classify as class 1 if the following is true:

$$\ln \left| \frac{P(\boldsymbol{x}|y=1)}{P(\boldsymbol{x}|y=0)} \right| > \ln \left| \frac{P(y=0)}{P(y=1)} \right|$$

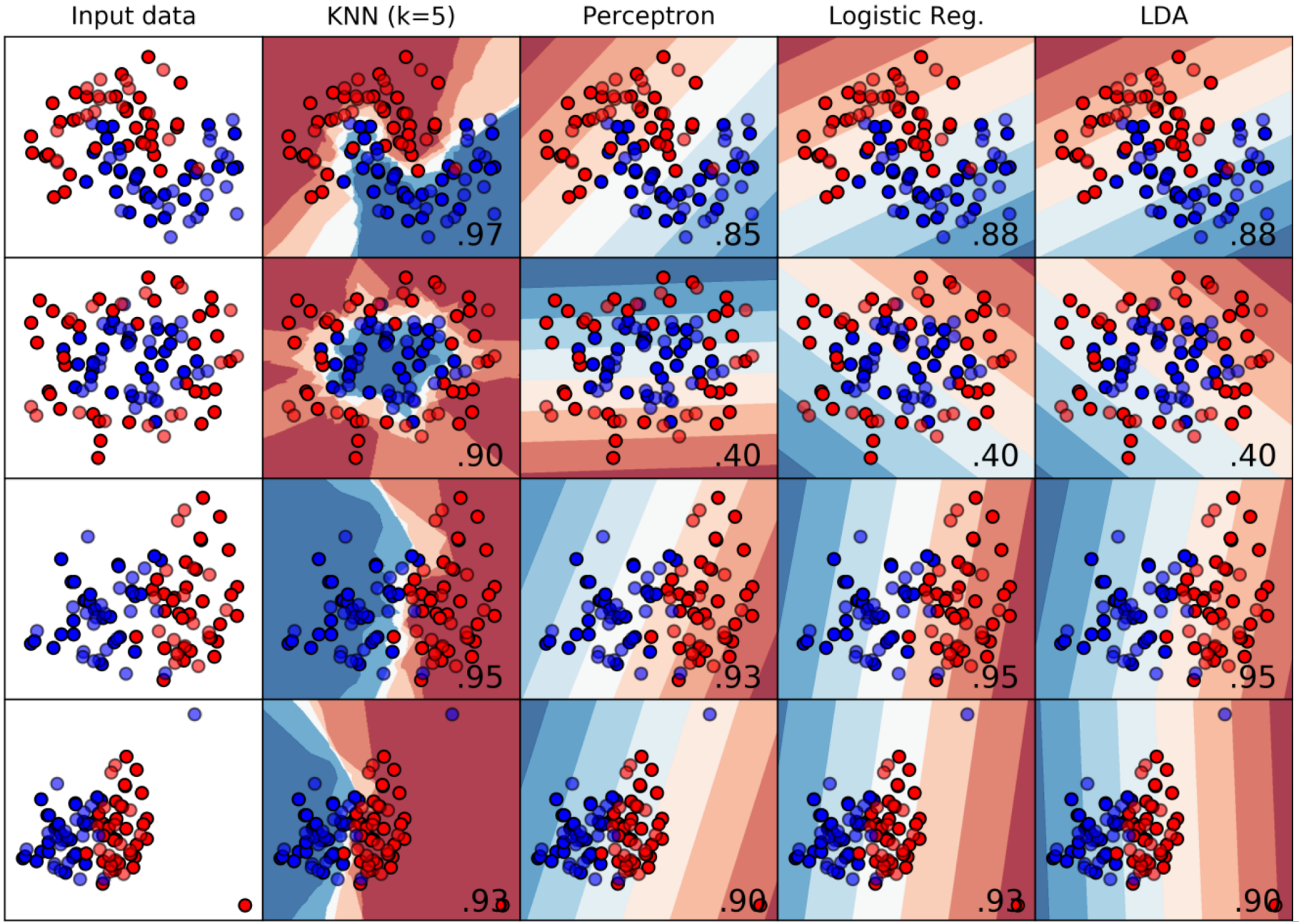
We can rewrite our decision rule as:

$$\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) > \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0)\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) + \ln \left| \frac{P(y=0)}{P(y=1)} \right|$$

Or simply as:

$$\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) > \lambda_{thresh}$$

If we define $\boldsymbol{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$, then this becomes $\boldsymbol{x}^T \boldsymbol{w} > \lambda_{thresh}$


Input data
KNN (k=5)
Perceptron
Logistic Reg.
LDA
.97
.85
.88
.88
.90
.40
.40
.40
.95
.93
.95
.95
.93
.90
.93
.90

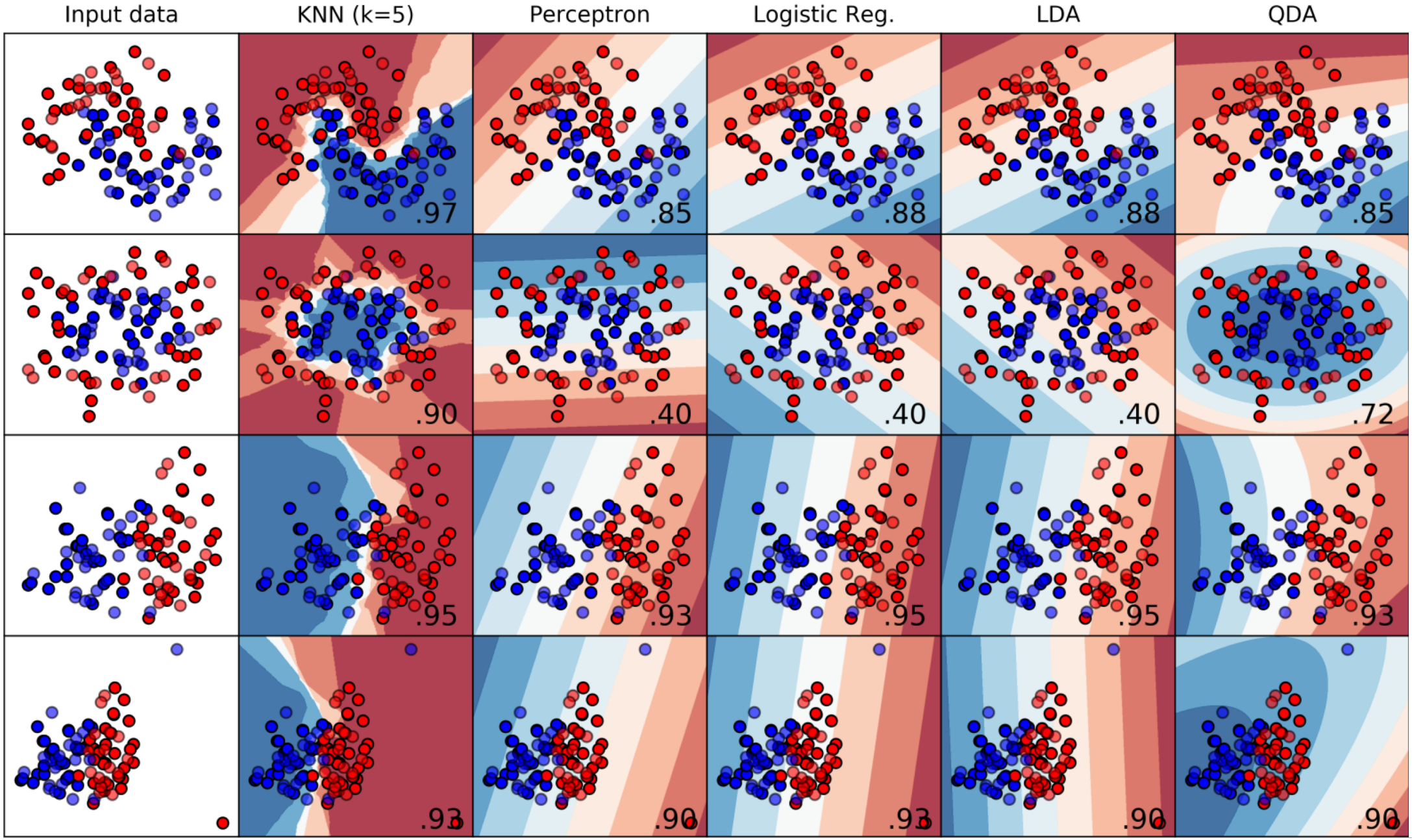
Quadratic Discriminant Analysis ($\Sigma_0 \neq \Sigma_1$)

We build a classifier that assigns the class with the higher posterior probability:

$$\frac{P(\boldsymbol{x}|y=1)}{P(\boldsymbol{x}|y=0)} = \frac{\frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_1|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)\right]}{\frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_0|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_0^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_0)\right]}$$

We assume a normal distribution, but different covariance matrices

Produces a quadric decision boundary


Input data
KNN (k=5)
Perceptron
Logistic Reg.
LDA
QDA
.97
.85
.88
.88
.85
.90
.40
.40
.40
.72
.95
.93
.95
.95
.93
.93
.90
.93
.90
.90

Naïve Bayes

Sometimes called "Idiot's Bayes"

Start with our original expression for our posterior [recall this was our discriminant function, $d_i(x)$]

$$P(y = i|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y = i)P(y = i)}{P(\boldsymbol{x})}$$

Write out the full expression with all the terms in $\boldsymbol{x}$

$$P(y = i|x_1, x_2, \dots, x_D) = \frac{P(x_1, x_2, \dots, x_D|y = i)P(y = i)}{P(x_1, x_2, \dots, x_D)}$$

Assumption: Given the class, the features are independent

$$P(y = i|x_1, x_2, \dots, x_D) = \frac{P(y = i)\prod_{j=1}^{D} P(x_j|y = i)}{P(x_1, x_2, \dots, x_D)}$$

Since the denominator is a proportionality constant

$$P(y = i|x_1, x_2, \dots, x_D) \propto P(y = i)\prod_{j=1}^{D} P(x_j|y = i)$$

Naïve Bayes

We assign the class that has the largest posterior, $P(y = i|x_1, x_2, \dots, x_D)$

$$P(y = i|x_1, x_2, \dots, x_D) \propto P(y = i) \prod_{j=1}^{D} P(x_j|y = i)$$

This implies we estimate the density of each feature **separately**

This independence assumption is a strong assumption that is rarely valid

Considerably simplifies computation and data needs

Is flexible to allow for different distributional forms (i.e. Gaussian) or nonparametric techniques

Naïve Bayes: Gaussian example

We assign the class that has the largest posterior, $P(y = i|x_1, x_2, \dots, x_D)$

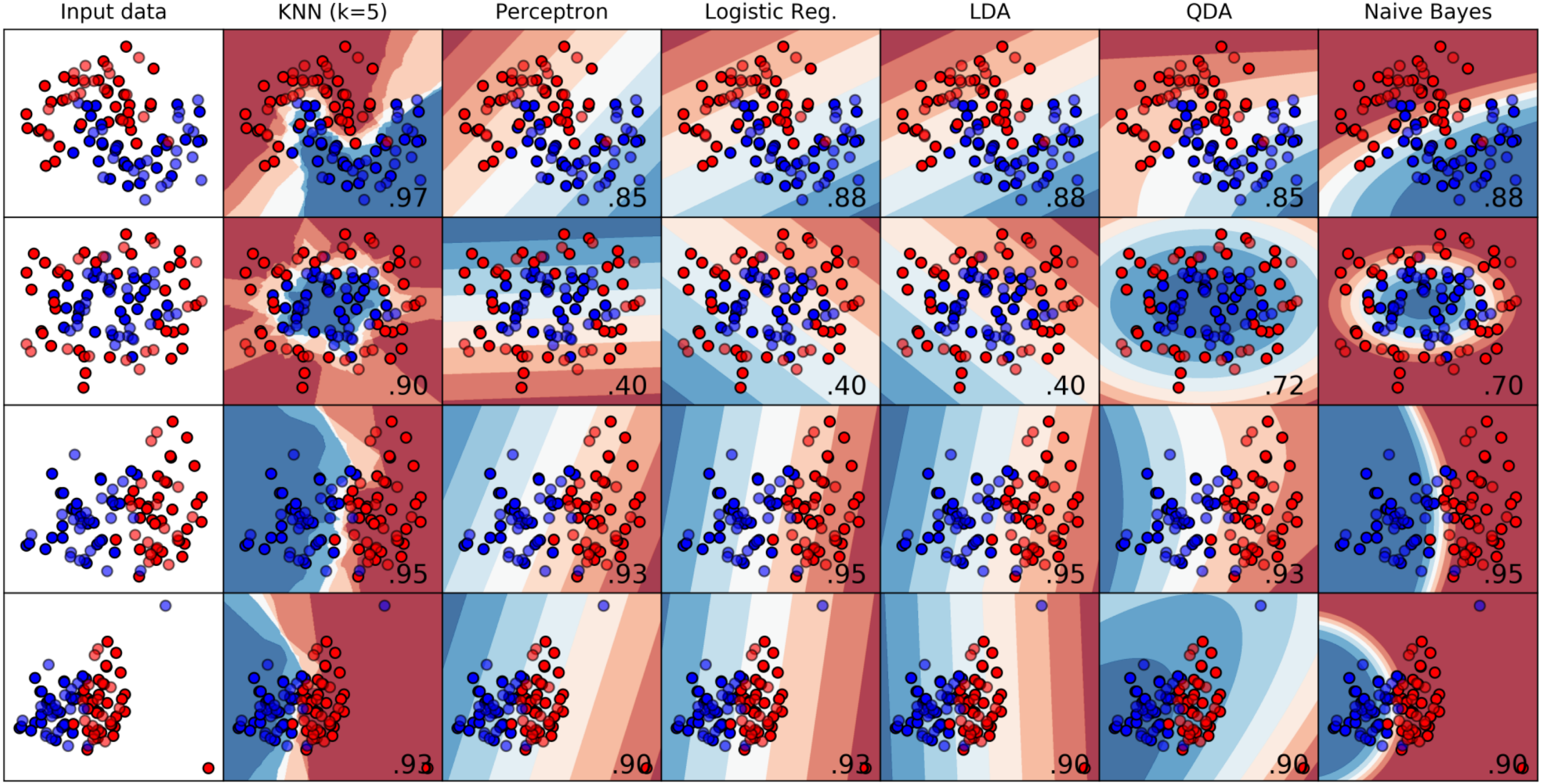
$$P(y = i|x_1, x_2, \dots, x_D) \propto P(y = i) \prod_{j=1}^{D} P\left(x_j|y = i\right)$$

This implies we estimate the density of each feature **separately**

If $P\left(x_j|y = i\right)$ is $N(\mu_{ji}, \sigma_{ji}^2)$, so for each class we estimate one mean and variance for each of the D features. We multiply **univariate** distributions together

$$P(y = i|x_1, x_2, \dots, x_D) \propto P(y = i) \prod_{j=1}^{D} N(\mu_{ji}, \sigma_{ji}^2)$$

If we didn't have the independence assumption we would have to estimate full covariance matrices for each of the c classes requiring cD^2 parameters instead of cD


Input data
KNN (k=5)
Perceptron
Logistic Reg.
LDA
QDA
Naive Bayes
.97
.85
.88
.88
.85
.88
.90
.40
.40
.40
.72
.70
.95
.93
.95
.95
.93
.95
.93
.90
.93
.90
.90
.90

Classifiers

Covered so far

- K-Nearest Neighbors
- Linear regression (can be used, but not recommended)
- Perceptron
- Logistic Regression
- Fisher's Linear Discriminant / Linear Discriminant Analysis
- Quadratic Discriminant Analysis
- Naïve Bayes

Rely on a linear combination of weights and features: $\boldsymbol{w}^T \boldsymbol{x}$

- Decision Trees
- Ensemble methods (random forests, gradient boosting)

Appendix (Derivations)

FLD: Fisher Criterion Maximization

$$J(\boldsymbol{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

$$= \frac{\boldsymbol{w}^T(\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T\boldsymbol{w}}{\sum_{i \in C_1}(y_n - m_k)^2 + \sum_{i \in C_2}(y_n - m_k)^2}$$

$$m_2 - m_1 = \boldsymbol{w}^T(\boldsymbol{m}_2 - \boldsymbol{m}_1)$$

$$s_k^2 = \sum_{i \in C_k}(y_n - m_k)^2$$

$$= \frac{\boldsymbol{w}^T(\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T\boldsymbol{w}}{\sum_{i \in C_1}(\boldsymbol{w}^T\boldsymbol{x}_i - \boldsymbol{w}^T\boldsymbol{m}_1)^2 + \sum_{i \in C_2}(\boldsymbol{w}^T\boldsymbol{x}_i - \boldsymbol{w}^T\boldsymbol{m}_2)^2}$$

$$m_k = \boldsymbol{w}^T\boldsymbol{m}_k$$

$$y_k = \boldsymbol{w}^T\boldsymbol{x}_k$$

$$= \frac{\boldsymbol{w}^T(\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T\boldsymbol{w}}{\boldsymbol{w}^T\left[\sum_{i \in C_1}(\boldsymbol{x}_i - \boldsymbol{m}_1)(\boldsymbol{x}_i - \boldsymbol{m}_1)^T + \sum_{i \in C_2}(\boldsymbol{x}_i - \boldsymbol{m}_2)(\boldsymbol{x}_i - \boldsymbol{m}_2)^T\right]\boldsymbol{w}}$$

Factoring out the $\boldsymbol{w}$ in denominator

FLD: Fisher Criterion Maximization

$$\boldsymbol{S}_B = (\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T$$

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T(\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T\boldsymbol{w}}{\boldsymbol{w}^T\left[\sum_{i\in C_1}(\boldsymbol{x}_i - \boldsymbol{m}_1)(\boldsymbol{x}_i - \boldsymbol{m}_1)^T + \sum_{i\in C_2}(\boldsymbol{x}_i - \boldsymbol{m}_2)(\boldsymbol{x}_i - \boldsymbol{m}_2)^T\right]\boldsymbol{w}}$$

$$\boldsymbol{S}_W = \sum_{i\in C_1}(\boldsymbol{x}_i - \boldsymbol{m}_1)(\boldsymbol{x}_i - \boldsymbol{m}_1)^T + \sum_{i\in C_2}(\boldsymbol{x}_i - \boldsymbol{m}_2)(\boldsymbol{x}_i - \boldsymbol{m}_2)^T$$

$$= \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2$$

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w}}$$

Generalized Raleigh Quotient

We want to maximize this and solve for $\boldsymbol{w}$

FLD: Fisher Criterion Maximization

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_W \boldsymbol{w}}$$

Take the derivative (gradient), set it equal to zero, solve for $\boldsymbol{w}$

Recall the quotient rule for differentiation:

$$f(x) = \frac{u(x)}{v(x)} \qquad \frac{df}{dx} = \frac{u'v - uv'}{v^2}$$

Matrix derivatives of the form $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}$ with respect to $\boldsymbol{x}$ are:

$$\frac{d\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}}{d\boldsymbol{x}} = \boldsymbol{x}^T (\boldsymbol{A} + \boldsymbol{A}^T)$$

If A is symmetric (as it is for our scatter matrices), then $\boldsymbol{A} = \boldsymbol{A}^T$, therefore:

$$\boldsymbol{x}^T (\boldsymbol{A} + \boldsymbol{A}^T) = 2\boldsymbol{x}^T \boldsymbol{A}$$

Therefore, we can write:

$$\frac{dJ(\boldsymbol{w})}{d\boldsymbol{w}} = \frac{(2\boldsymbol{w}^T \boldsymbol{S}_B)(\boldsymbol{w}^T \boldsymbol{S}_W \boldsymbol{w}) - (\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w})(2\boldsymbol{w}^T \boldsymbol{S}_W)}{(\boldsymbol{w}^T \boldsymbol{S}_W \boldsymbol{w})^2} = 0$$

We want to solve this for $\boldsymbol{w}$

FLD: Fisher Criterion Maximization

$$\frac{dJ(\boldsymbol{w})}{d\boldsymbol{w}} = \frac{(2\boldsymbol{w}^T\boldsymbol{S}_B)(\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w}) - (\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w})(2\boldsymbol{w}^T\boldsymbol{S}_W)}{(\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w})^2} = 0$$

We want to solve this for $\boldsymbol{w}$

Since the denominator will not approach infinity, only the numerator matters

$$(2\boldsymbol{w}^T\boldsymbol{S}_B)(\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w}) - (\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w})(2\boldsymbol{w}^T\boldsymbol{S}_W) = 0$$

$$(\underbrace{\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w}}_{\alpha})(\boldsymbol{w}^T\boldsymbol{S}_B) = (\underbrace{\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w}}_{\beta})(\boldsymbol{w}^T\boldsymbol{S}_W)$$

$[1\times D][D\times D][D\times 1] \rightarrow$ scalar

These will only affect magnitude. We assume that $\boldsymbol{w}$ is of unit length, so we replace these with variables α and β.

$$\alpha\boldsymbol{w}^T\boldsymbol{S}_B = \beta\boldsymbol{w}^T\boldsymbol{S}_W$$

FLD: Fisher Criterion Maximization

$$\alpha \boldsymbol{w}^T \boldsymbol{S}_B = \beta \boldsymbol{w}^T \boldsymbol{S}_W$$

$$\alpha \boldsymbol{S}_B^T \boldsymbol{w} = \beta \boldsymbol{S}_W^T \boldsymbol{w}$$

$$\alpha \boldsymbol{S}_B \boldsymbol{w} = \beta \boldsymbol{S}_W \boldsymbol{w}$$

$$\alpha(\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T \boldsymbol{w} = \beta \boldsymbol{S}_W \boldsymbol{w}$$

scalar $\boldsymbol{m}_2 - \boldsymbol{m}_1$, call this γ

$$\alpha\gamma(\boldsymbol{m}_2 - \boldsymbol{m}_1) = \beta \boldsymbol{S}_W \boldsymbol{w}$$

Property of matrix transposition:

$$(\boldsymbol{AB})^T = \boldsymbol{B}^T \boldsymbol{A}^T$$

The scatter matrices are symmetric:

$$\boldsymbol{S}_B = \boldsymbol{S}_B^T \qquad \boldsymbol{S}_W = \boldsymbol{S}_W^T$$

Between-class scatter matrix:

$$\boldsymbol{S}_B = (\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T$$

Aside: dimensionality reduction

Rearranging, this is an eigenvalue problem

$$\boldsymbol{S}_W^{-1} \boldsymbol{S}_B \boldsymbol{w} = \lambda \boldsymbol{w}$$

For multiclass problems, we can use the eigenvalue of $\boldsymbol{S}_W^{-1} \boldsymbol{S}_B$, much like PCA to get projections into lower dimensional subspaces where the classes are well-separated

FLD: Fisher Criterion Maximization

$$\alpha\gamma(\boldsymbol{m}_2 - \boldsymbol{m}_1) = \beta \boldsymbol{S}_W \boldsymbol{w}$$

Solving for $\boldsymbol{w}$:

$$\boldsymbol{w} = \frac{\alpha\gamma}{\beta} \boldsymbol{S}_W^{-1}(\boldsymbol{m}_2 - \boldsymbol{m}_1)$$

We only care about the direction of $\boldsymbol{w}$

$$\boldsymbol{w} \propto \boldsymbol{S}_W^{-1}(\boldsymbol{m}_2 - \boldsymbol{m}_1)$$

Note: if $\boldsymbol{S}_w$ is isotropic (proportional to the identity matrix, i.e. if $\boldsymbol{S}_w = a\boldsymbol{I}$), then this is just the difference between the means

$$\boldsymbol{w} \propto (\Sigma_1 + \Sigma_2)^{-1} (\boldsymbol{m}_2 - \boldsymbol{m}_1)$$

Linear Discriminant Analysis ($\Sigma_0 = \Sigma_1$)

We build a classifier that assigns the class with the higher posterior probability:

$$\frac{P(\boldsymbol{x}|y=1)}{P(\boldsymbol{x}|y=0)} = \frac{\frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)\right]}{\frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_0)\right]}$$

$$= \frac{\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)\right]}{\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_0)\right]}$$

$$\ln\left|\frac{P(\boldsymbol{x}|y=1)}{P(\boldsymbol{x}|y=0)}\right| = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_0)$$

Linear Discriminant Analysis ($\Sigma_0 = \Sigma_1$)

$$\ln\left|\frac{P(\boldsymbol{x}|y=1)}{P(\boldsymbol{x}|y=0)}\right| = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_0)$$

Expanding this expression yields:

These combine since $\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i = \boldsymbol{\mu}_i^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}$ for symmetric matrices

$$= -\frac{1}{2}[\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} + \boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1$$
$$-\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 + \boldsymbol{\mu}_0^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \boldsymbol{\mu}_0^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0$$

$$= \frac{1}{2}[2\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0) - (\boldsymbol{\mu}_1+\boldsymbol{\mu}_0)\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)]$$

$$= \boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0) - \frac{1}{2}(\boldsymbol{\mu}_1+\boldsymbol{\mu}_0)\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)$$

Linear Discriminant Analysis ($\Sigma_0 = \Sigma_1$)

$$\ln\left|\frac{P(\boldsymbol{x}|y=1)}{P(\boldsymbol{x}|y=0)}\right| = \boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0)\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$$

Since our decision rule is to classify as class 1 if the following is true:

$$\ln\left|\frac{P(\boldsymbol{x}|y=1)}{P(\boldsymbol{x}|y=0)}\right| > \ln\left|\frac{P(y=0)}{P(y=1)}\right|$$

We can rewrite our decision rule as:

$$\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) > \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0)\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) + \ln\left|\frac{P(y=0)}{P(y=1)}\right|$$

Or simply as:

$$\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) > C$$

If we define $\boldsymbol{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$, then this becomes $\boldsymbol{x}^T\boldsymbol{w} > C$