

Classification

Lecture 10

Classifiers

Covered so far

K-Nearest Neighbors

Perceptron

Logistic Regression

Fisher's Linear Discriminant

Linear Discriminant Analysis

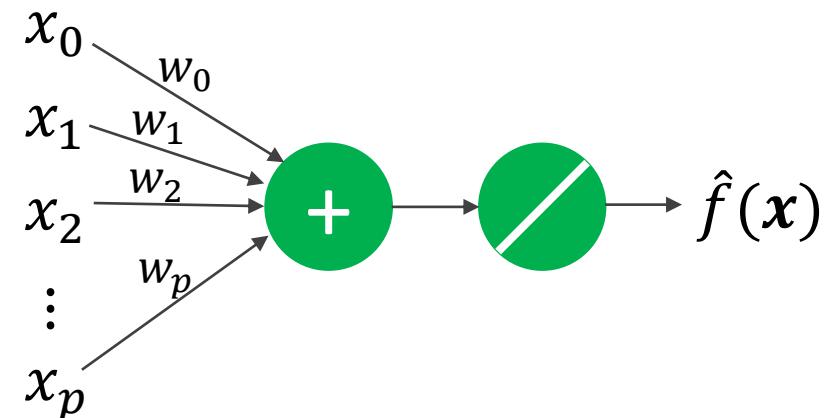
Quadratic Discriminant Analysis

Naïve Bayes

Remember linear models?

Linear Regression

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

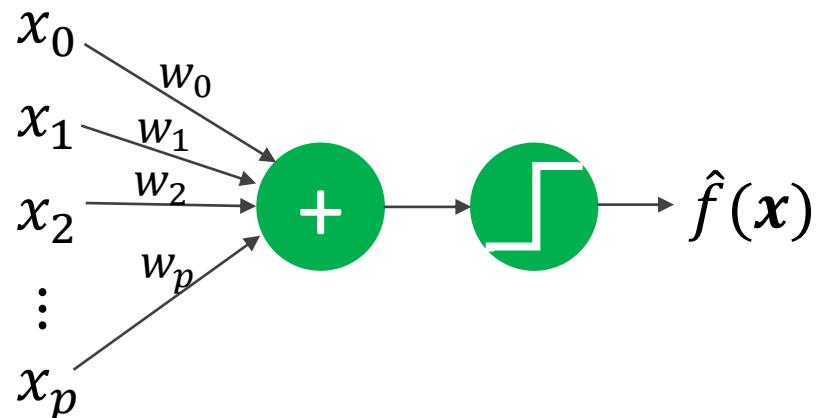


Linear Classification

Perceptron

$$\hat{f}(\mathbf{x}) = \text{sign} \left(\sum_{i=0}^p w_i x_i \right)$$

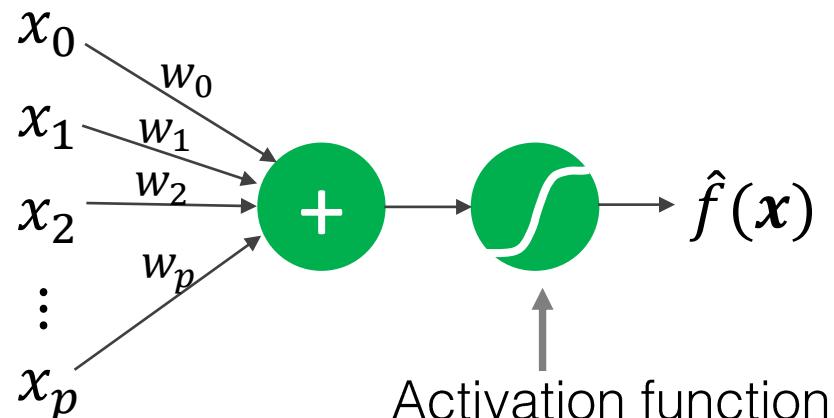
$$\text{sign}(x) = \begin{cases} 1 & x > 0 \\ -1 & \text{else} \end{cases}$$



Logistic Regression

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

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



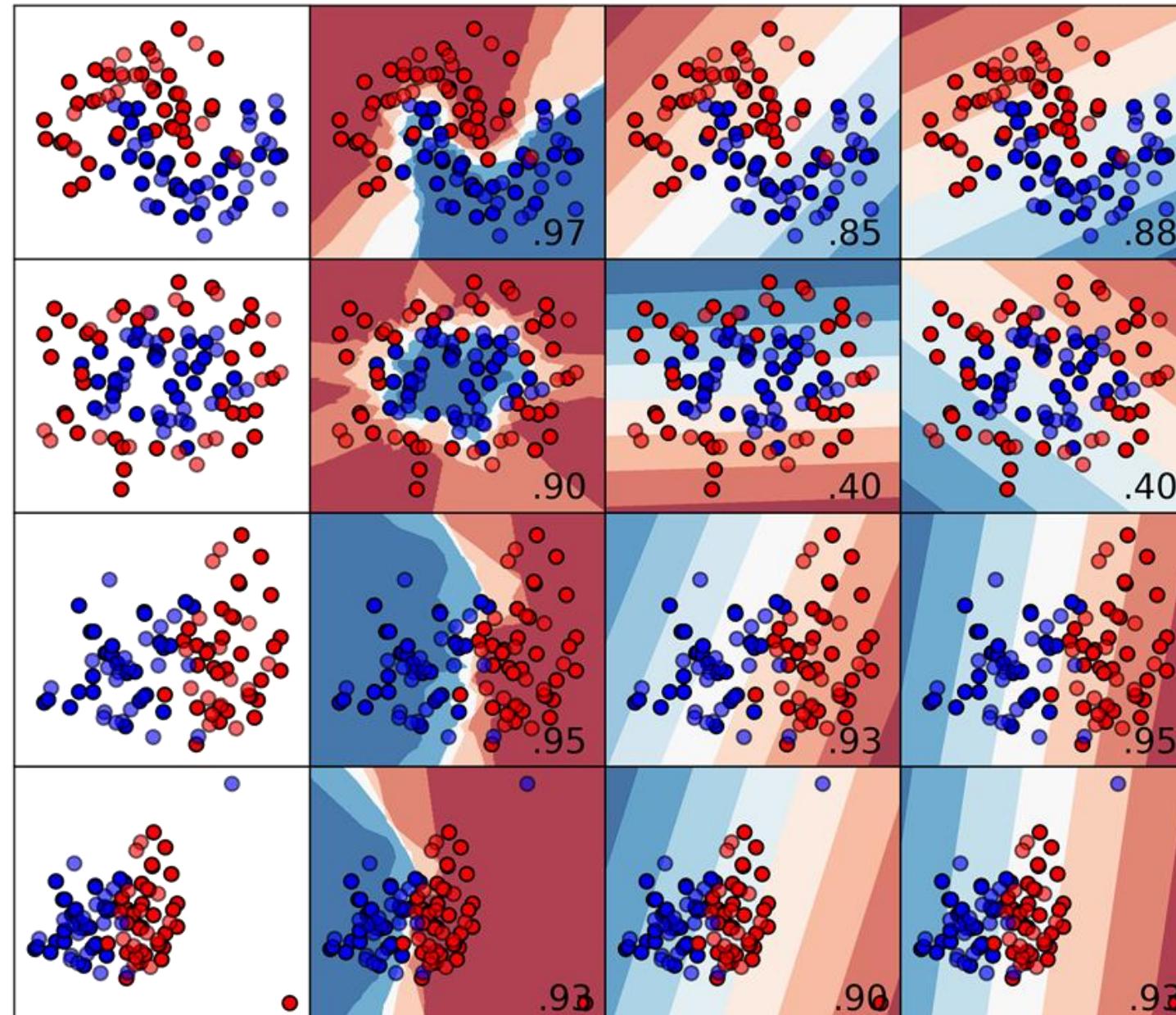
Source: Abu-Mostafa, Learning from Data, Caltech

Input data

KNN (k=5)

Perceptron

Logistic Reg.



Comparison of classifiers
we've seen so far

Projections

$$\mathbf{u}^T \mathbf{z} = [u_1 \quad u_2] \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

$$= u_1 z_1 + u_2 z_2$$

$$= (-0.5)(0.7) + (0.87)(2)$$

$$= 1.39$$

Length (magnitude) of the projection of \mathbf{z} onto \mathbf{u}

This is valid because \mathbf{u} is a unit vector (length is 1: $\|\mathbf{u}\|_2 = \sqrt{u_1^2 + u_2^2} = \sqrt{(-0.5)^2 + (0.87)^2} \cong 1$)

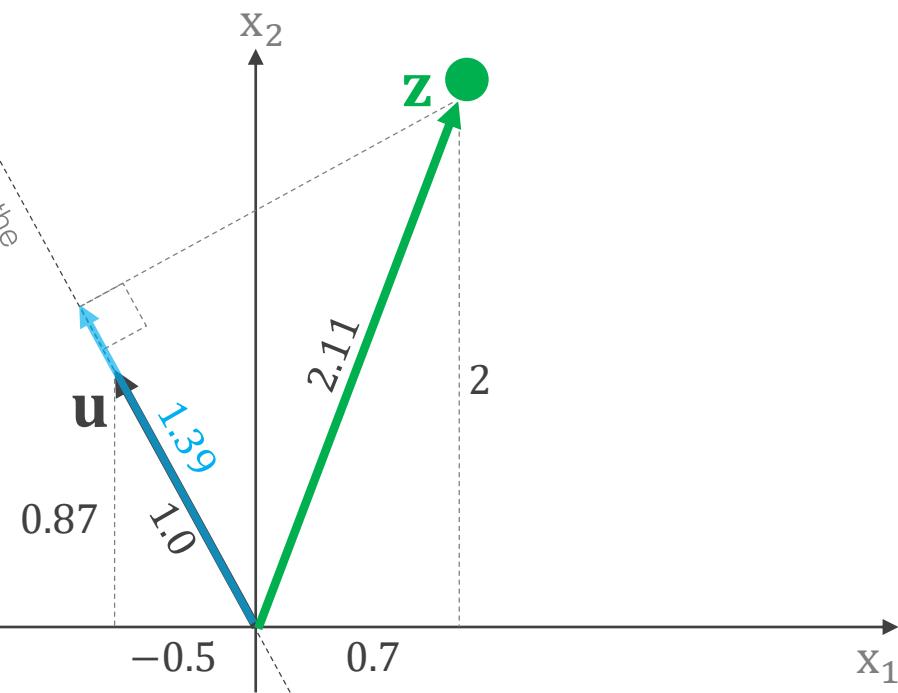
Notes on projections:

If \mathbf{u} was NOT a unit vector, the magnitude (length) of the projection of \mathbf{z} onto \mathbf{u} would be calculated by normalizing the result by the length of \mathbf{u} :

$$\frac{\mathbf{u}^T \mathbf{z}}{\|\mathbf{u}\|_2}$$

In our case above,
 $\mathbf{u}^T \mathbf{u} = 1$

This process projects the data onto the line defined by the direction of \mathbf{u}

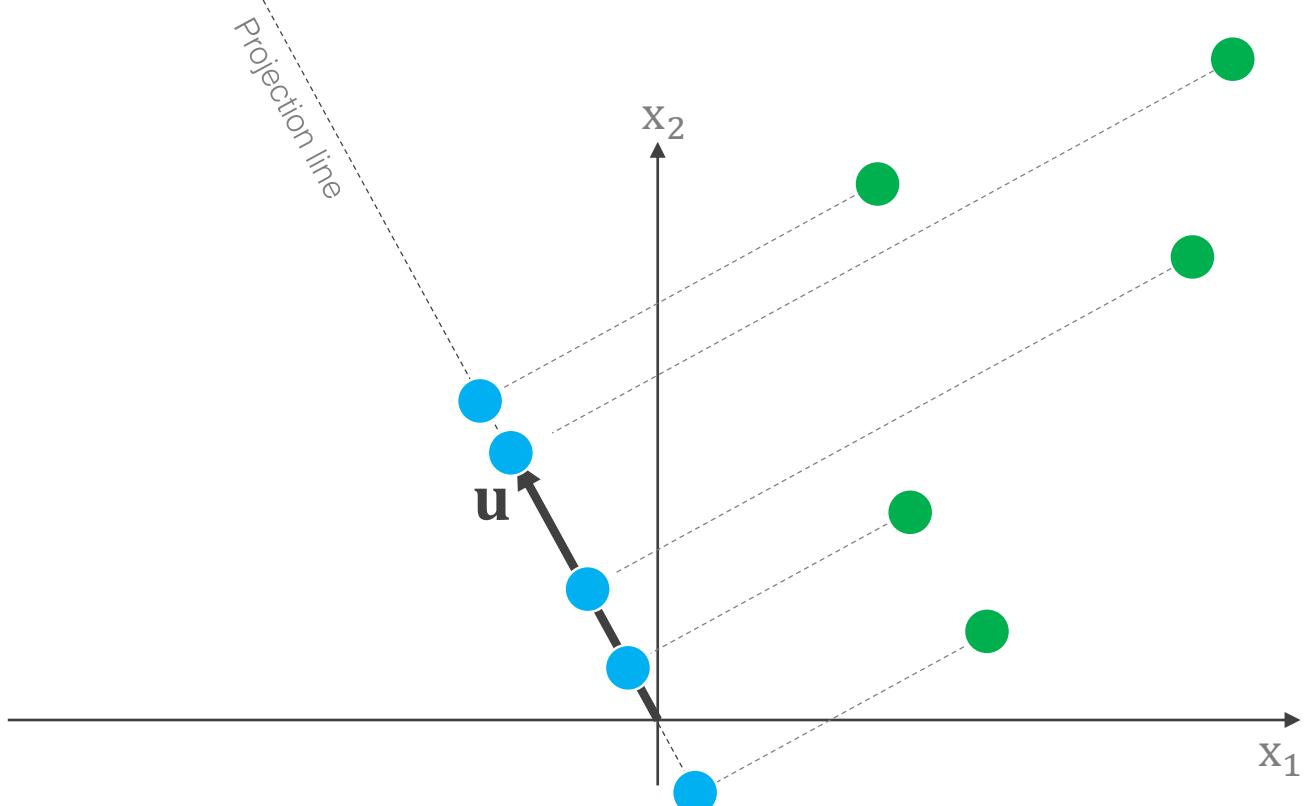


The vector projection of \mathbf{z} onto \mathbf{u} would multiply the length by the direction of \mathbf{u} :

$$\text{proj}_{\mathbf{u}}(\mathbf{z}) = \left(\frac{\mathbf{u}^T \mathbf{z}}{\mathbf{u}^T \mathbf{u}} \right) \frac{\mathbf{u}}{\|\mathbf{u}\|_2}$$

Projections

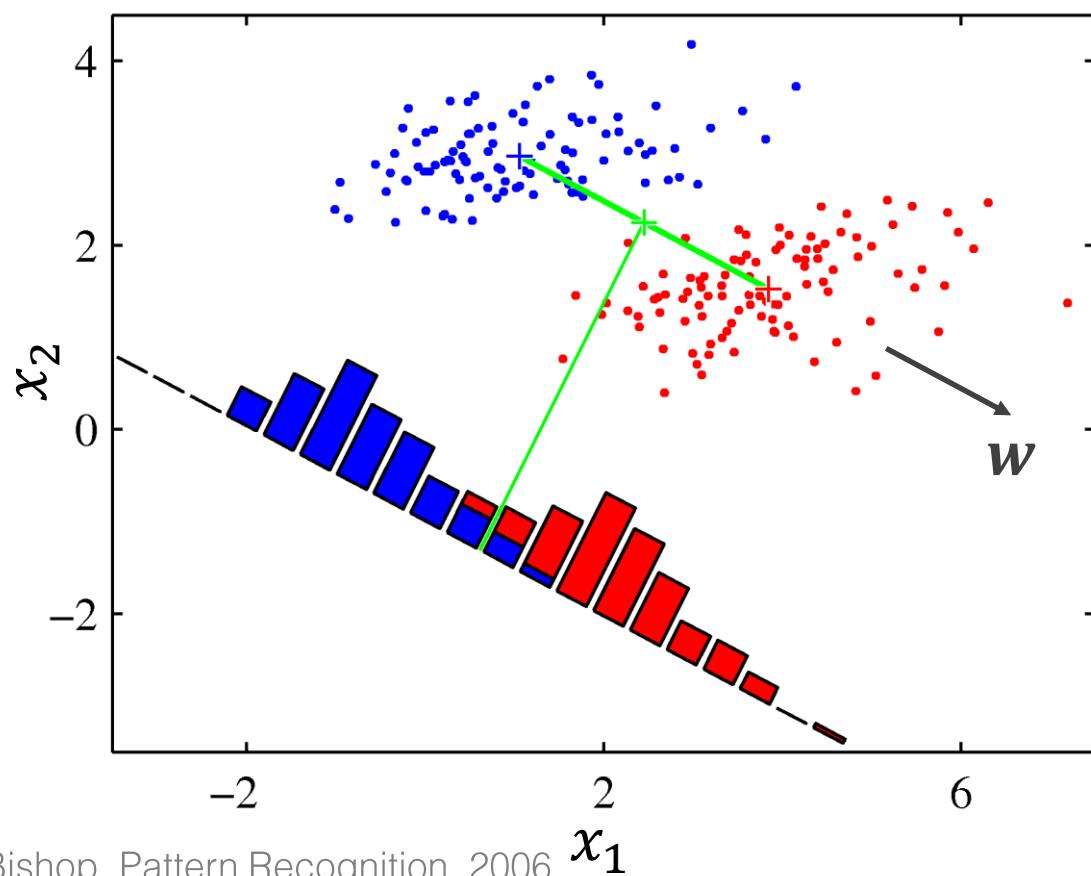
We could project any points in this space onto the line defined by the direction of unit vector \mathbf{u}



Fisher's Linear Discriminant

Looks for the projection into the one dimension that “best” separates the classes

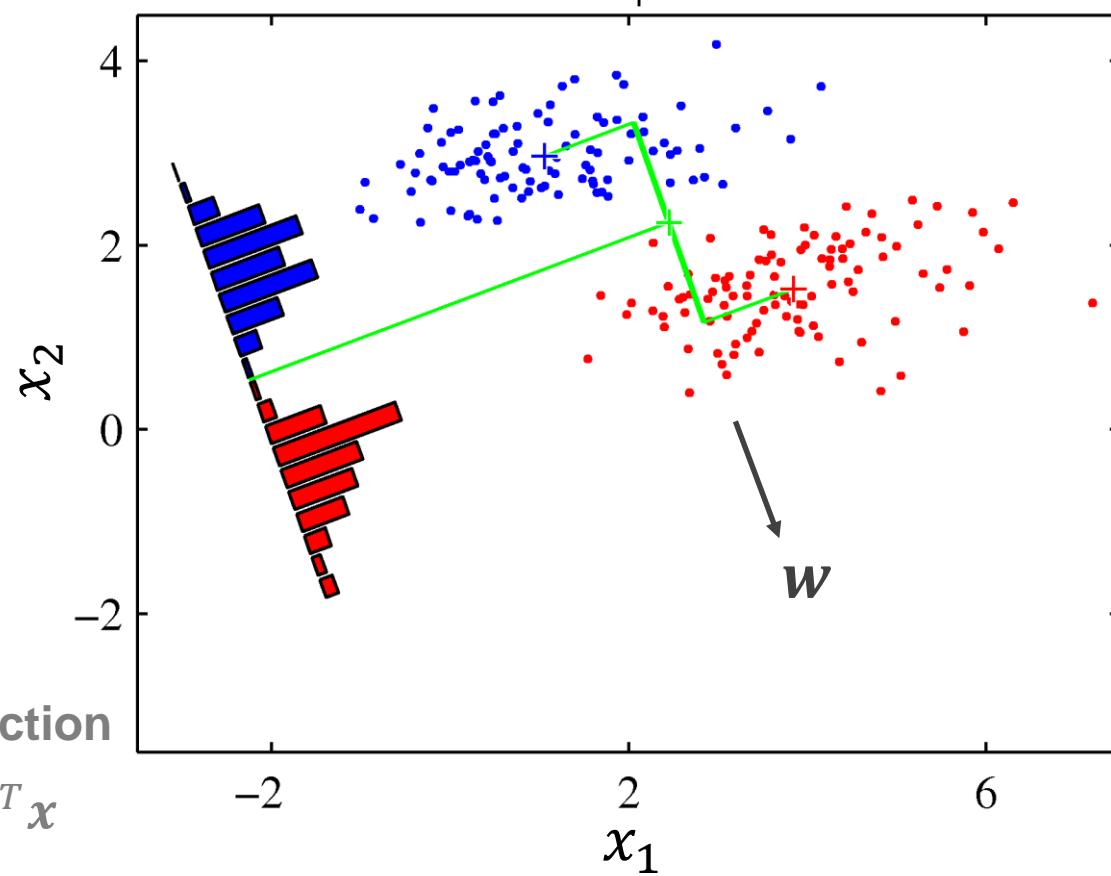
Projection onto line connecting the means



Linear projection

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

Projection onto a line providing improved class separation



Fisher's Linear Discriminant (FLD)

- 1 Finds a projection into a lower dimension that “best” separates the classes

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

Consider \mathbf{w} is a unit vector of parameters

- 2 We then classify the data in this space

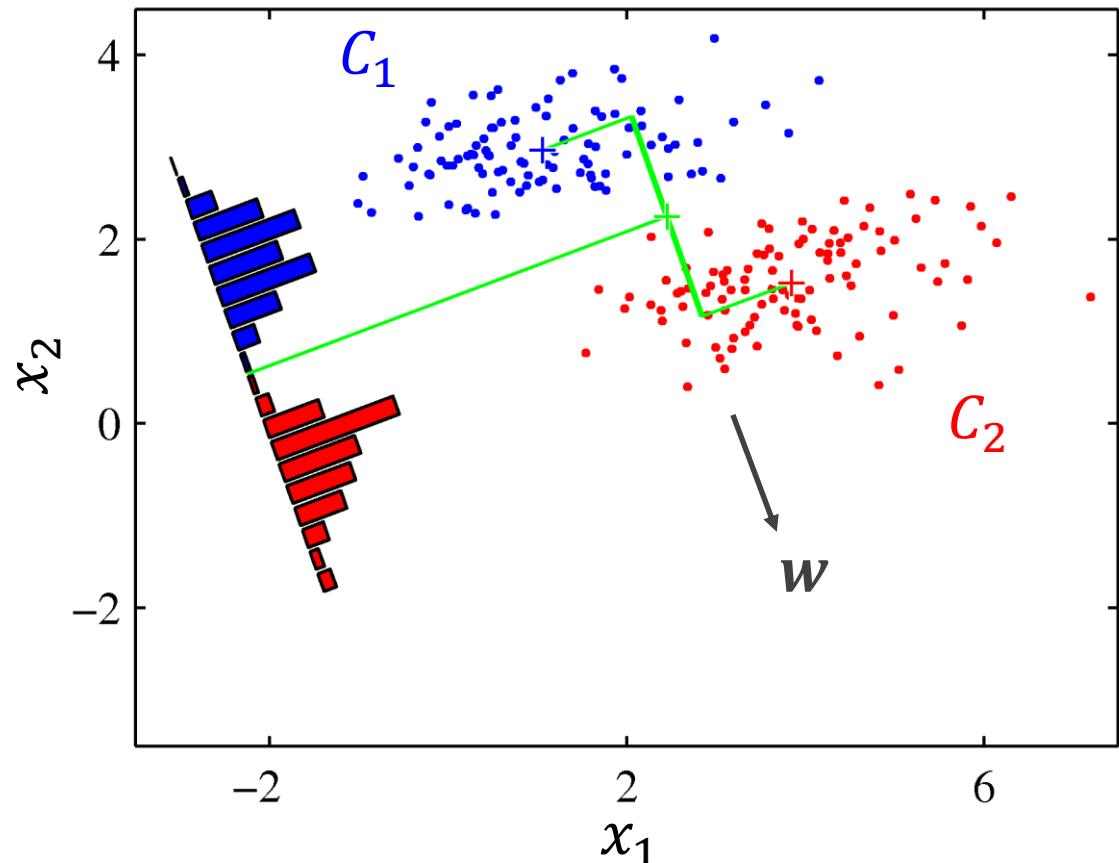
Similar to PCA, but accounts for class separability

Our decision rule becomes:

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

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

FLD: how do we choose the vector w ?



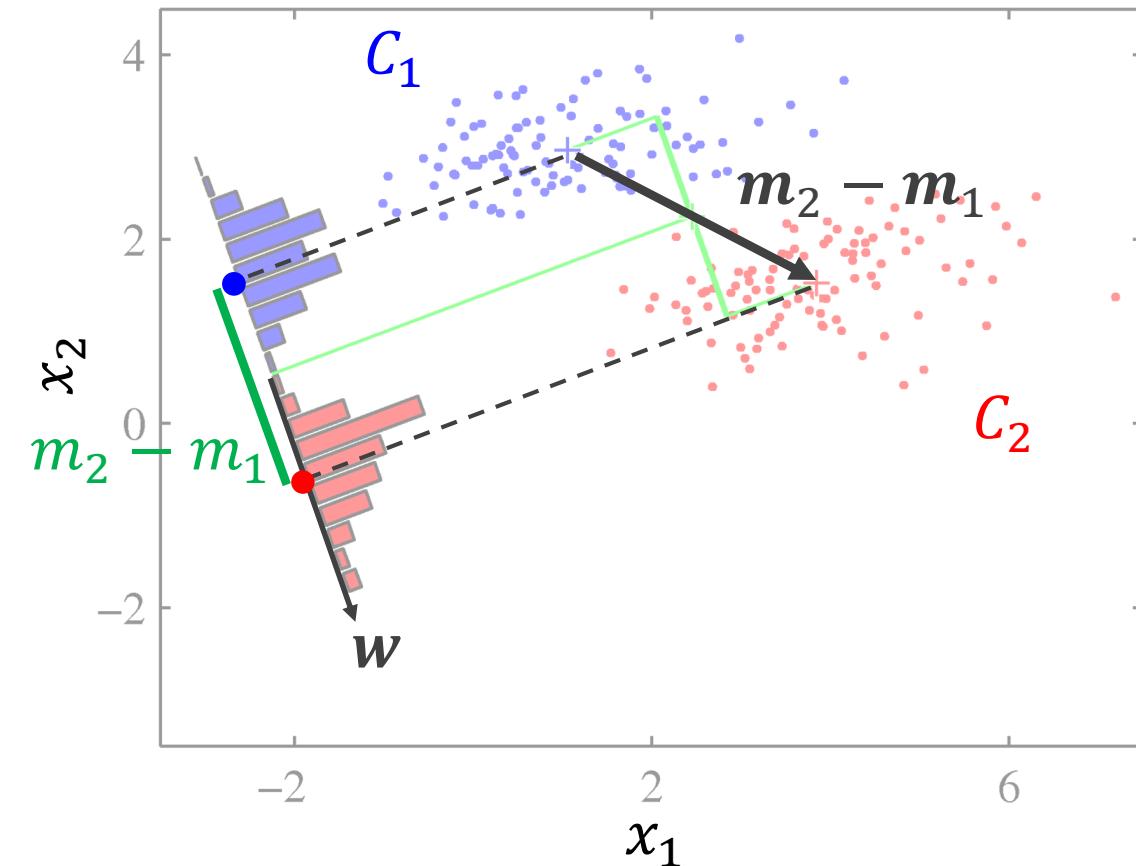
Increase the distance between the **means**

Decrease the **variance** within each class

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

FLD: how do we choose the vector w ?

Increase the distance between the **means**



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

$$\mathbf{m}_1 = \frac{1}{N_1} \sum_{i \in C_1} \mathbf{x}_i$$

mean of class 1

$$\mathbf{m}_2 = \frac{1}{N_2} \sum_{i \in C_2} \mathbf{x}_i$$

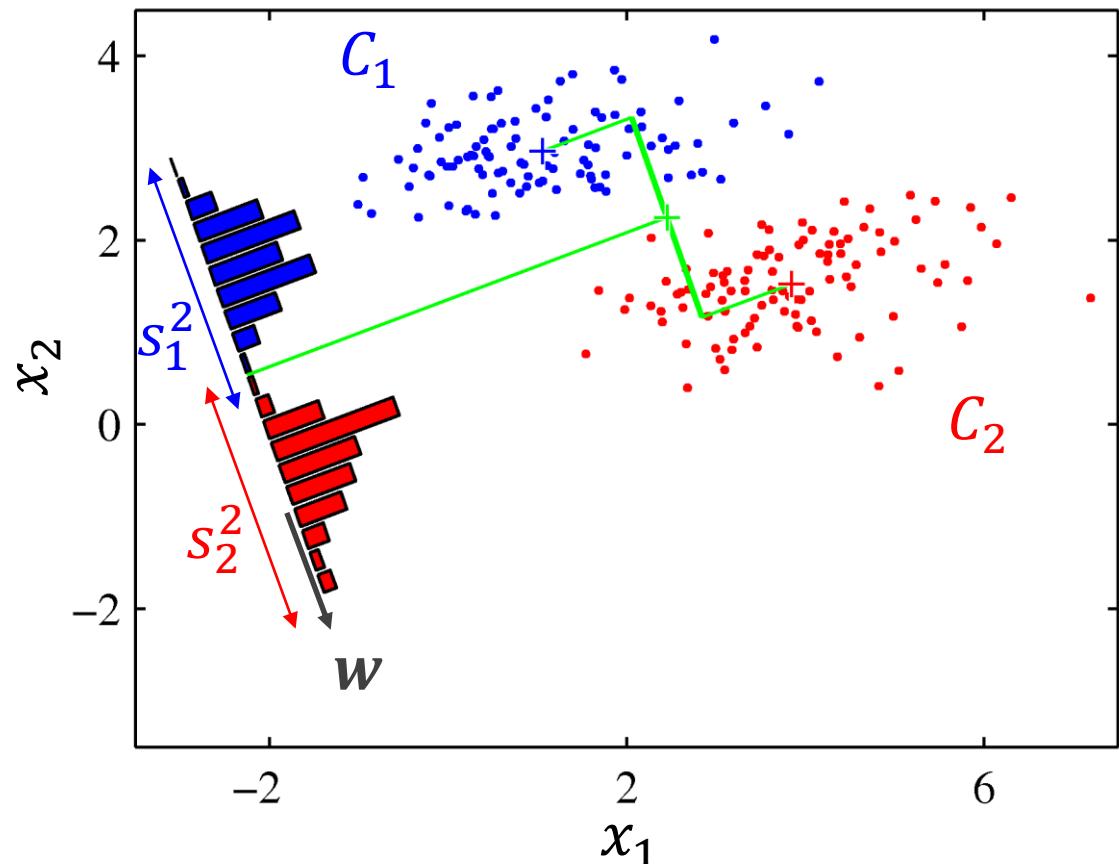
mean of class 2

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

The distance between the means:

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

FLD: how do we choose the vector w ?



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

Decrease the **variance** within each class

The “scatter” of the **projected** data:

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

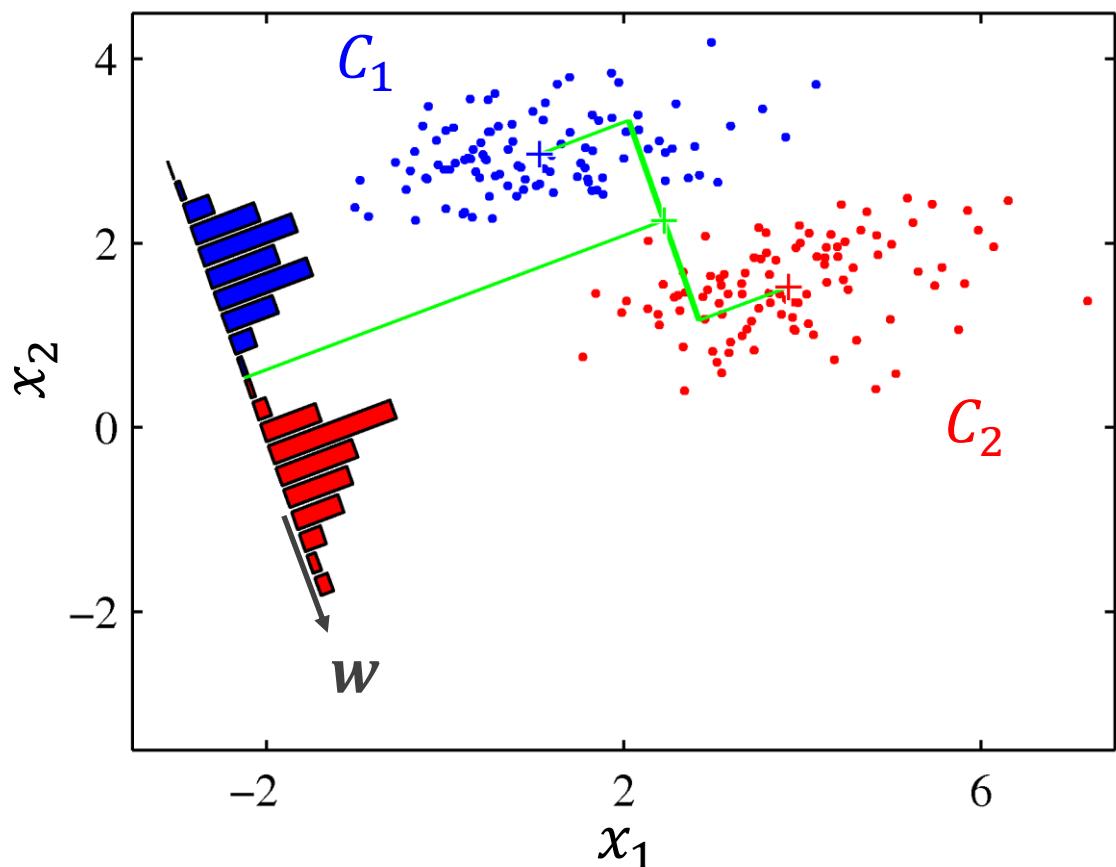
where $m_k = w^T m_k$

$$y_i = w^T x_i$$

Therefore the total within-class scatter:

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

FLD: how do we choose the vector w ?



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

Increase the distance between the **means**

$$m_2 - m_1 = w^T(m_2 - m_1)$$

Decrease the **variance** within each class

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

The Fisher criterion is then:

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

We want to maximize this and solve for w

FLD: how do we choose the vector w ?

We want to maximize this and solve for w

$$J(w) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$
$$= \frac{w^T S_B w}{w^T S_W w} \quad (\text{see appendix slides for full derivation})$$

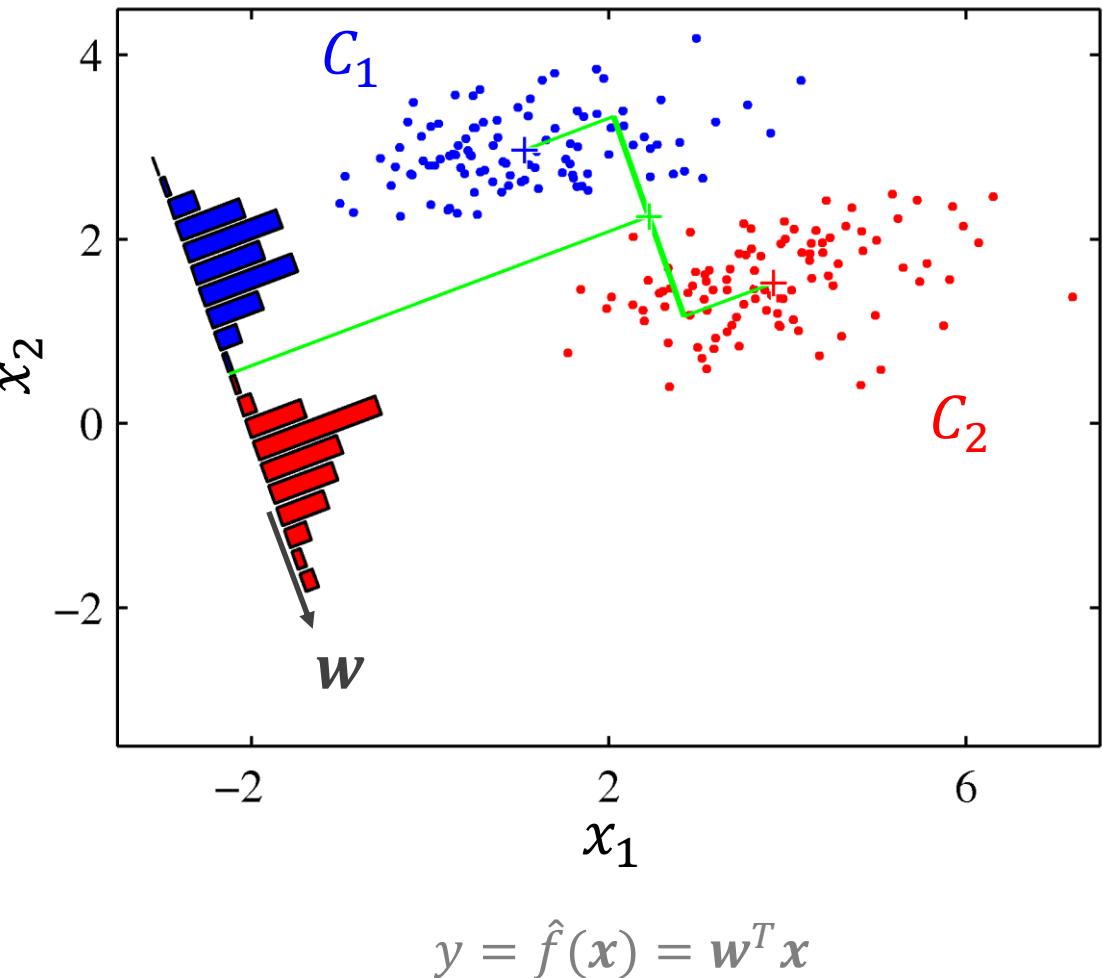
Take the derivative (gradient), set it equal to zero, solve for w

(see appendix slides for full derivation)

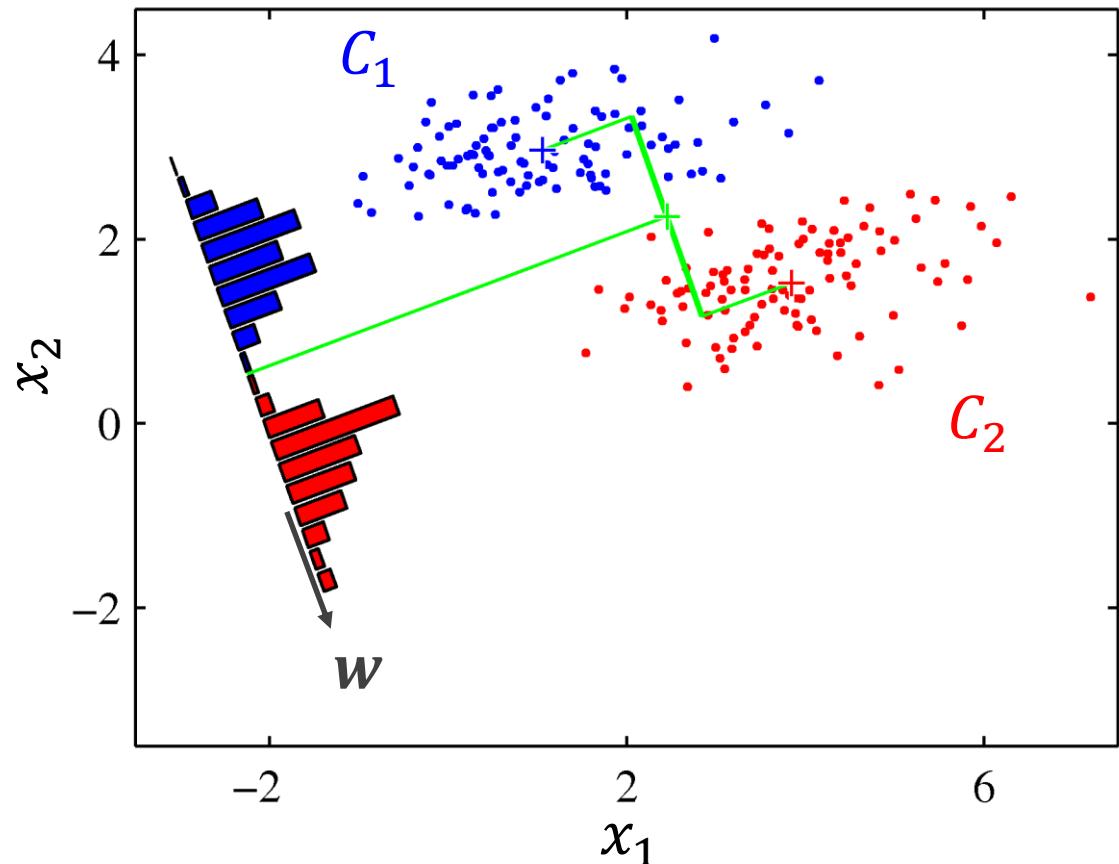
$$w \propto S_W^{-1} (m_2 - m_1)$$

$$w \propto (\Sigma_1 + \Sigma_2)^{-1} (m_2 - m_1)$$

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



Fisher's Linear Discriminant



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

No assumptions about the distribution of the data and allows for different covariance matrices for each class

Only applicable for 2 classes

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

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

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

Bayes rule in the context of classification



Class 1: Light Post

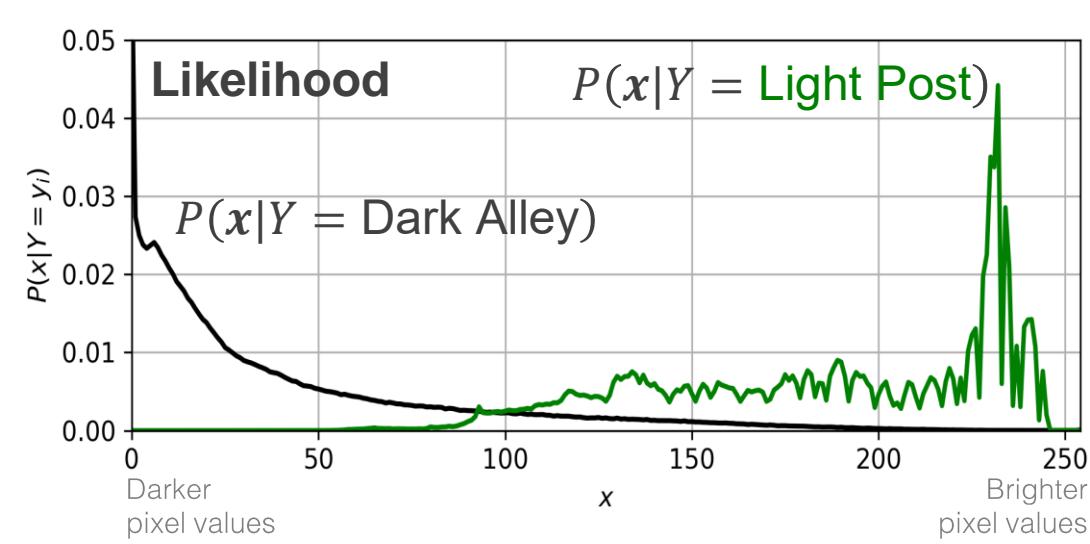
Randomly draw a pixel from either of the images: $x_i = 149$



Class 0: Dark Alley

Darker pixel values are lower numbers (closer to 0), brighter pixels are higher numbers (closer to 255)

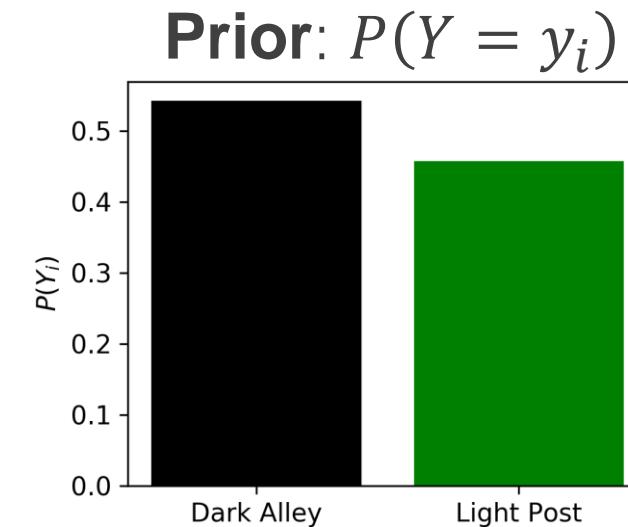
How do we determine which image it was most likely to have come from?



Class 1: Light Post y_1

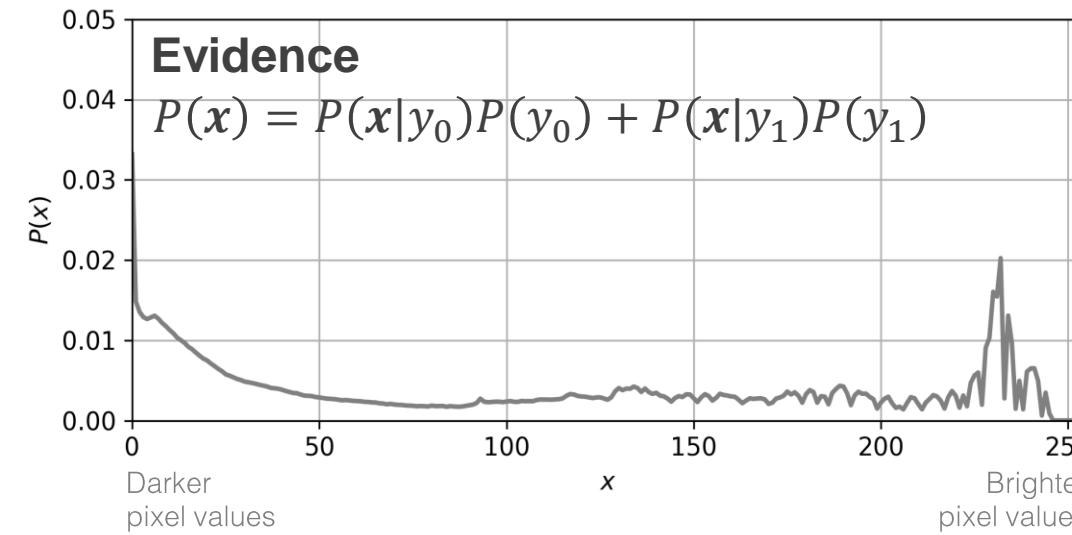
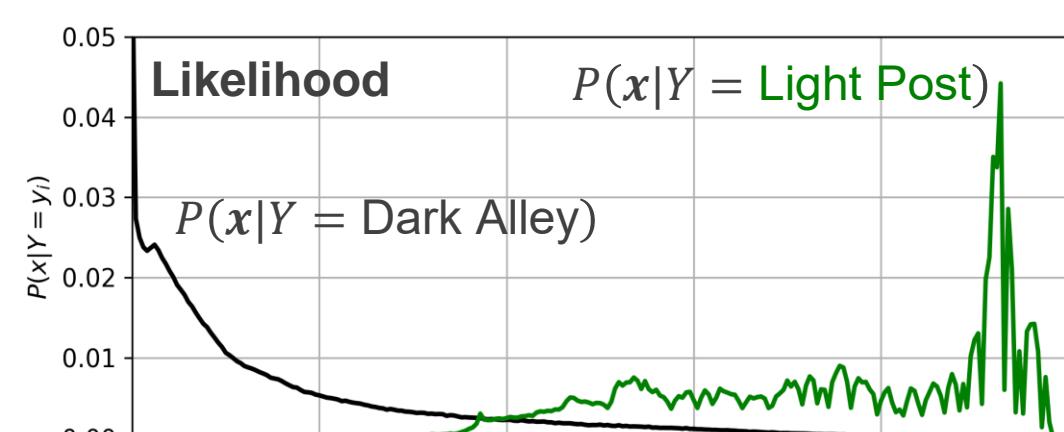


Class 0: Dark Alley y_0



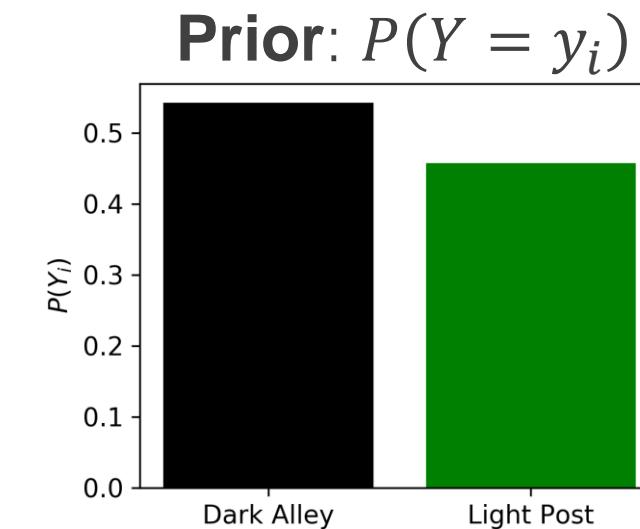
Bayes' Rule

$$\text{Posterior } P(Y = y_i | x) = \frac{\text{Likelihood} \quad \text{Prior}}{\text{Evidence}} \quad P(x|Y = y_i)P(Y = y_i)$$



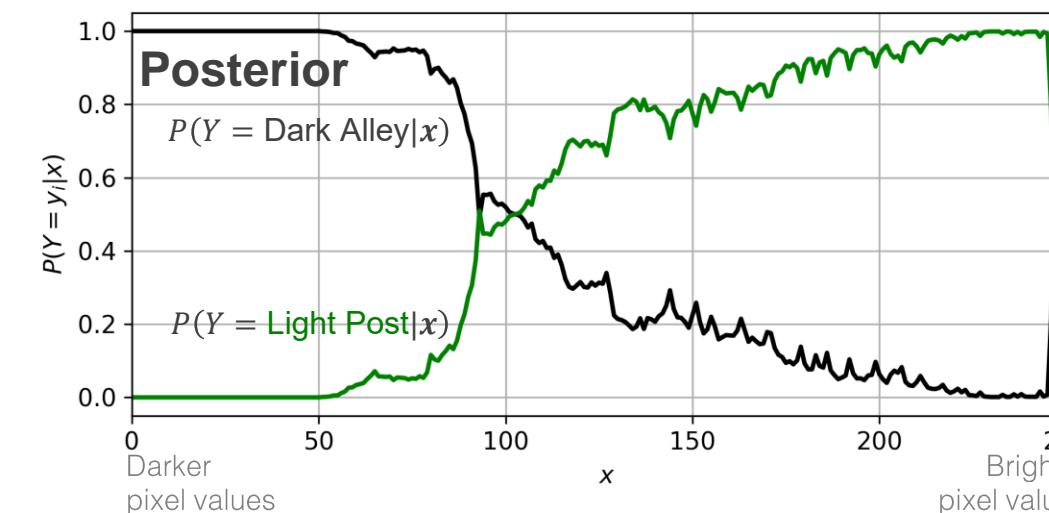
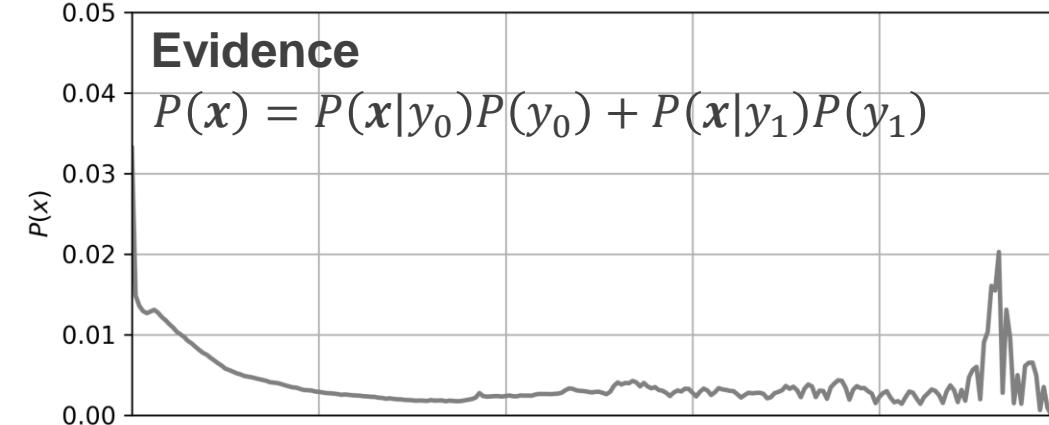
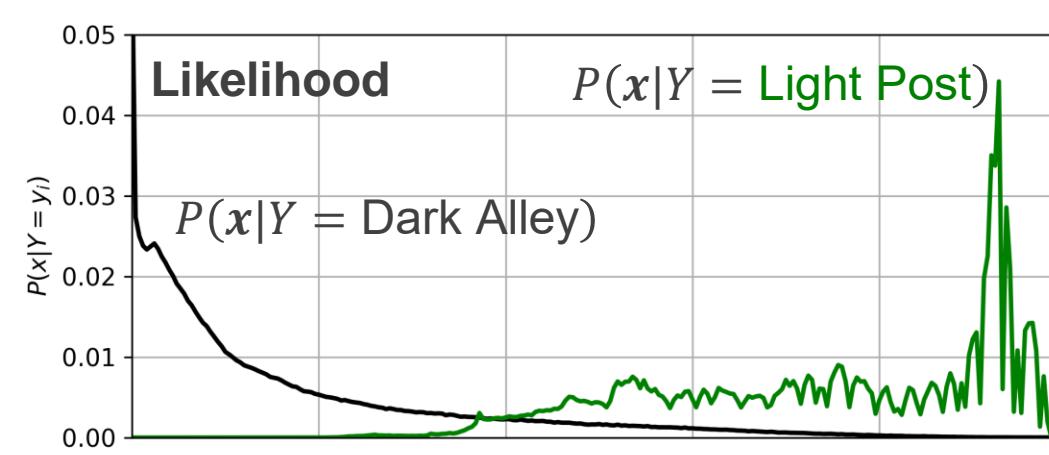
Class 1: Light Post y_1

Class 0: Dark Alley y_0



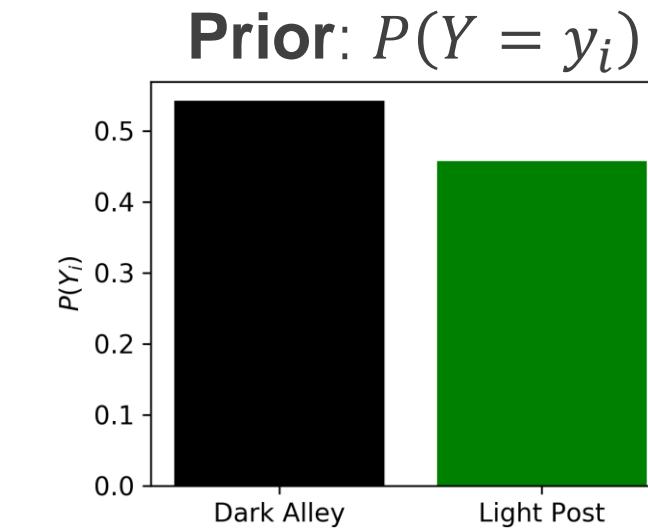
Bayes' Rule

$$\text{Posterior } P(Y = y_i|x) = \frac{\text{Likelihood} \quad \text{Prior}}{\text{Evidence}} \quad P(x|Y = y_i)P(Y = y_i)$$



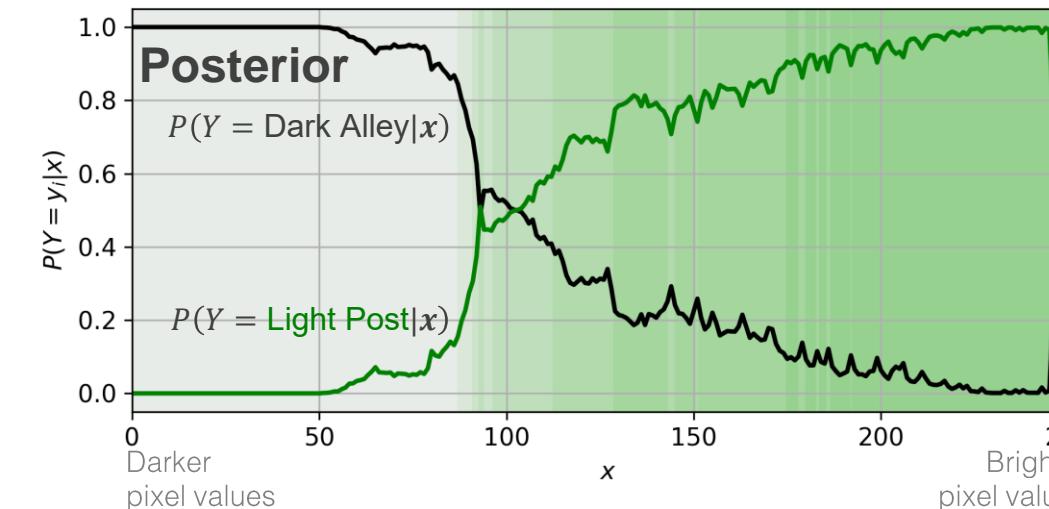
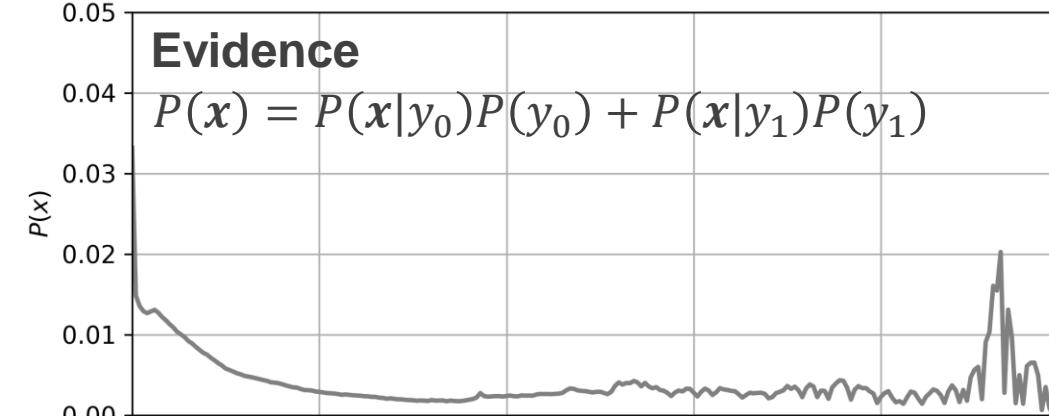
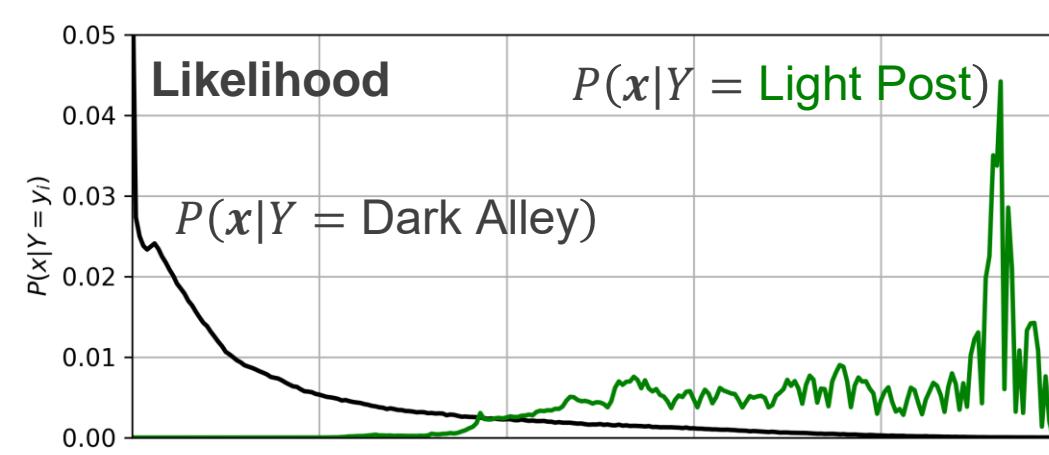
Class 1: Light Post y_1

Class 0: Dark Alley y_0



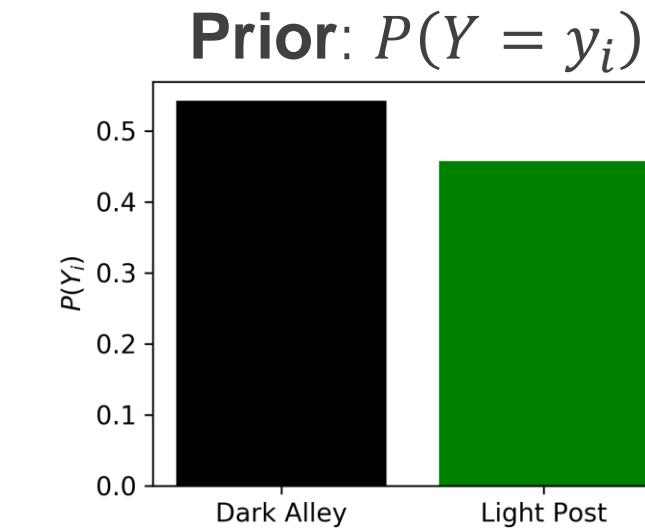
Bayes' Rule

$$\text{Posterior} \quad P(Y = y_i|x) = \frac{\text{Likelihood} \quad P(x|Y = y_i)P(Y = y_i)}{\text{Evidence} \quad P(x)}$$



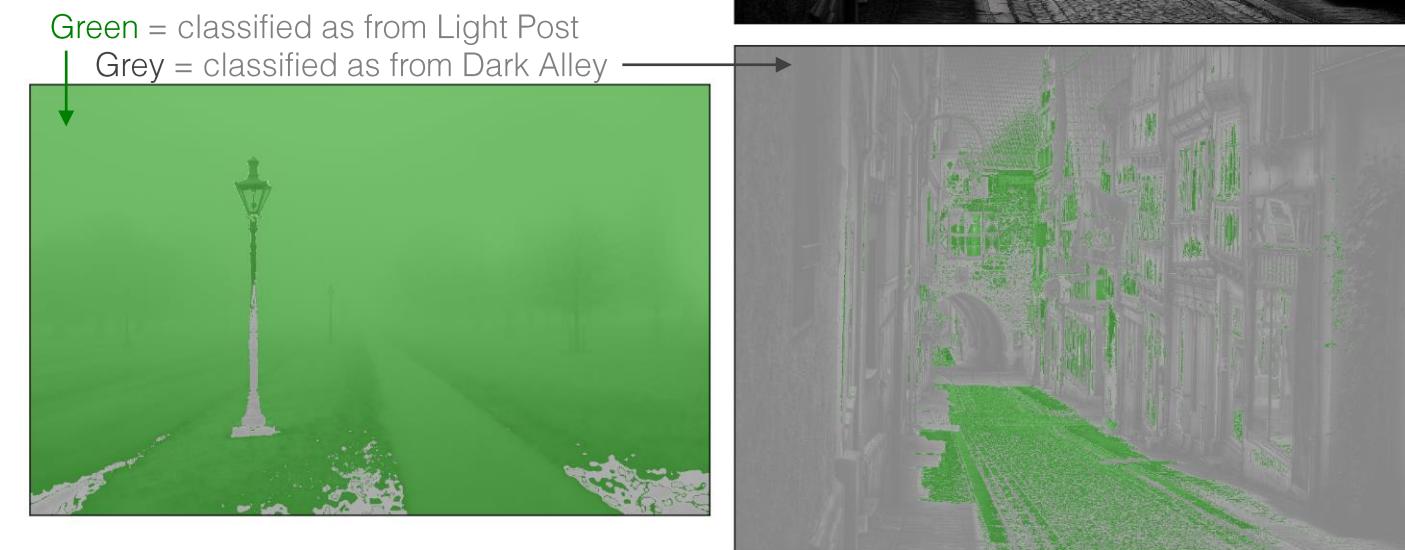
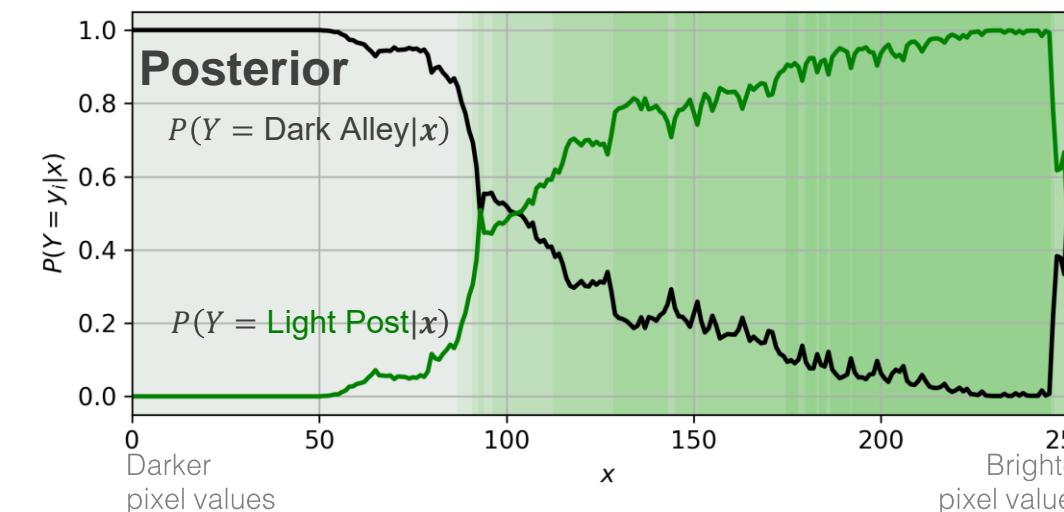
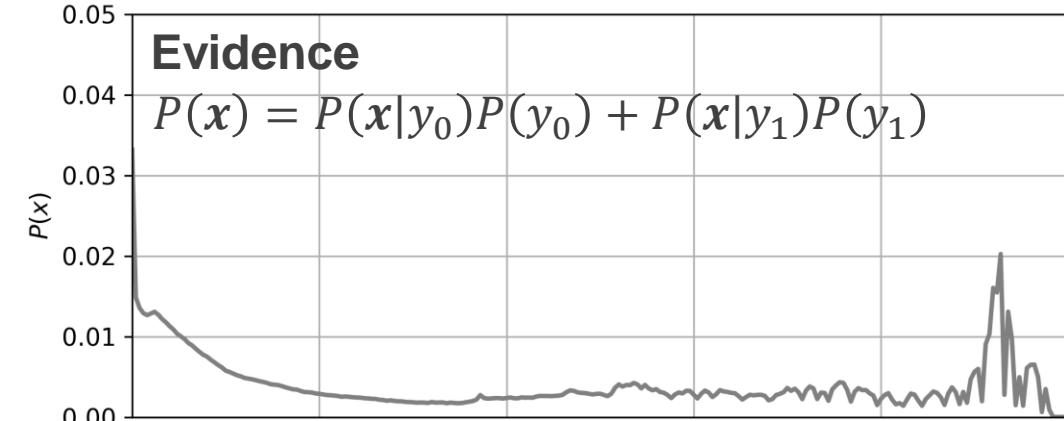
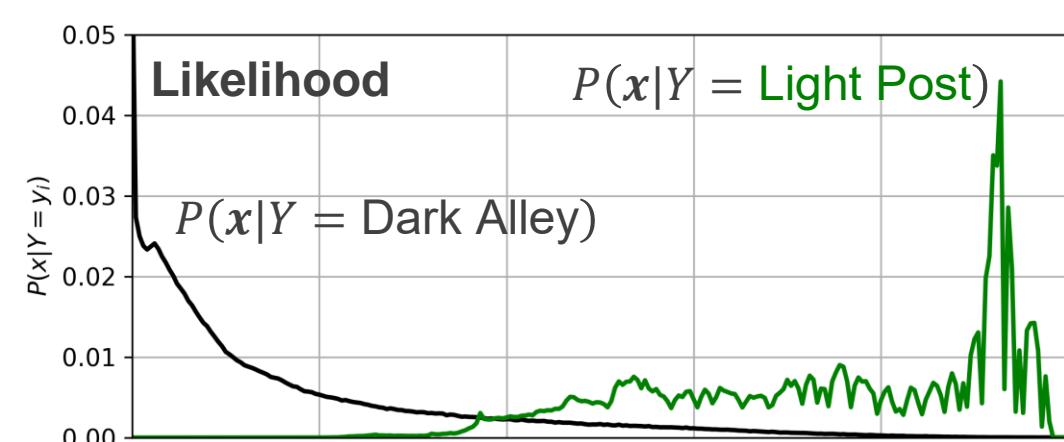
Class 1: Light Post y_1

Class 0: Dark Alley y_0



Decision rule:

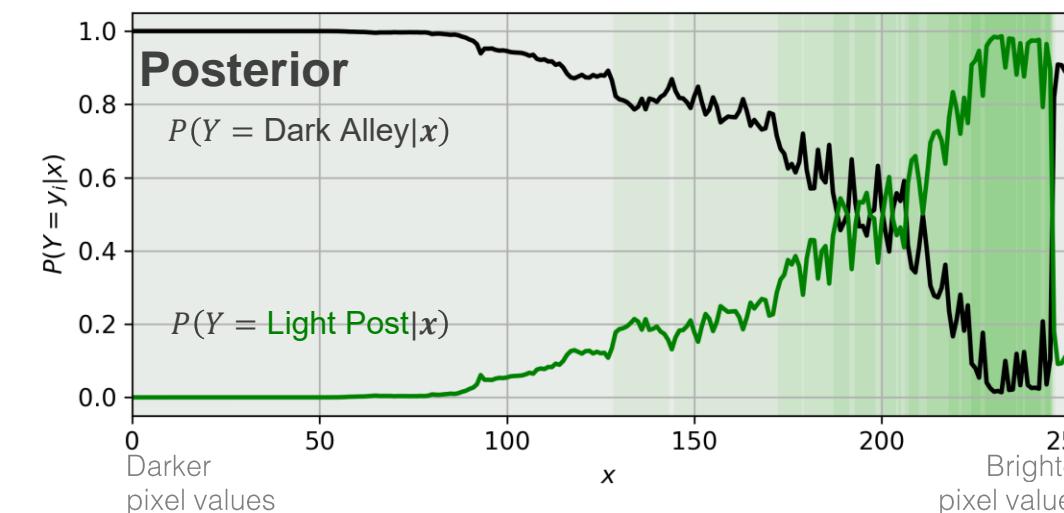
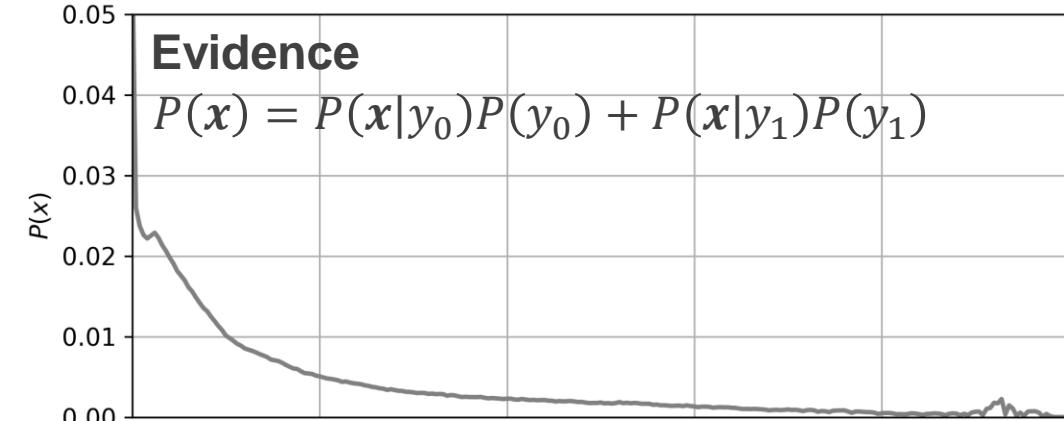
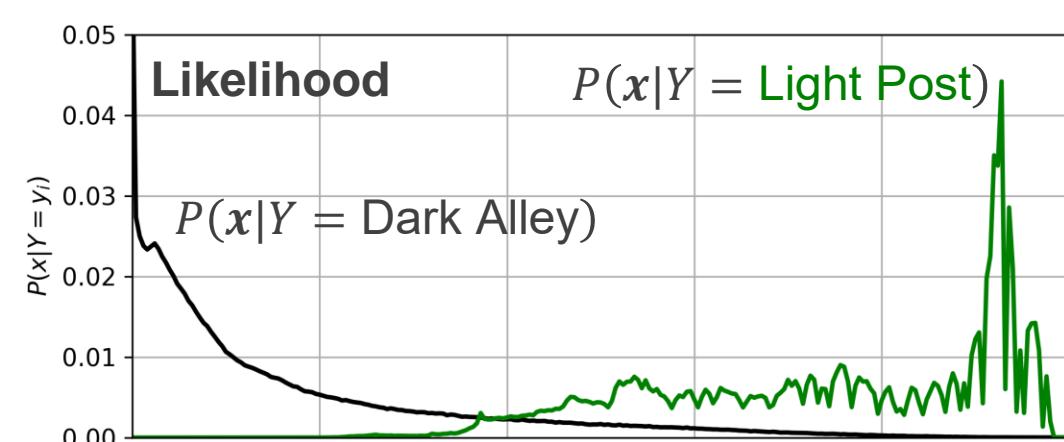
If $P(Y = \text{Light Post}|x) > P(Y = \text{Dark Alley}|x)$ then **Light Post**
else **Dark Alley**



Classifying each of the individual pixels as being either from **Light Post** or **Dark Alley** results in classification above

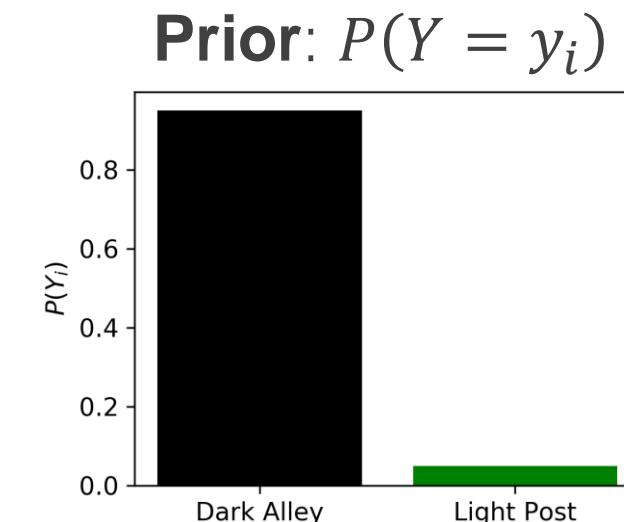
Decision rule:

If $P(Y = \text{Light Post}|x) > P(Y = \text{Dark Alley}|x)$ then **Light Post**
else **Dark Alley**



Class 1: Light Post y_1

Class 0: Dark Alley y_0



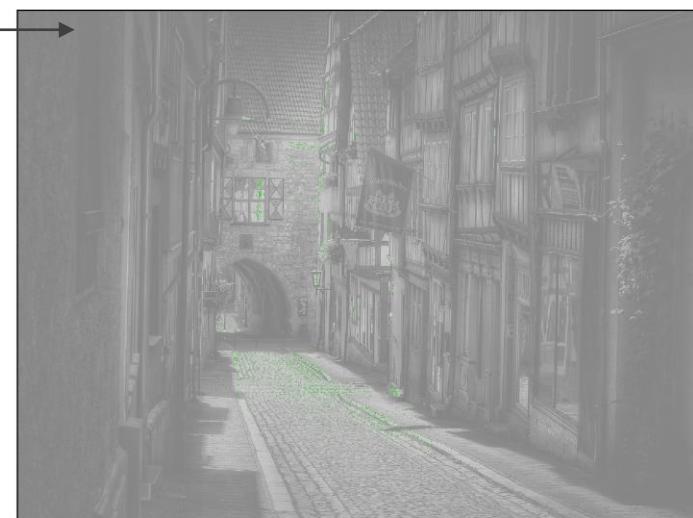
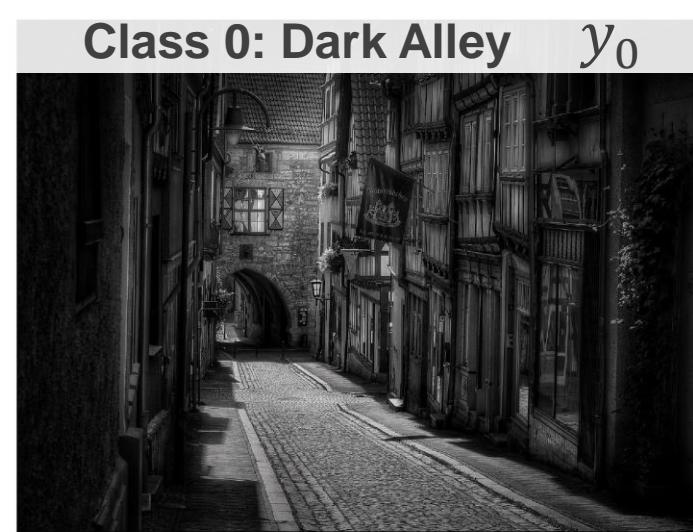
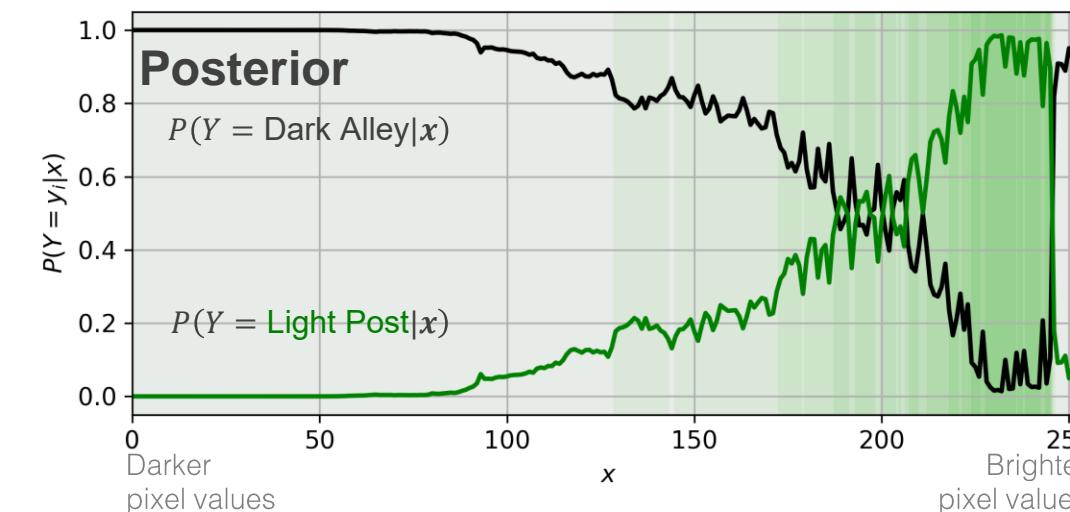
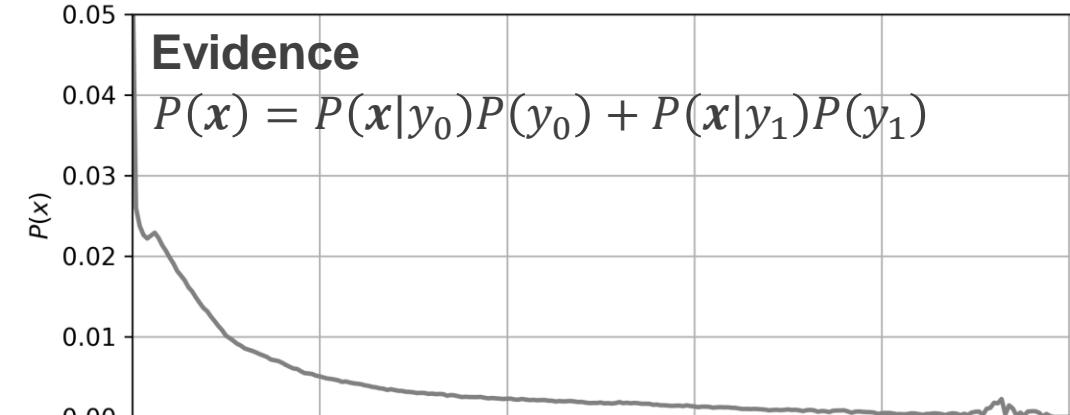
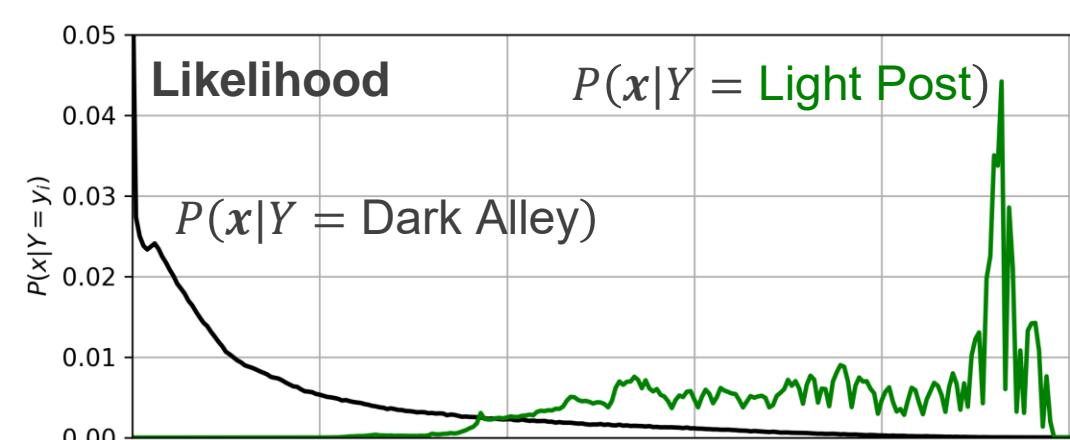
Let's assume we had reason to believe that the random sampling of pixels favored the **Dark Alley**

Bayes' Rule

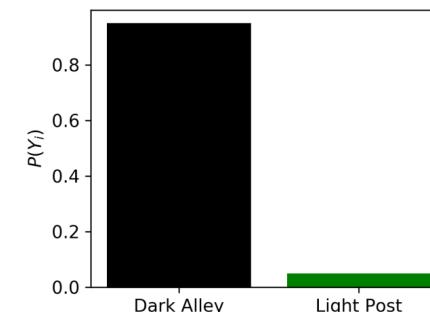
Posterior

$$P(Y = y_i|x) = \frac{\text{Likelihood} \quad \text{Prior}}{\text{Evidence}}$$

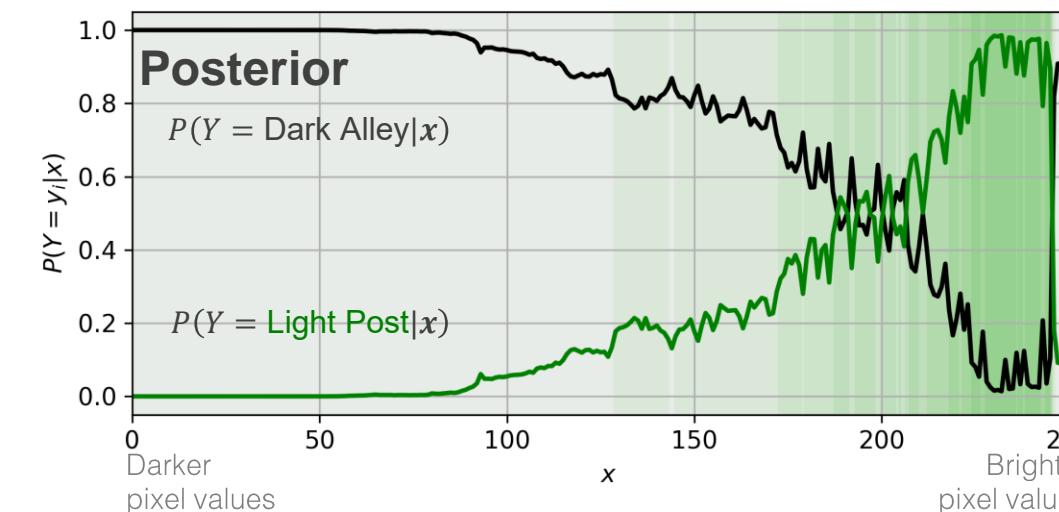
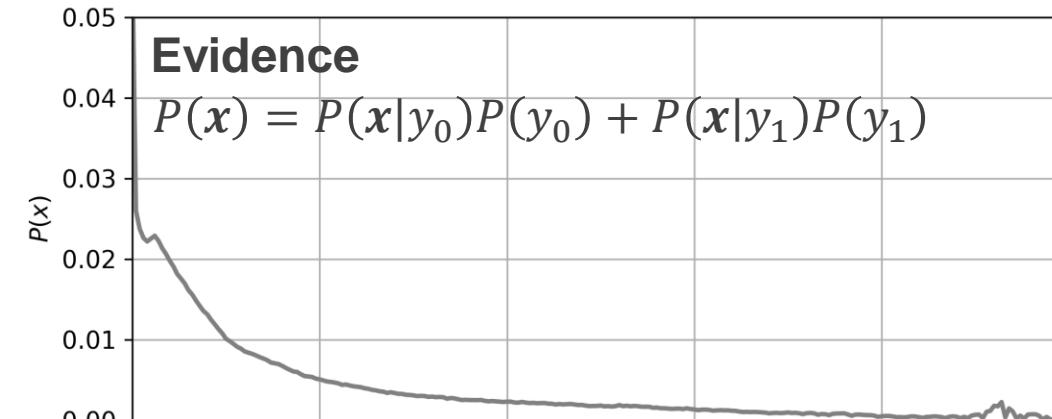
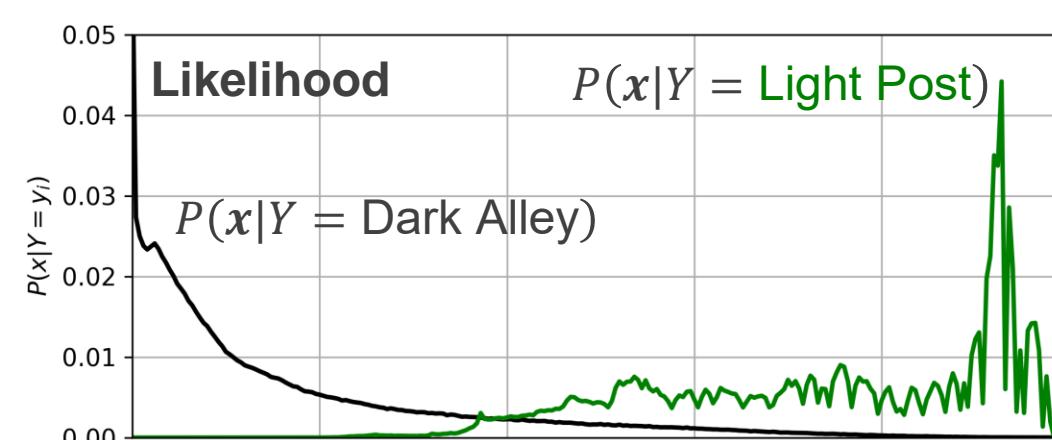
$$P(Y = y_i|x) = \frac{P(x|Y = y_i)P(Y = y_i)}{P(x)}$$



Prior: $P(Y = y_i)$



Let's assume we had reason to believe that the random sampling of pixels favored the **Dark Alley**



Generative models model the **likelihood**

These can also be used to generate synthetic data

$$P(Y = y_i|x) = \frac{\text{Likelihood} \quad \text{Prior}}{\text{Evidence}}$$

Likelihood

Posterior

Prior

Evidence

Discriminative models model the **posterior**

Or they just directly estimate labels without any probabilistic interpretation, $f(\mathbf{x}) \rightarrow \mathbf{y}$

Discriminant Functions

Assume we have c different classes

Define the **posterior probability** as a discriminant function: $d_i(x) = P(y = i|x)$

For a new sample, classify it as the class with the **largest $d_i(x)$**

Discriminant Functions

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

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

$$\begin{aligned}d_i(x) &= P(y = i | \mathbf{x}) = \frac{P(\mathbf{x}|y = i)P(y = i)}{P(\mathbf{x})} \\&= \frac{P(\mathbf{x}|y = i)P(y = i)}{\sum_{i=1}^c P(\mathbf{x}|y = i)P(y = i)}\end{aligned}\rightarrow$$

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

Denominator is the same for all classes i , so it won't help us tell which class's posterior is higher relative to other classes, so we ignore it going forward

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

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

Discriminant Functions

$$d_i(x) = P(y = i|x) = P(x|y = i)P(y = i)$$

Or in log form:

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

Equally valid discriminant functions
since log is monotonic

- 1 Assume a form for $P(x|y = i)$

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

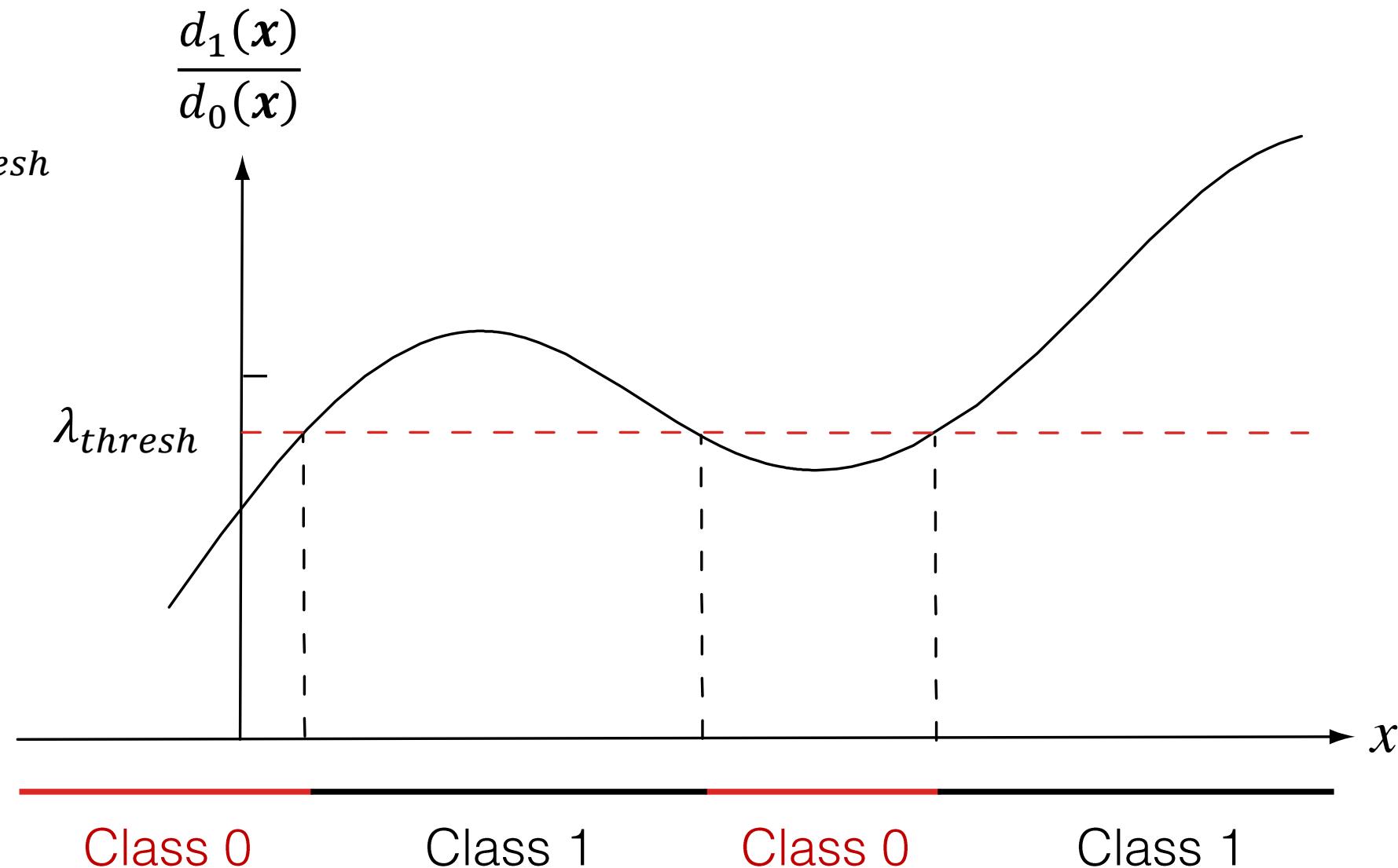
Applies to both binary and multiclass problems

Discriminant Functions: 2 classes

Decision rule:

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

Otherwise, class 0

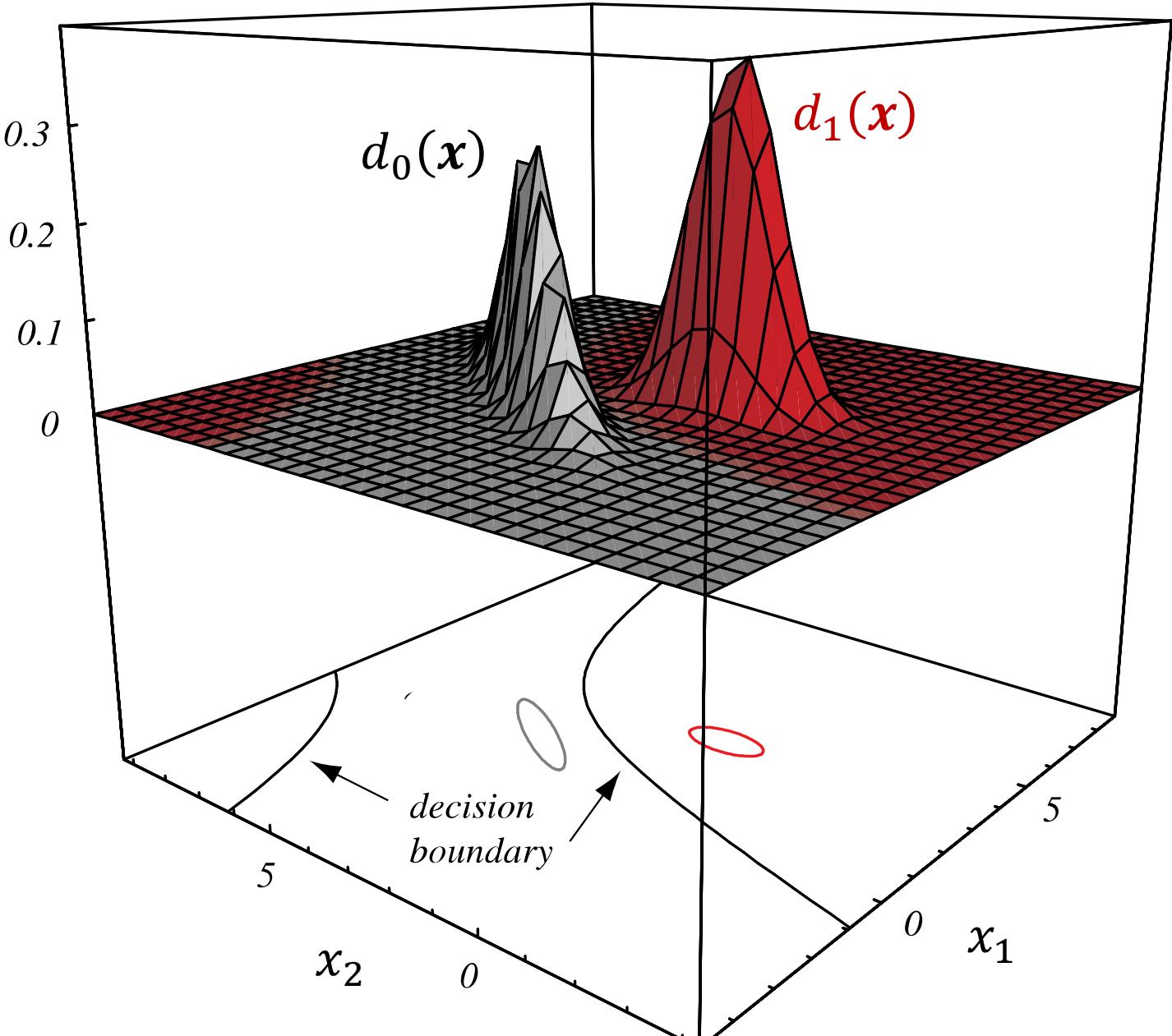


Discriminant Functions: 2 classes, 2 dimensions

Decision rule:

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

Otherwise, class 0



Discriminant Function: 2 classes

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

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

Assumes these likelihoods are normal

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

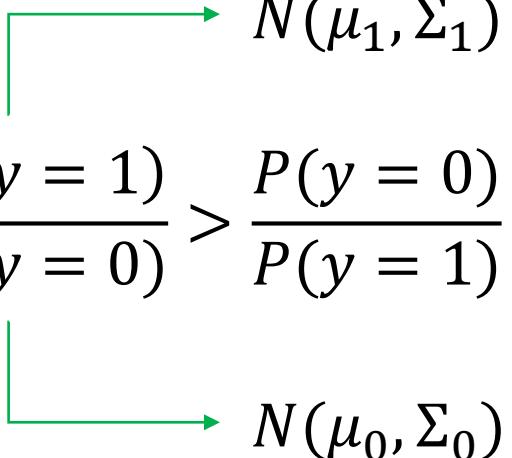
$N(\mu_1, \Sigma_1)$ $N(\mu_0, \Sigma_0)$

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



The diagram consists of two green arrows originating from the terms $P(\mathbf{x}|y=1)$ and $P(\mathbf{x}|y=0)$ in the likelihood ratio equation. The top arrow points to the term $N(\mu_1, \Sigma_1)$, and the bottom arrow points to the term $N(\mu_0, \Sigma_0)$.

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

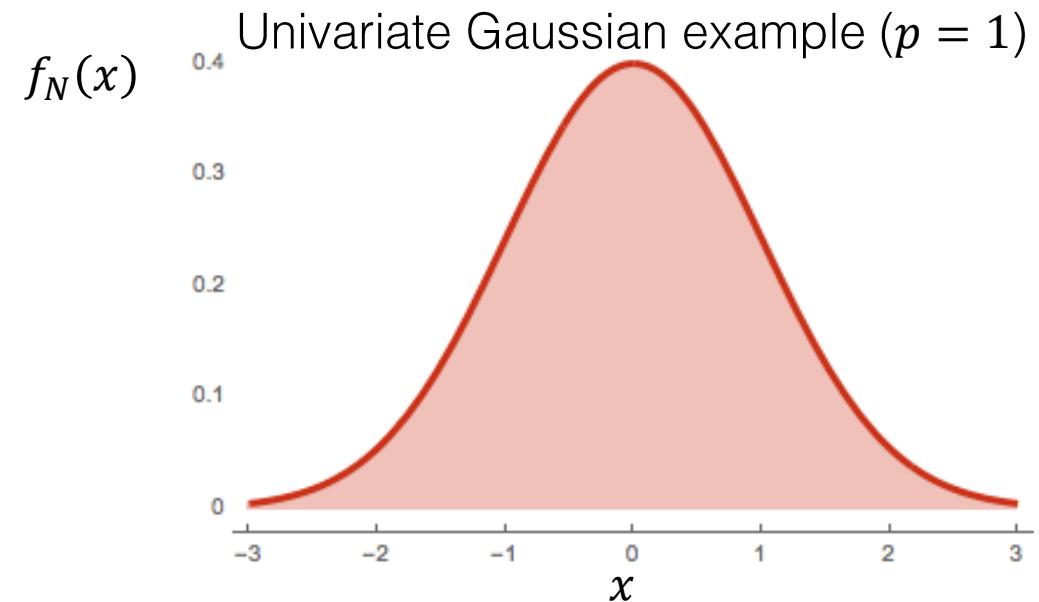
Multivariate Gaussian

Univariate Gaussian (1 predictor)

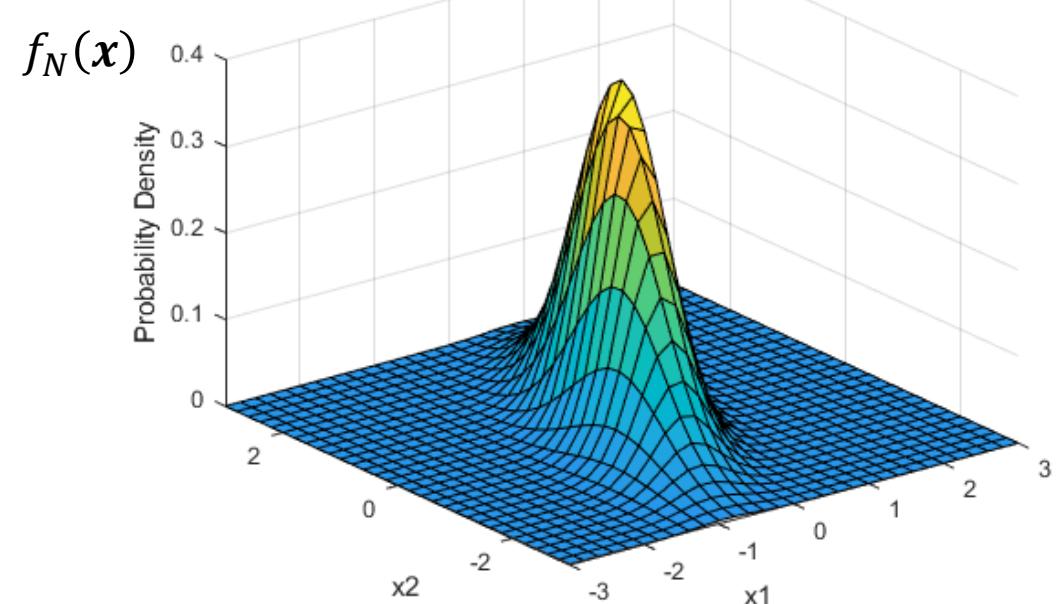
$$f_N(x) = \frac{1}{(2\pi)^{1/2}\sigma} \exp\left[-\frac{1}{2\sigma^2}(x - \mu)^2\right]$$

Multivariate Gaussian (p predictors)

$$f_N(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right]$$



Multivariate Gaussian example ($p = 2$)



Images from Brilliant.org (top) and the Mathworks (bottom)

Linear Discriminant Analysis: 2 Class

Σ is the same for both classes

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

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

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

We can rewrite our decision rule as:

(see appendix slides
for full derivation)

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

Or simply as:

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

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

Linear Discriminant Analysis: Multiclass

Σ is the same for all classes

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

$$\delta_k(\mathbf{x}) = P(\mathbf{x}|y=k)P(y=k) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\right] \pi_k \quad P(y=k) \triangleq \pi_k$$

$$\begin{aligned} \ln|\delta_k(\mathbf{x})| &= -\frac{p}{2}\ln|2\pi| - \frac{p}{2}\ln|\Sigma| - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_k) + \ln|\pi_k| \\ &\quad - \frac{1}{2}[x^T \boldsymbol{\Sigma}^{-1} x - 2\mathbf{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k + \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k] \end{aligned}$$

$$\ln|\delta_k(\mathbf{x})| = -\cancel{\frac{p}{2}\ln|2\pi|} - \cancel{\frac{p}{2}\ln|\Sigma|} - \frac{1}{2}\cancel{\mathbf{x}^T \boldsymbol{\Sigma}^{-1} x} - \mathbf{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k + \frac{1}{2}\boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k + \ln|\pi_k|$$

$$\ln|\delta_k(\mathbf{x})| = -\mathbf{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k + \frac{1}{2}\boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k + \ln|\pi_k|$$

Since we'll be looking for the choice of class k that maximizes the value of $\delta_k(\mathbf{x})$, we can ignore terms that are independent of class

We compute this for each k and assign the class with the largest discriminant $\delta_k(\mathbf{x})$

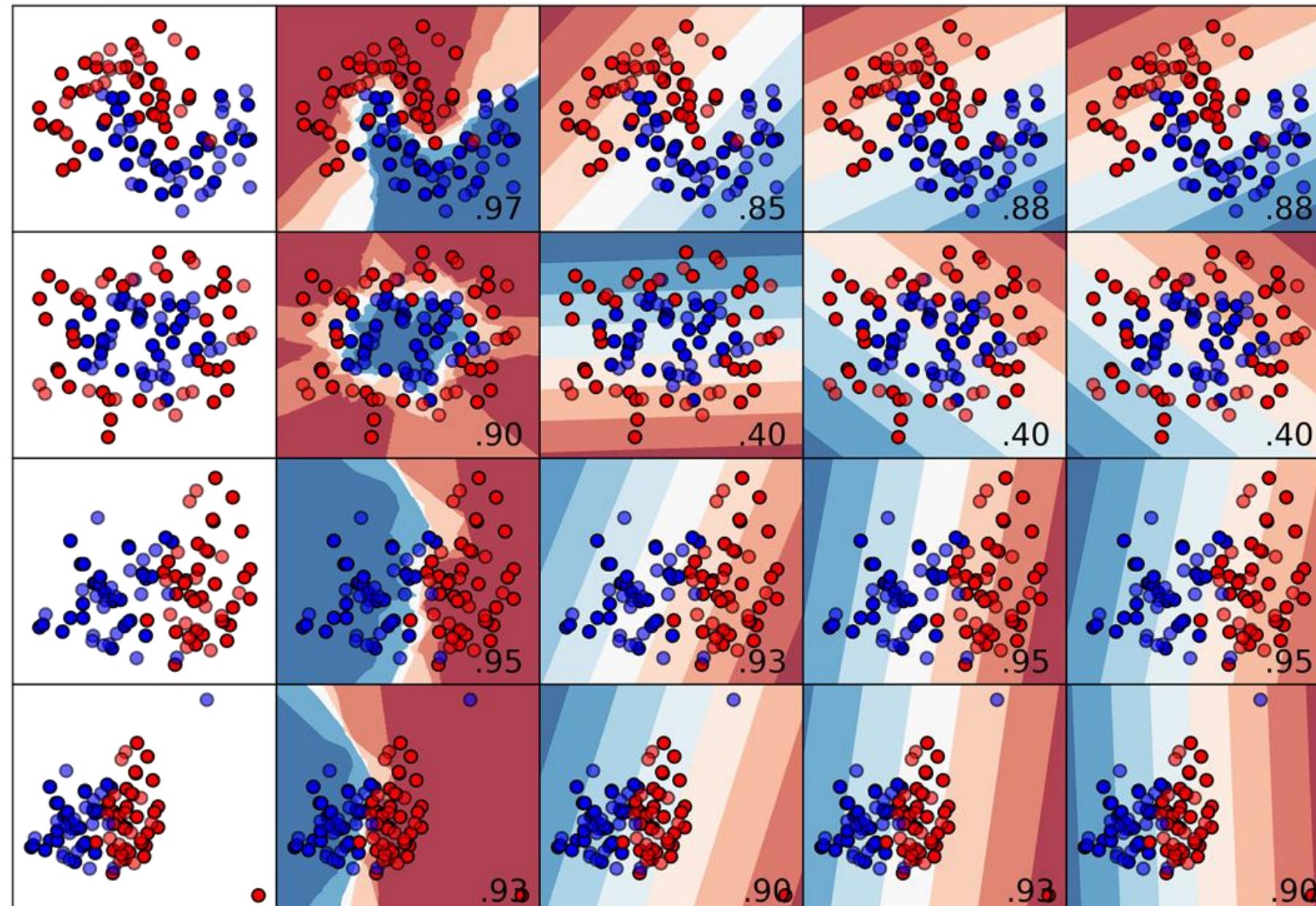
Input data

KNN (k=5)

Perceptron

Logistic Reg.

LDA



Quadratic Discriminant Analysis

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

$$d_k(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right] \pi_k$$

We assume a normal distribution, but **different covariance matrices**

Produces a quadric decision boundary

Summary Comparison

	Fisher's Linear Discriminant (FLD)	Linear Discriminant Analysis (LDA)	Quadratic Discriminant Analysis (QDA)
Assumes Gaussian Likelihood $P(\mathbf{x} y)$ (class conditional density)	No	Yes	Yes
Assumes equivalent covariance matrices $\Sigma_i = \Sigma_j$	No	Yes	No

Reduce the dimensionality of the data to make them more separable

LDA and QDA easily extend to multiclass problems

**These discriminant analysis
techniques are supervised
dimensionality reduction techniques
that we use for classification**

Input data

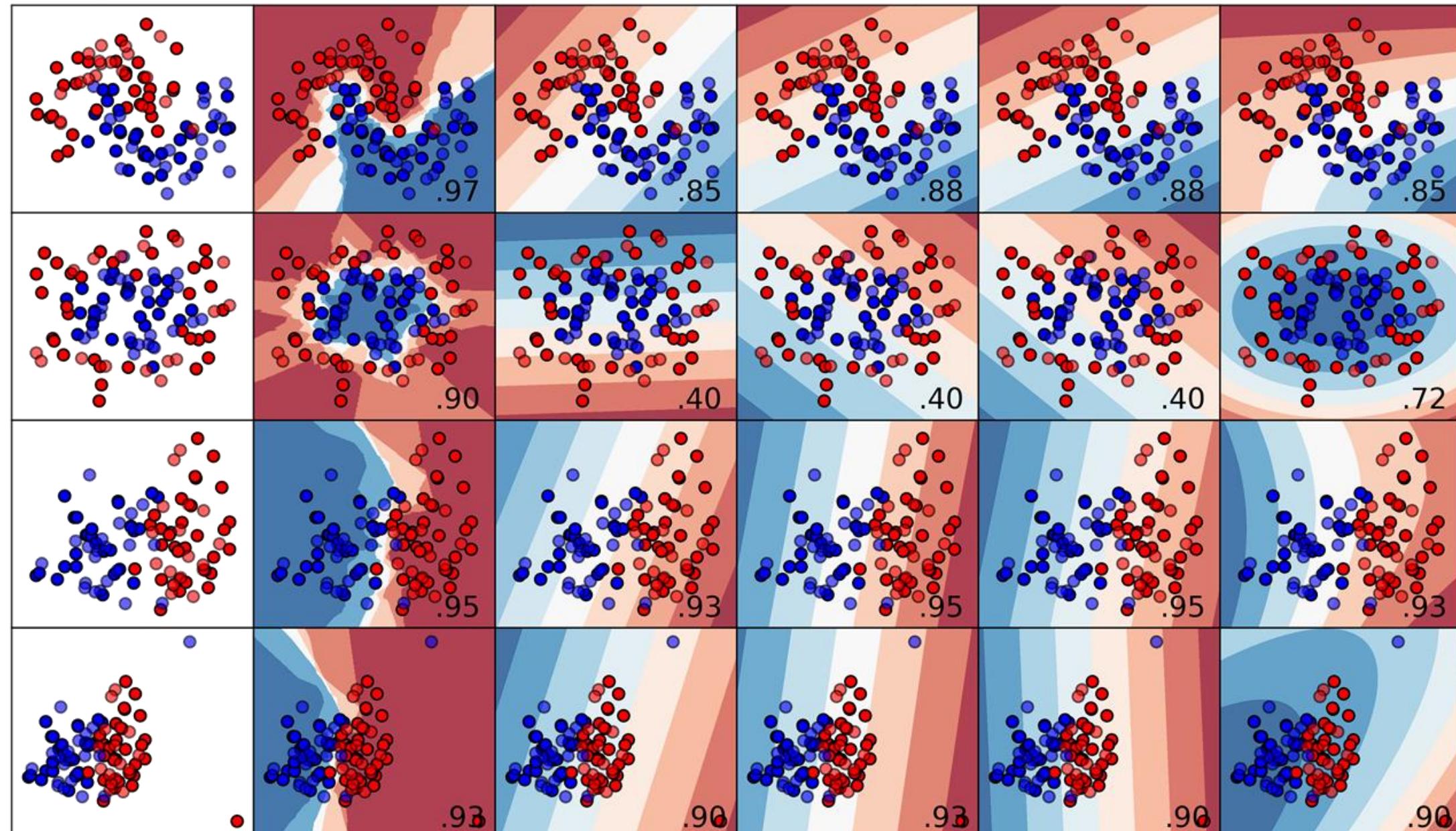
KNN (k=5)

Perceptron

Logistic Reg.

LDA

QDA



Naïve Bayes

Sometimes called “Idiot’s Bayes”

For independent events: A, B, and C
 $P(A \text{ and } B \text{ and } C) = P(A)P(B)P(C)$

Start with our original expression for our posterior distribution

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

Write out the full expression with all the terms in x
(assume p predictors/features)

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

Assumption: Given the class, the features are independent

Note: The denominator (evidence) is a constant if we know the values of the predictor variables

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

Predict the class with the highest posterior probability

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

Naïve Bayes

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

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

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, $P(y = i|x_1, x_2, \dots, x_p)$

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

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

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

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

Naïve Bayes: Parameters

p predictors, c classes

For each predictor, x_i , and class, y_j :

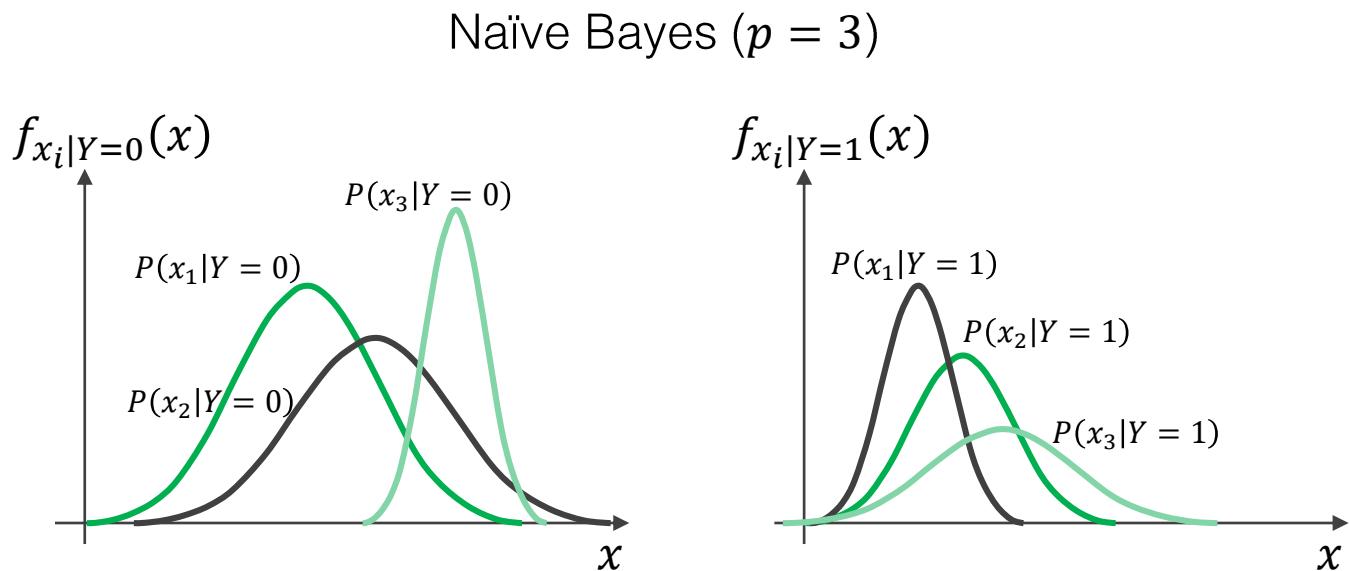
$$(\mu_{ij}, \sigma_{ij}^2)$$

Total parameters = $2cp$

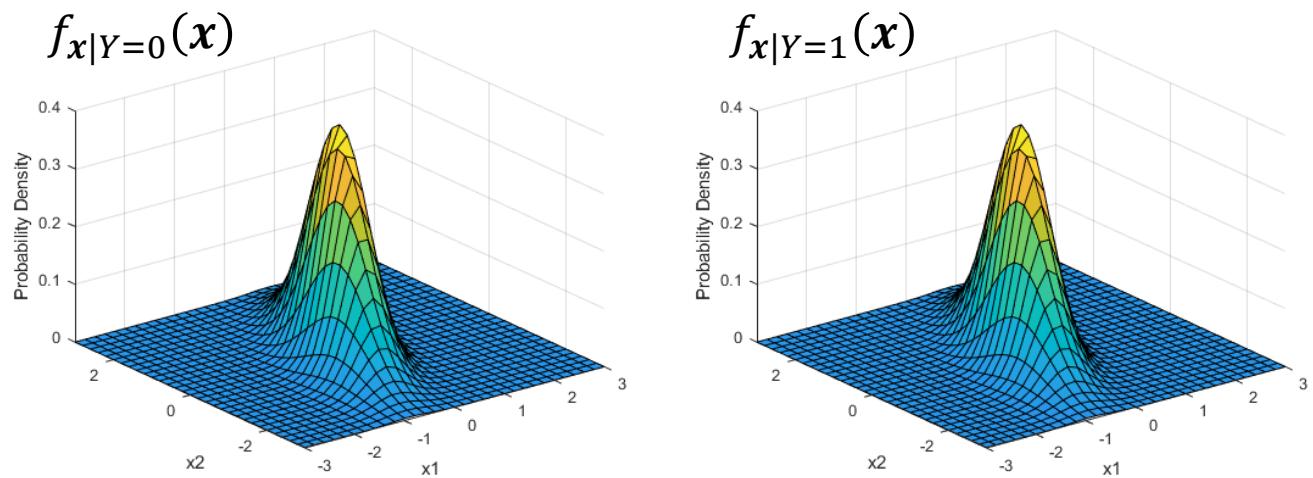
Without the Naïve Bayes assumption,
each class would be a multivariate
Gaussian with $(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$

$$\boldsymbol{\mu}_j = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix}, \boldsymbol{\Sigma}_j = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \cdots & \sigma_{1p}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \cdots & \sigma_{2p}^2 \\ \vdots & \vdots & & \vdots \\ \sigma_{p1}^2 & \sigma_{p2}^2 & \cdots & \sigma_{pp}^2 \end{bmatrix}$$

Total parameters = $c(p + p^2)$



Multivariate Gaussian (example shown for $p = 2$)



Images from Brilliant.org (top) and the Mathworks (bottom)

Input data

KNN (k=5)

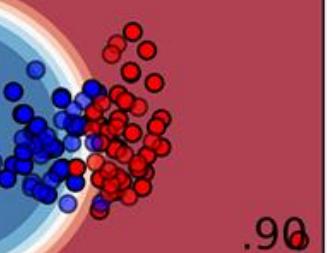
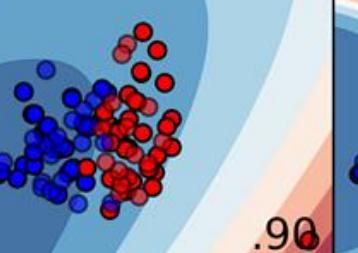
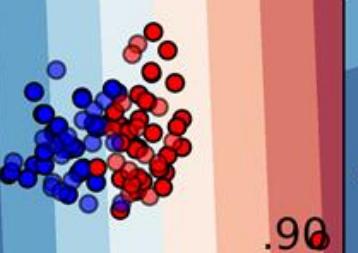
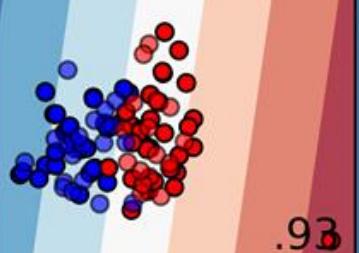
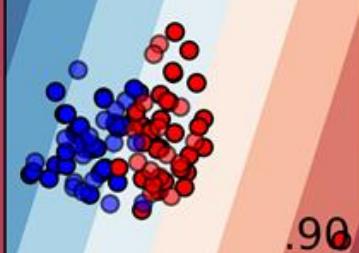
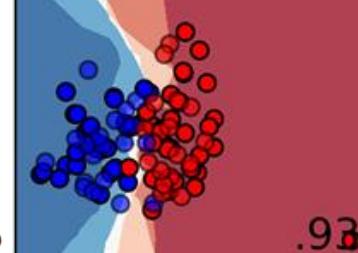
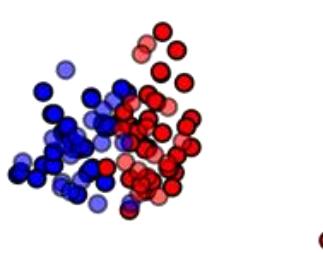
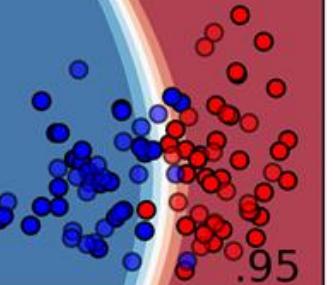
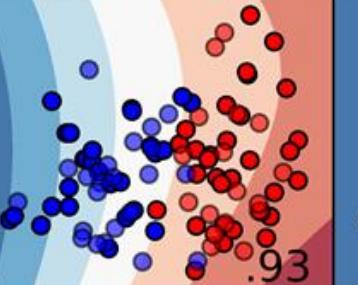
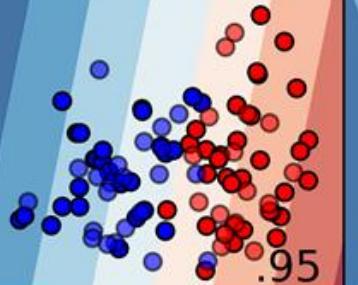
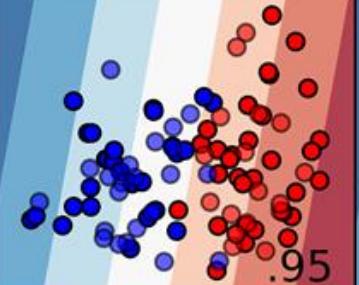
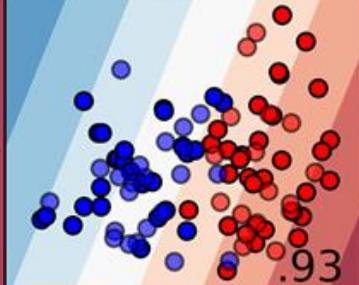
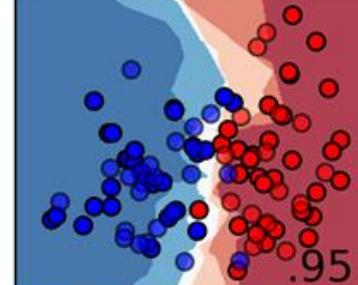
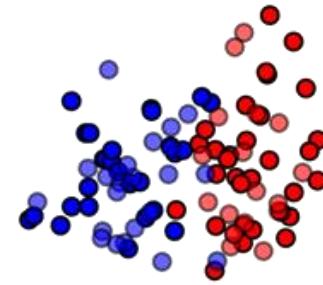
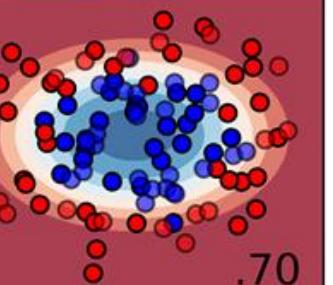
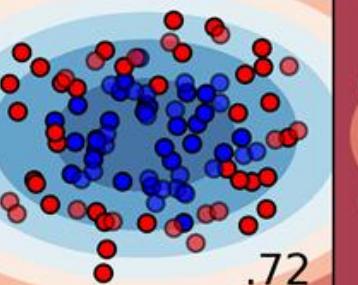
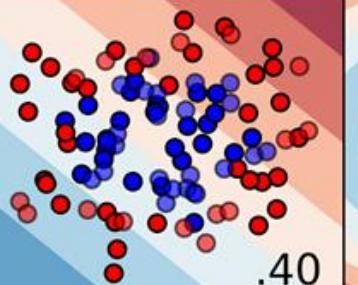
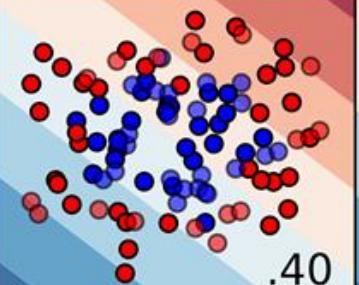
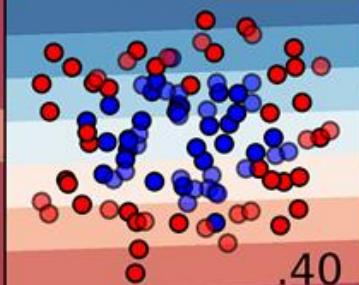
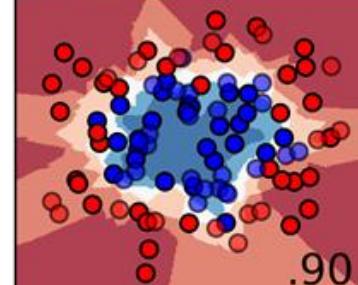
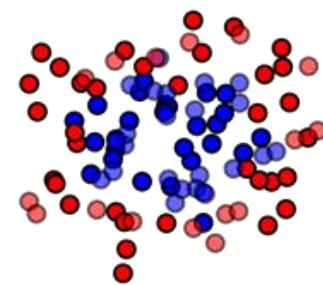
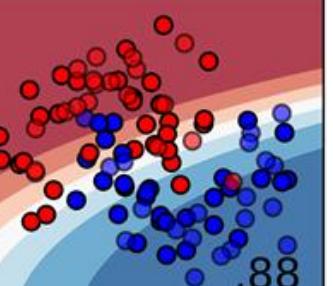
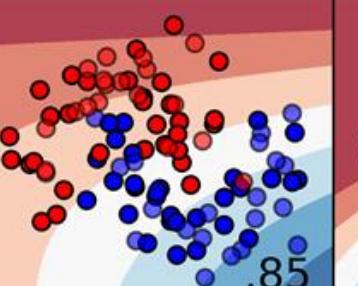
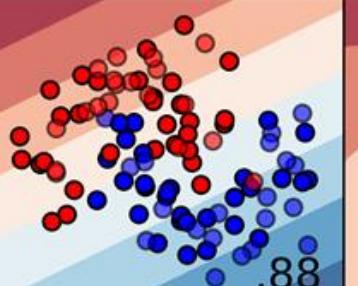
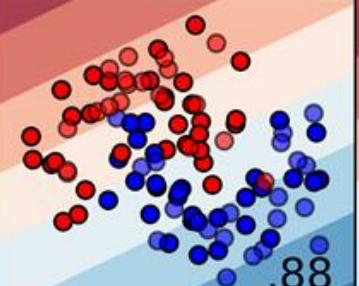
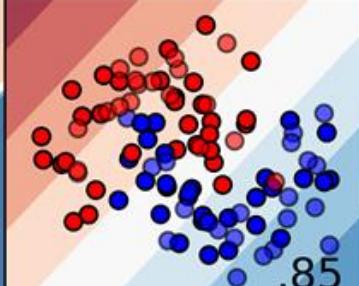
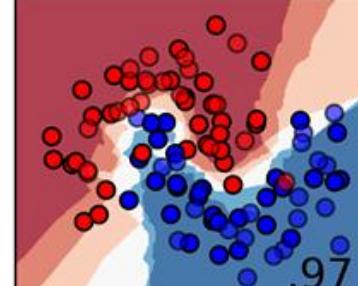
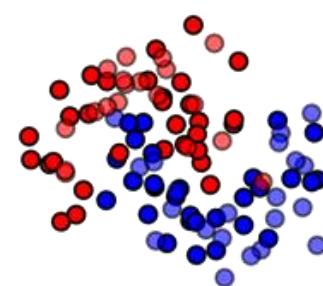
Perceptron

Logistic Reg.

LDA

QDA

Naive Bayes



Classifiers

Covered so far

K-Nearest Neighbors

Perceptron

Logistic Regression

Fisher's Linear Discriminant

Linear Discriminant Analysis

Quadratic Discriminant Analysis

Naïve Bayes

Appendix (Derivations)

FLD: Fisher Criterion Maximization

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

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

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

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

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

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

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

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

Factoring out the
 \mathbf{w} in denominator

FLD: Fisher Criterion Maximization

$$J(\mathbf{w}) = \frac{\mathbf{w}^T (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T \mathbf{w}}{\mathbf{w}^T [\sum_{i \in C_1} (\mathbf{x}_i - \mathbf{m}_1)(\mathbf{x}_i - \mathbf{m}_1)^T + \sum_{i \in C_2} (\mathbf{x}_i - \mathbf{m}_2)(\mathbf{x}_i - \mathbf{m}_2)^T] \mathbf{w}}$$
$$\mathcal{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T$$
$$\mathcal{S}_W = \sum_{i \in C_1} (\mathbf{x}_i - \mathbf{m}_1)(\mathbf{x}_i - \mathbf{m}_1)^T + \sum_{i \in C_2} (\mathbf{x}_i - \mathbf{m}_2)(\mathbf{x}_i - \mathbf{m}_2)^T$$
$$= \Sigma_1 + \Sigma_2 \quad \Sigma_i = \text{covariance matrix for class } i$$

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

Generalized
Raleigh Quotient

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

FLD: Fisher Criterion Maximization

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

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

Recall the quotient rule for differentiation:

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

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

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

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

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

Therefore, we can write:

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

We want to solve this for \mathbf{w}

FLD: Fisher Criterion Maximization

$$\frac{dJ(\mathbf{w})}{d\mathbf{w}} = \frac{(2\mathbf{w}^T \mathbf{S}_B)(\mathbf{w}^T \mathbf{S}_W \mathbf{w}) - (\mathbf{w}^T \mathbf{S}_B \mathbf{w})(2\mathbf{w}^T \mathbf{S}_W)}{(\mathbf{w}^T \mathbf{S}_W \mathbf{w})^2} = 0 \quad \text{We want to solve this for } \mathbf{w}$$

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

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

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

α β $[1 \times D][D \times D][D \times 1] \rightarrow \text{scalar}$

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

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

FLD: Fisher Criterion Maximization

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

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

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

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

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

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

Property of matrix transposition:

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

The scatter matrices are symmetric:

$$\mathbf{S}_B = \mathbf{S}_B^T$$

$$\mathbf{S}_W = \mathbf{S}_W^T$$

Between-class scatter matrix:

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

Aside: dimensionality reduction

Rearranging, this is an eigenvalue problem

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

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

FLD: Fisher Criterion Maximization

$$\alpha\gamma(\mathbf{m}_2 - \mathbf{m}_1) = \beta S_W \mathbf{w}$$

Solving for \mathbf{w} :

$$\mathbf{w} = \frac{\alpha\gamma}{\beta} S_W^{-1} (\mathbf{m}_2 - \mathbf{m}_1) \quad \text{We only care about the direction of } \mathbf{w}$$

$$\mathbf{w} \propto S_W^{-1} (\mathbf{m}_2 - \mathbf{m}_1)$$

Note: if S_w is isotropic
(proportional to the identity matrix, i.e. if $S_w = aI$),
then this is just the difference between the means

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

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

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

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

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

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

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

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

Expanding this expression yields:

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

$$\begin{aligned} &= -\frac{1}{2} [\cancel{\mathbf{x}^T \boldsymbol{\Sigma}^{-1} \mathbf{x}} - \mathbf{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1 - \boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1} \mathbf{x} + \boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1 \\ &\quad - \cancel{\mathbf{x}^T \boldsymbol{\Sigma}^{-1} \mathbf{x}} + \mathbf{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_0 + \boldsymbol{\mu}_0^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - \boldsymbol{\mu}_0^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_0] \\ &= \frac{1}{2} [2\mathbf{x}^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0) \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)] \\ &= \mathbf{x}^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - \frac{1}{2} (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0) \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \end{aligned}$$

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

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

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

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

We can rewrite our decision rule as:

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

Or simply as:

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

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