Classification

Lecture 10

Classifiers

Covered so far

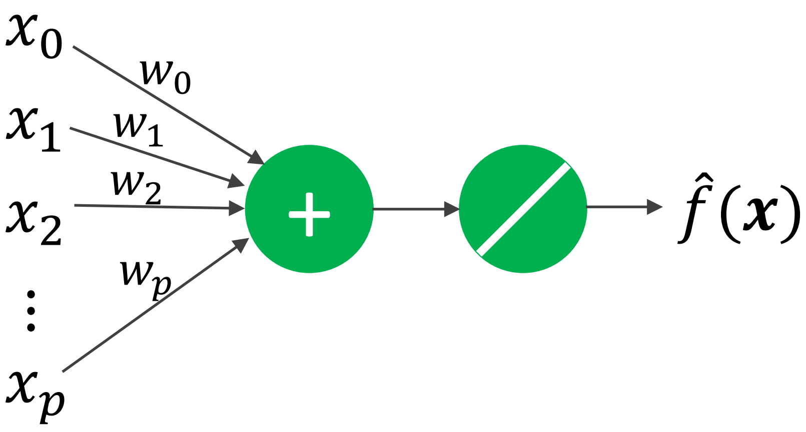
- K-Nearest Neighbors
- Perceptron
- Logistic Regression
- Fisher's Linear Discriminant
- Linear Discriminant Analysis
- Quadratic Discriminant Analysis
- Naïve Bayes

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

Remember linear models?

Linear Regression

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

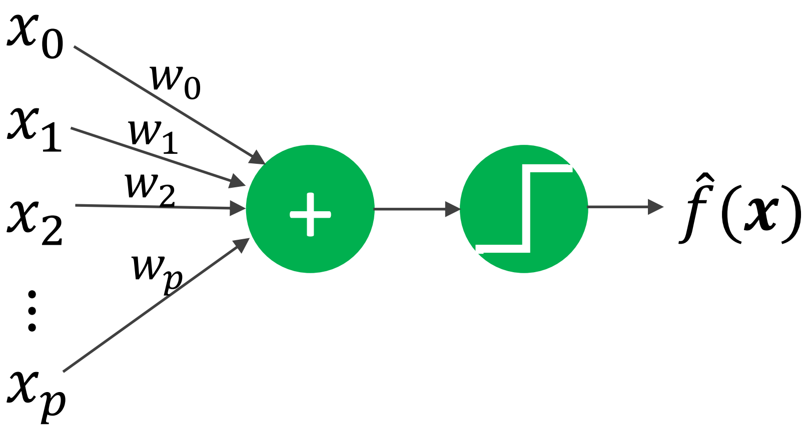


Linear Classification

Perceptron

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

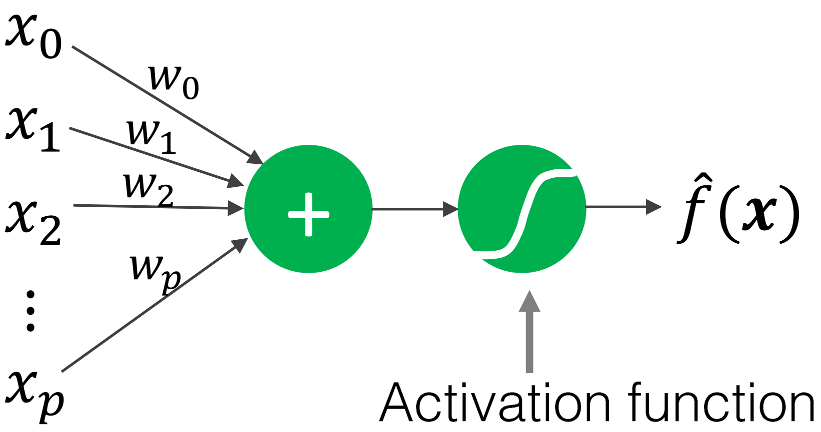
$$sign(x) = \begin{cases} 1 & x > 0 \\ -1 & \text{else} \end{cases}$$



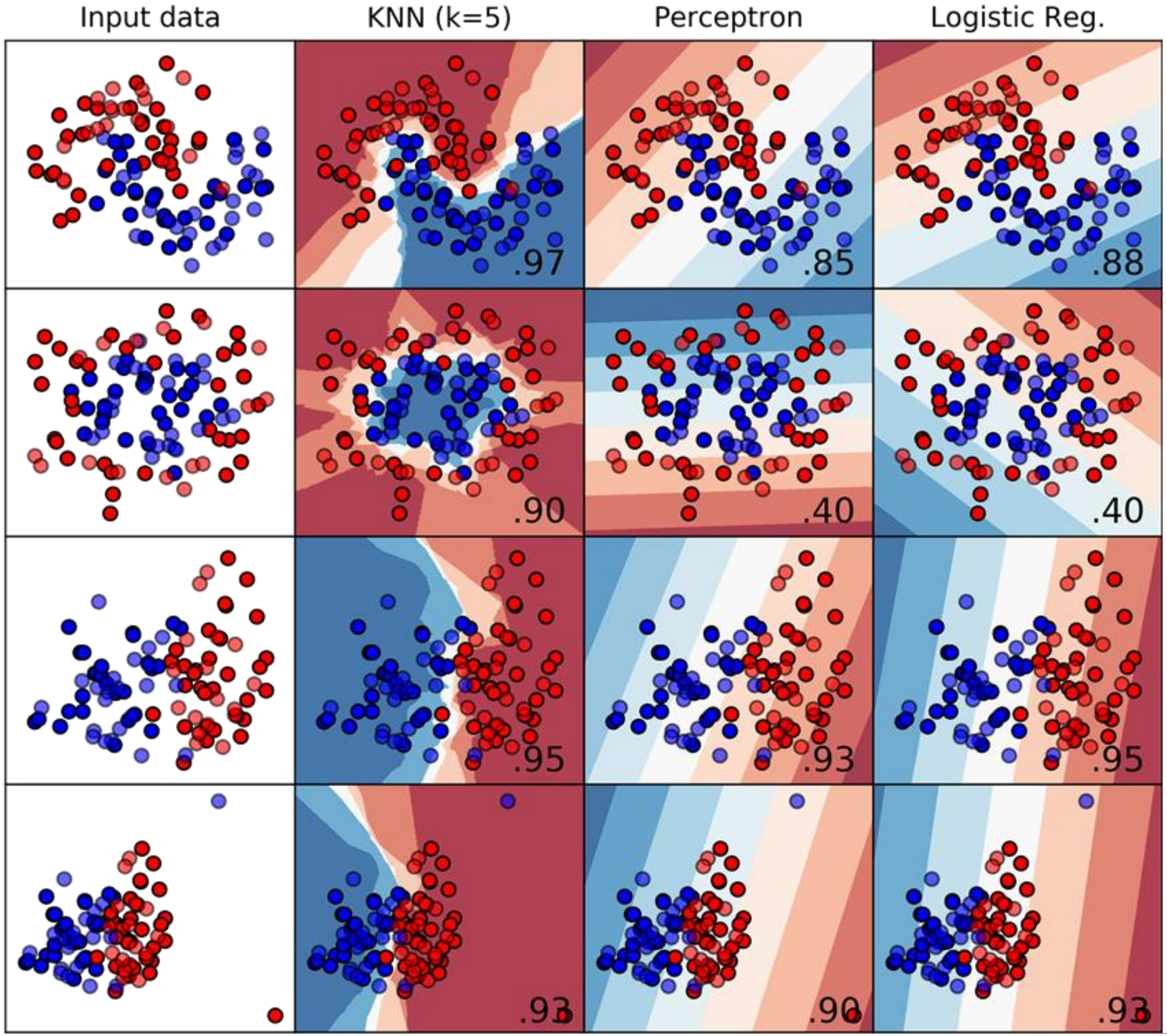
Logistic Regression

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

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



Source: Abu-Mostafa, Learning from Data, Caltech



Comparison of classifiers we've seen so far

Projections

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

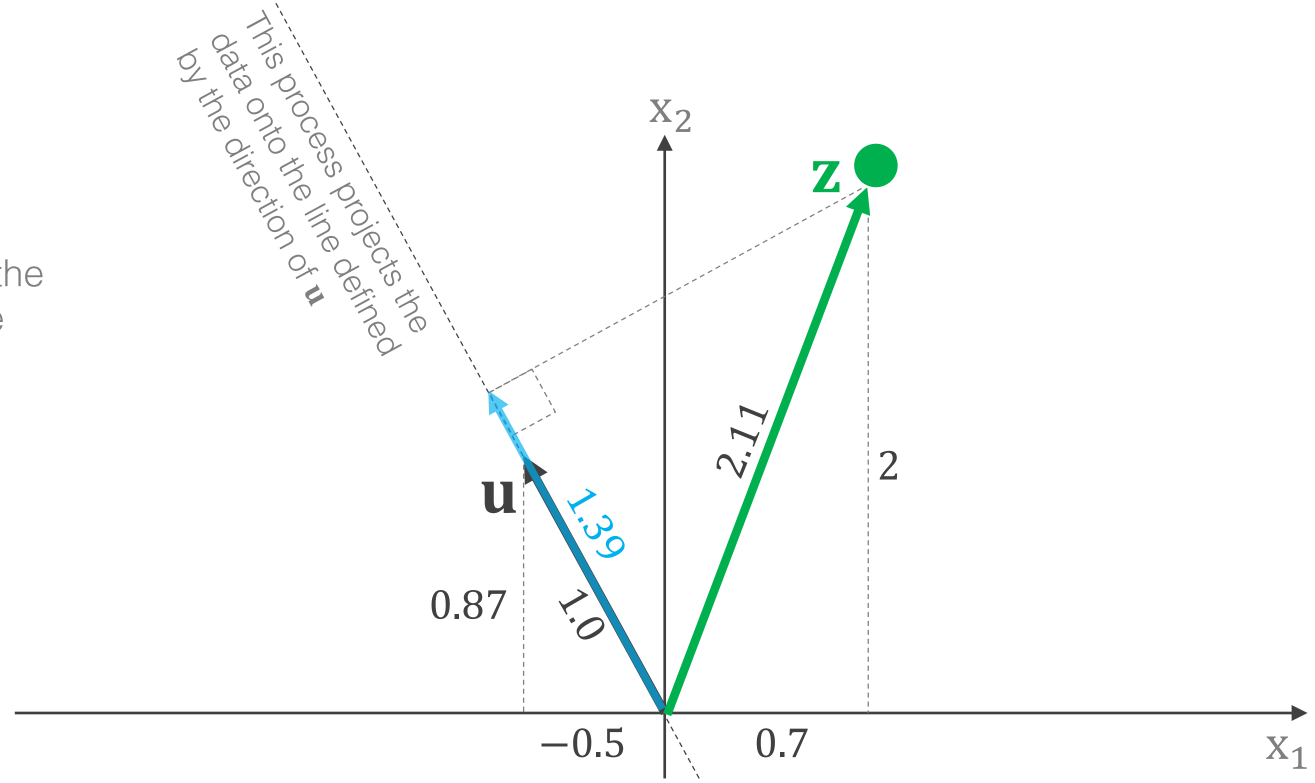
This is an inner product, but assuming $\mathbf{u}$ is a unit vector computes this as the magnitude (length) of the projection of $\mathbf{z}$ onto $\mathbf{u}$

$$= u_1 z_1 + u_2 z_2$$

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

$$= 1.39$$

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 $\boldsymbol{z}$ onto $\mathbf{u}$ would be calculated by normalizing the result by the length of $\mathbf{u}$:

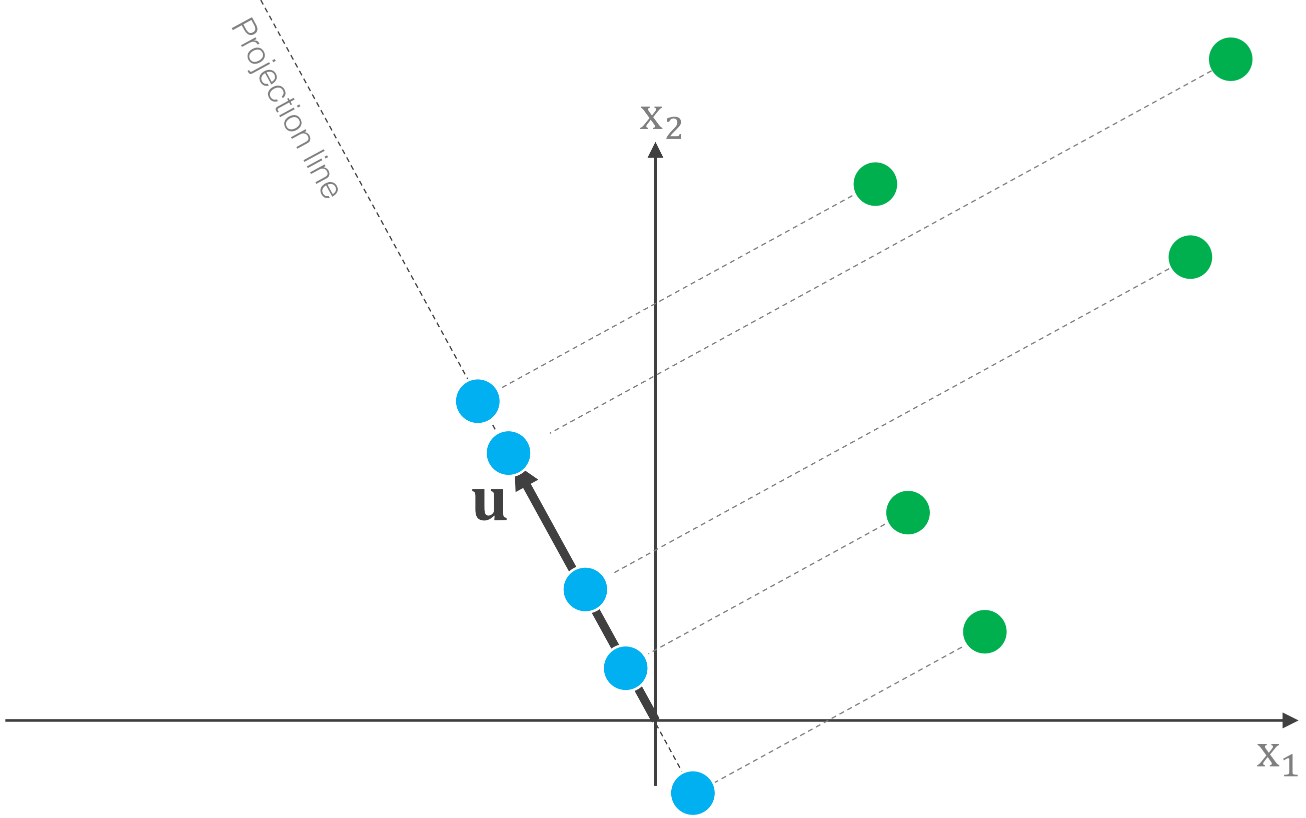
$$\frac{\mathbf{u}^T\mathbf{z}}{\mathbf{u}^T\mathbf{u}}$$

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

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

$$\mathrm{proj}_{\mathbf{u}}(\boldsymbol{z}) = \left(\frac{\mathbf{u}^T\mathbf{z}}{\mathbf{u}^T\mathbf{u}}\right)\frac{\mathbf{u}}{\mathbf{u}^T\mathbf{u}}$$

Projections

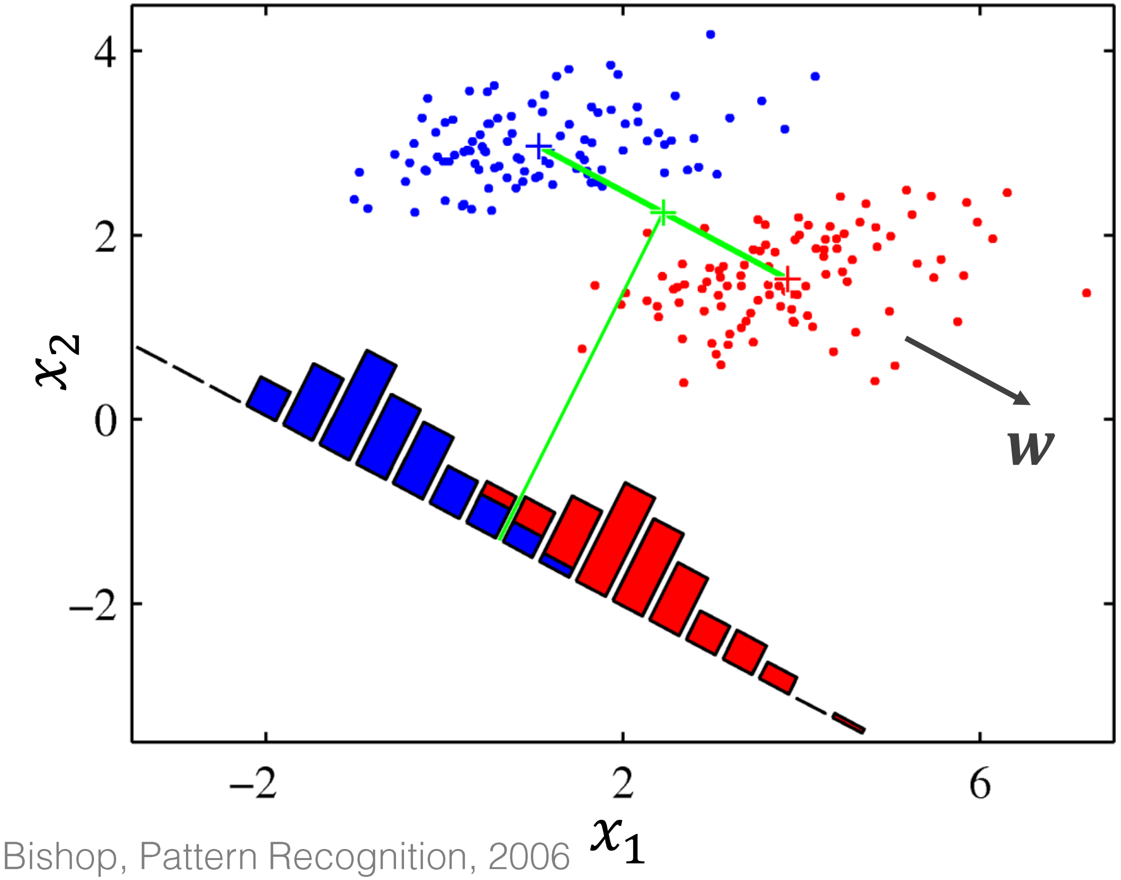


We could project any points in this space onto the line defined by the direction of unit vector $\mathbf{u}$

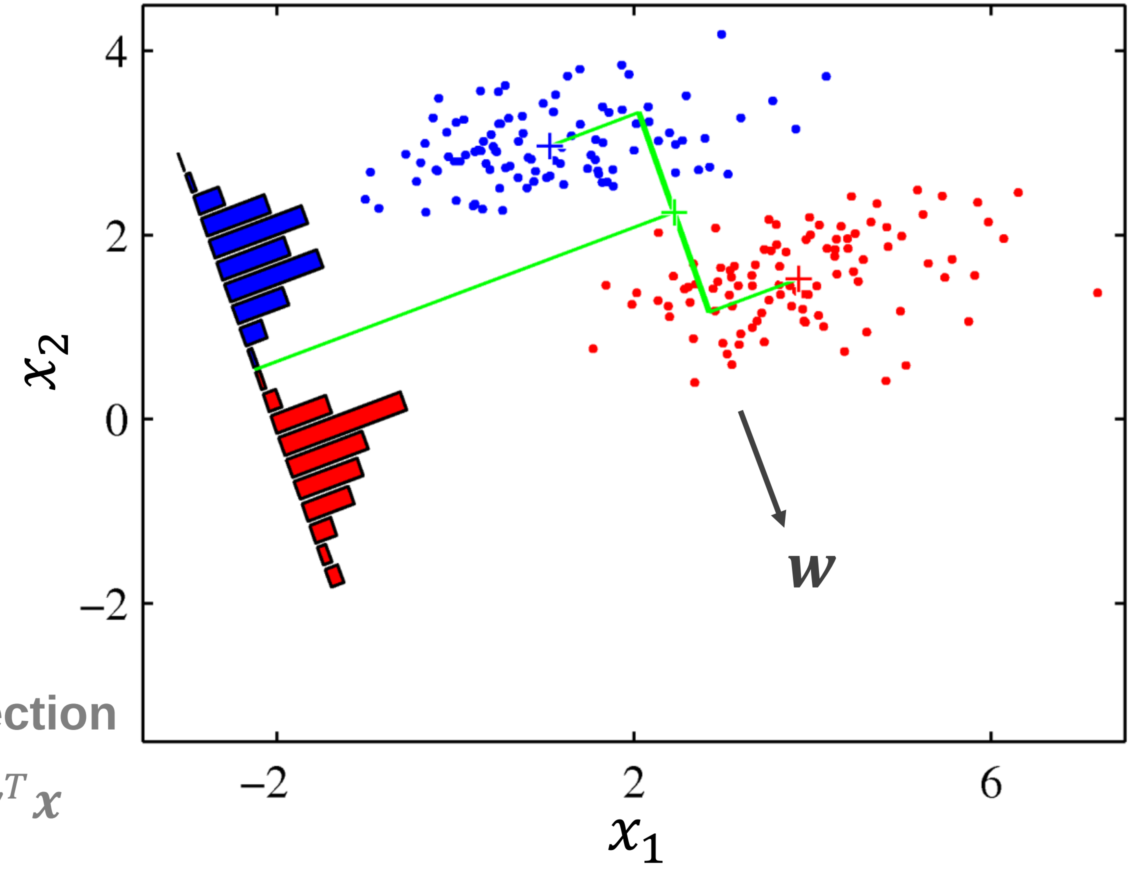
Fisher's Linear Discriminant

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

Projection onto line connecting the means



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

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

Consider $\boldsymbol{w}$ is a unit vector of parameters

2. We then classify the data in this space

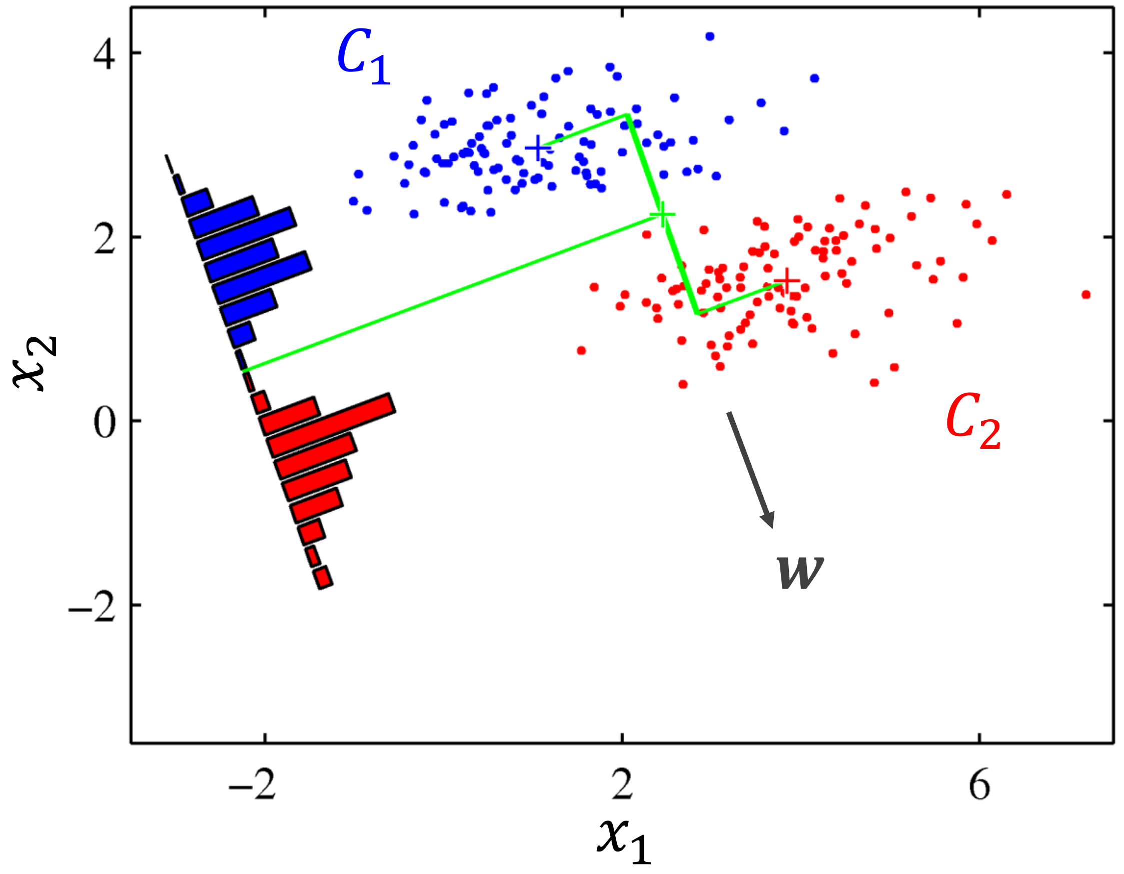
Similar to PCA, but accounts for class separability

Our decision rule becomes:

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

else Class 2

FLD: how do we choose the vector w?

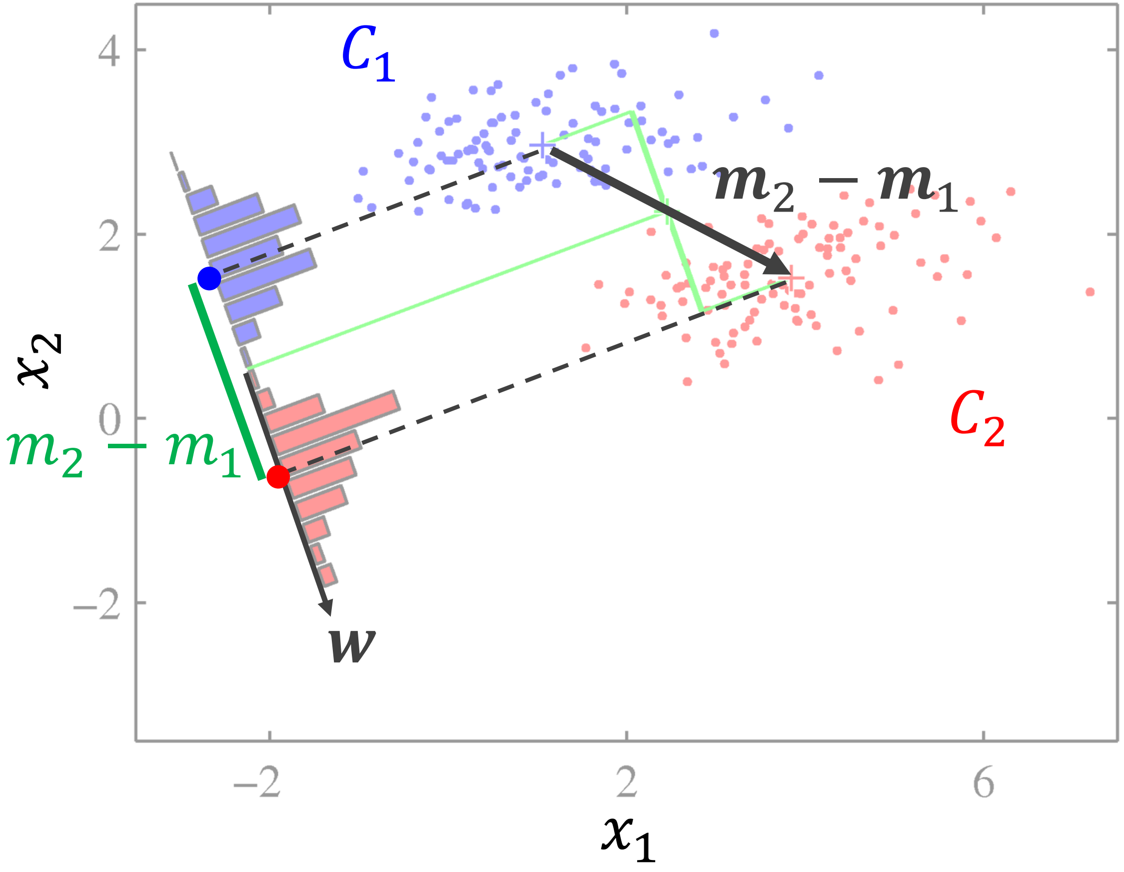


$$y = \hat{f}(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$$

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**

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

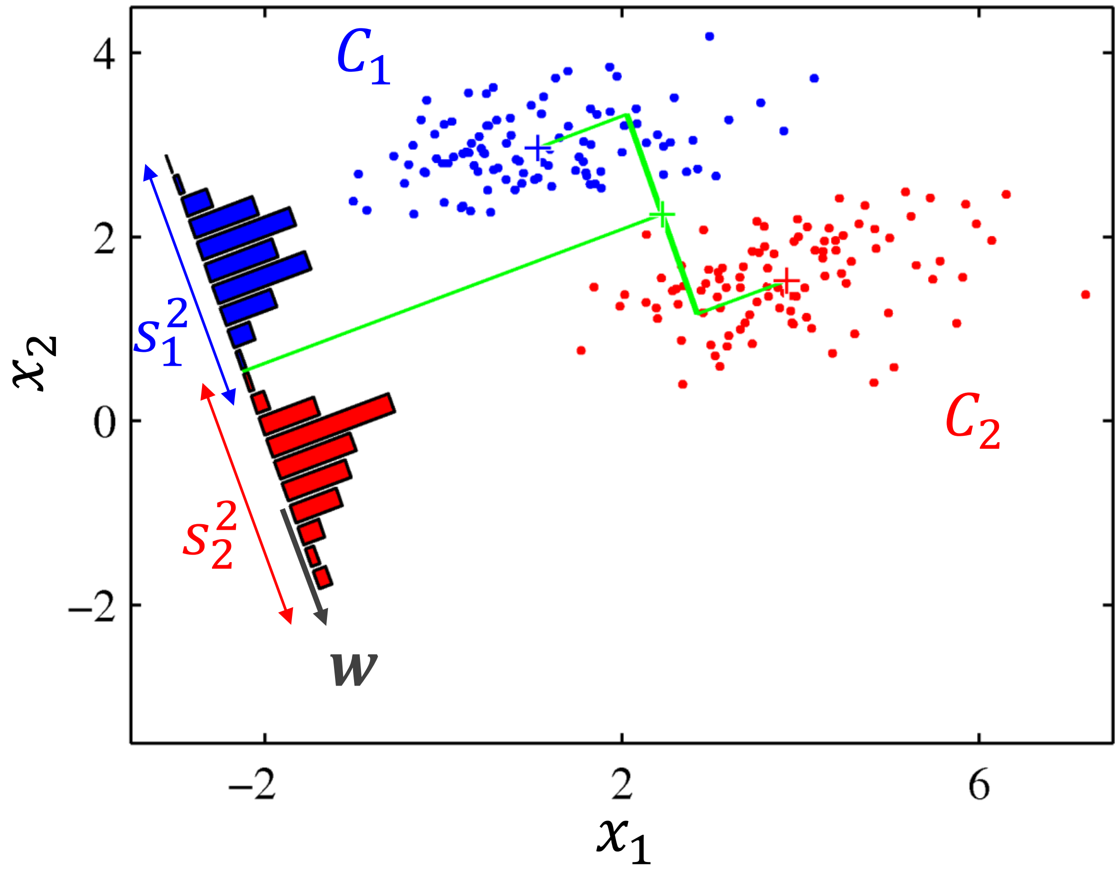
mean of class 1 — mean of class 2

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 $\boldsymbol{w}$?



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

Decrease the **variance** within each class

The "scatter" of the **projected** data:

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

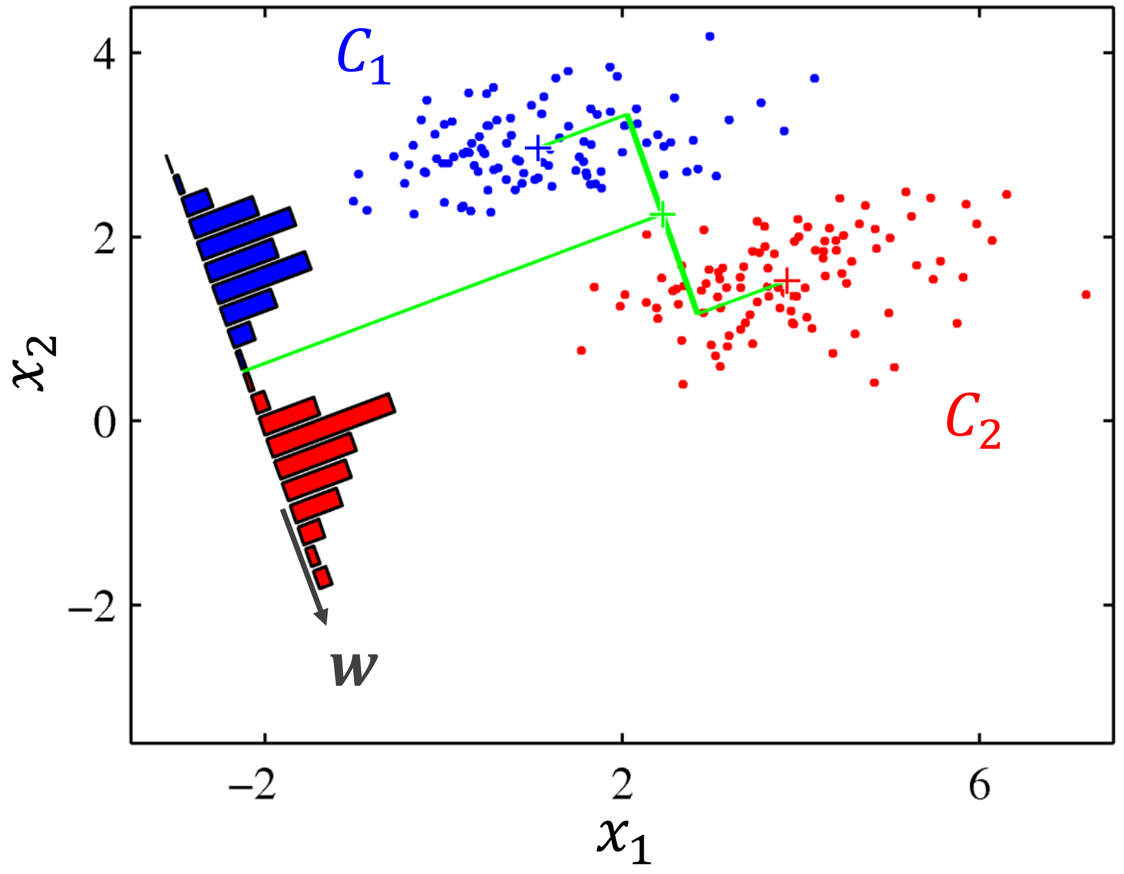
where $m_k = \boldsymbol{w}^T \boldsymbol{m}_k$

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

Therefore the total within-class scatter:

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

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



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

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

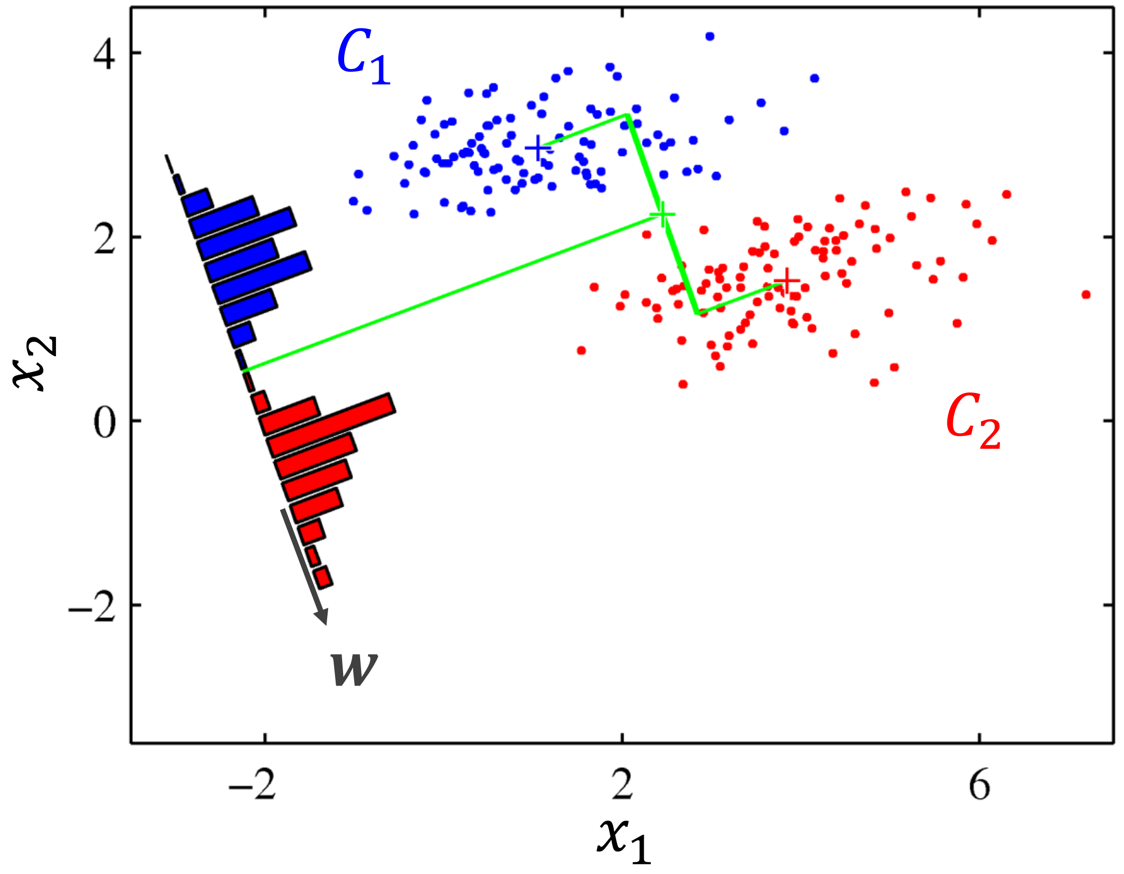
$$S = s_1^2 + s_2^2$$

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 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 $\boldsymbol{w}$

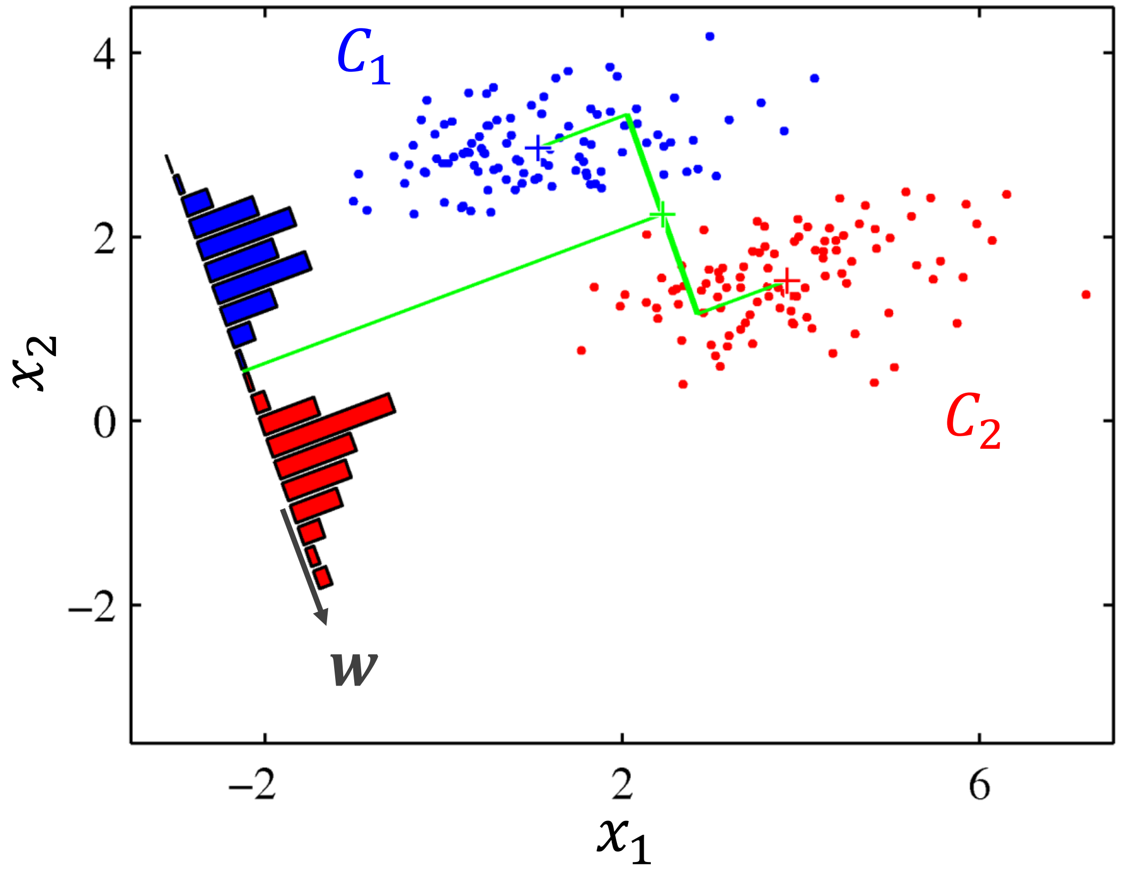
(see appendix slides for full derivation)

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

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

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

Fisher's Linear Discriminant



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

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)

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

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

Bayes rule in the context of classification

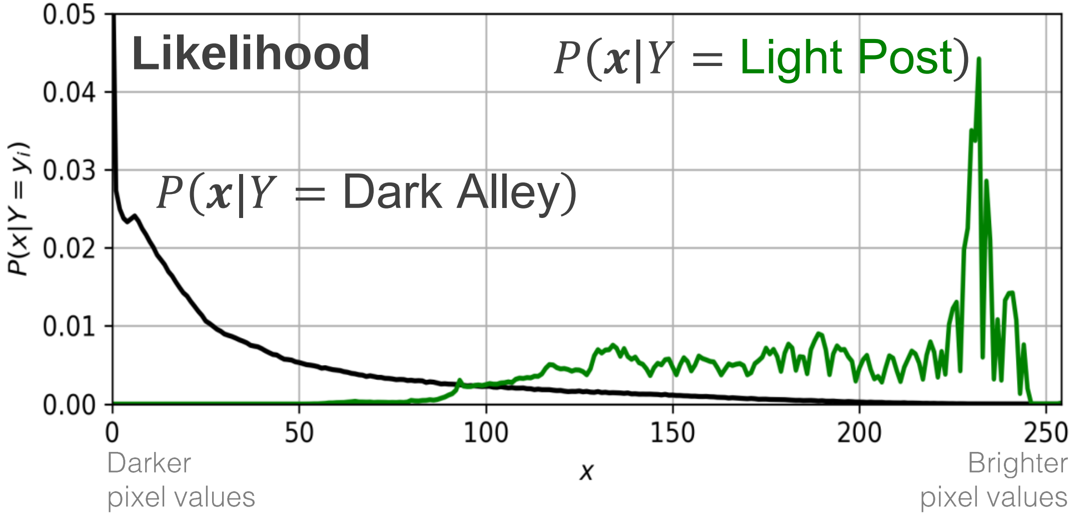
Class 1: Light Post

Class 0: Dark Alley

Randomly draw a pixel from either of the images: $x_i = \boxed{149}$

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

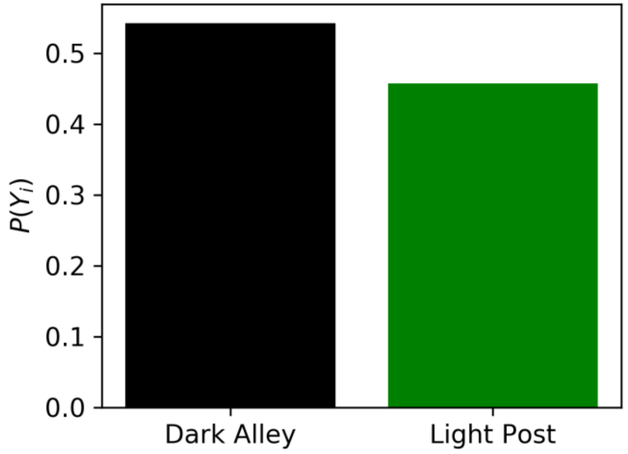
How do we determine which image it was most likely to have come from?



Class 1: Light Post y_1

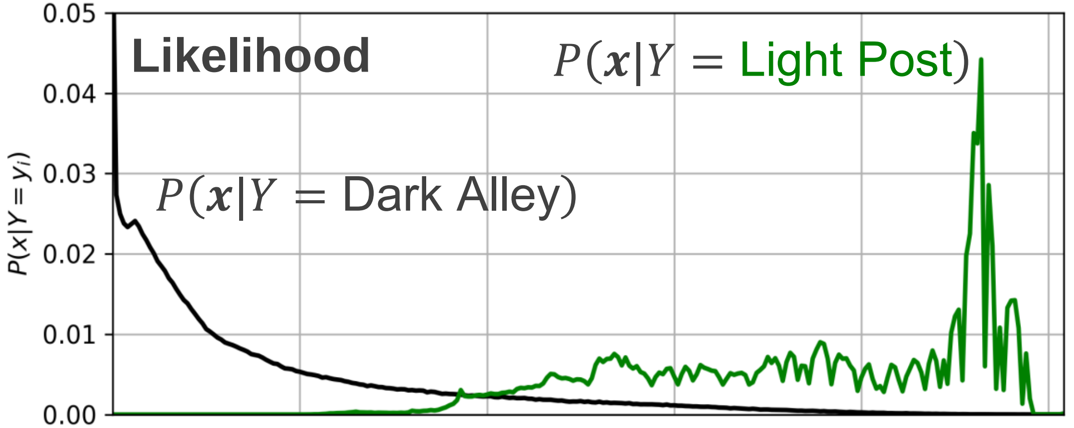
Class 0: Dark Alley y_0

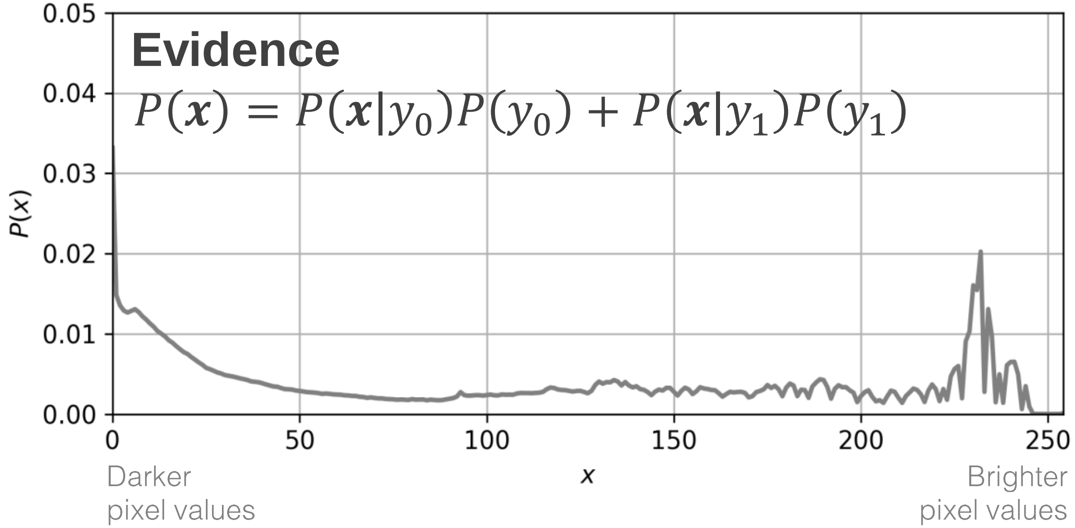
Prior: $P(Y = y_i)$



Bayes' Rule

$$\underbrace{P(Y = y_i|\boldsymbol{x})}_{\text{Posterior}} = \frac{\overbrace{P(\boldsymbol{x}|Y = y_i)}^{\text{Likelihood}}\overbrace{P(Y = y_i)}^{\text{Prior}}}{\underbrace{P(\boldsymbol{x})}_{\text{Evidence}}}$$

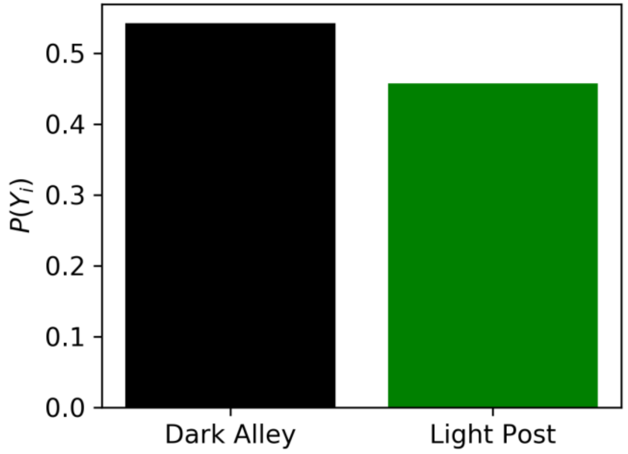




Class 1: Light Post y_1

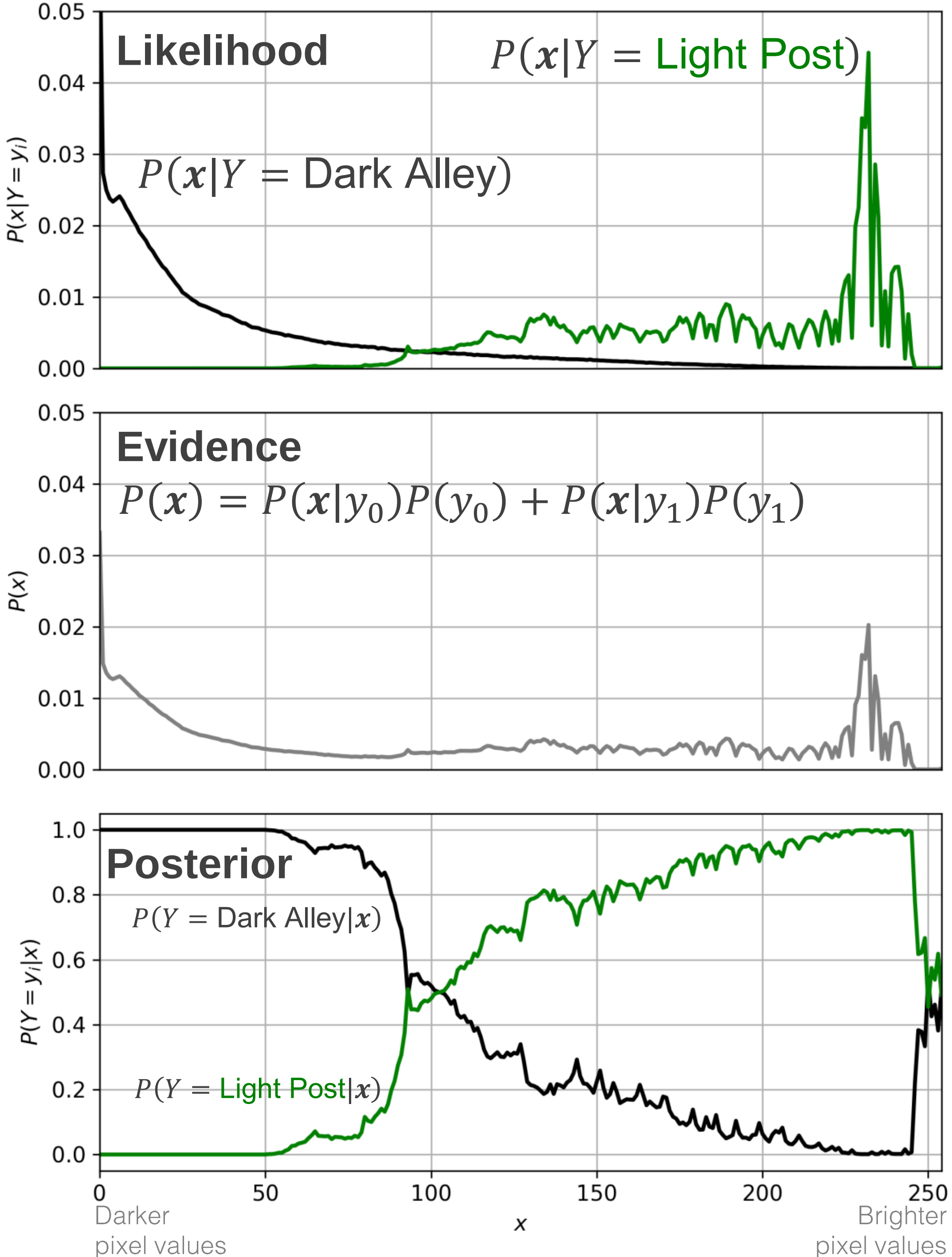
Class 0: Dark Alley y_0

Prior: $P(Y = y_i)$



Bayes' Rule

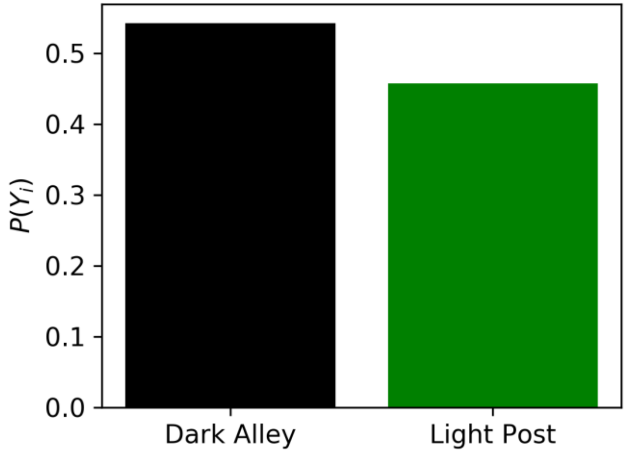
$$\underbrace{P(Y = y_i|\boldsymbol{x})}_{\text{Posterior}} = \frac{\overbrace{P(\boldsymbol{x}|Y = y_i)}^{\text{Likelihood}}\overbrace{P(Y = y_i)}^{\text{Prior}}}{\underbrace{P(\boldsymbol{x})}_{\text{Evidence}}}$$



Class 1: Light Post y_1

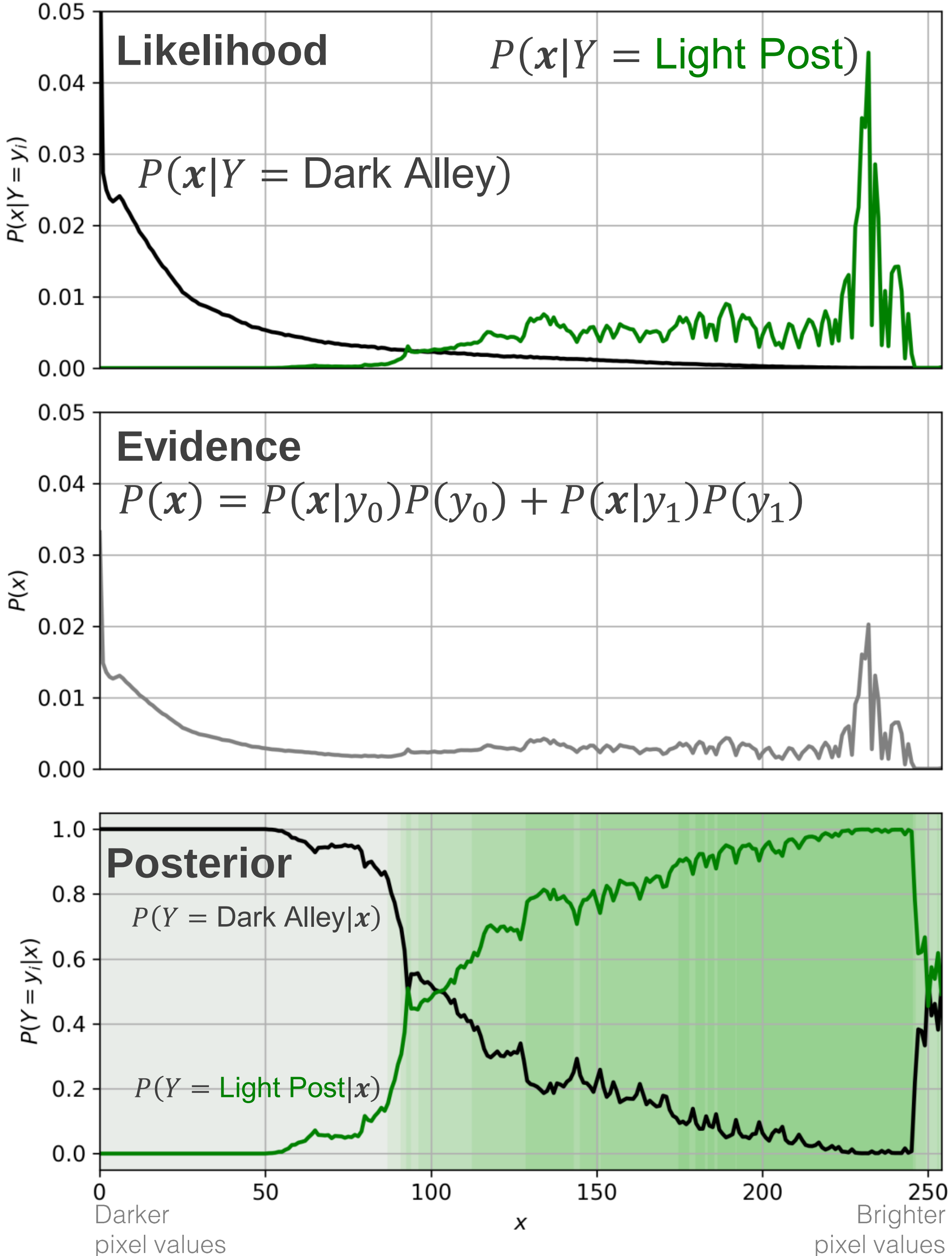
Class 0: Dark Alley y_0

Prior: $P(Y = y_i)$



Bayes' Rule

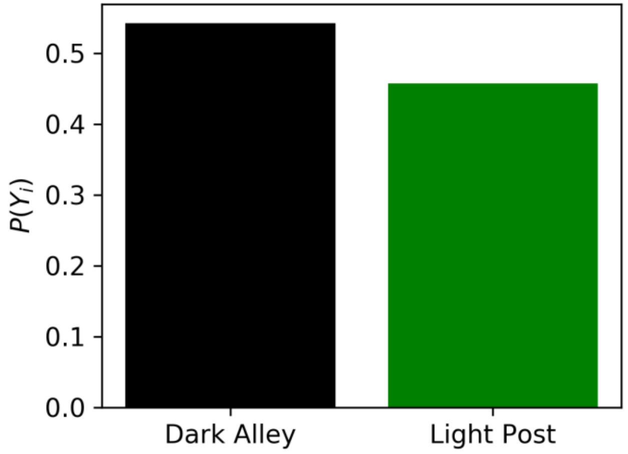
$$\underbrace{P(Y = y_i|\boldsymbol{x})}_{\text{Posterior}} = \frac{\overbrace{P(\boldsymbol{x}|Y = y_i)}^{\text{Likelihood}}\overbrace{P(Y = y_i)}^{\text{Prior}}}{\underbrace{P(\boldsymbol{x})}_{\text{Evidence}}}$$



Class 1: Light Post y_1

Class 0: Dark Alley y_0

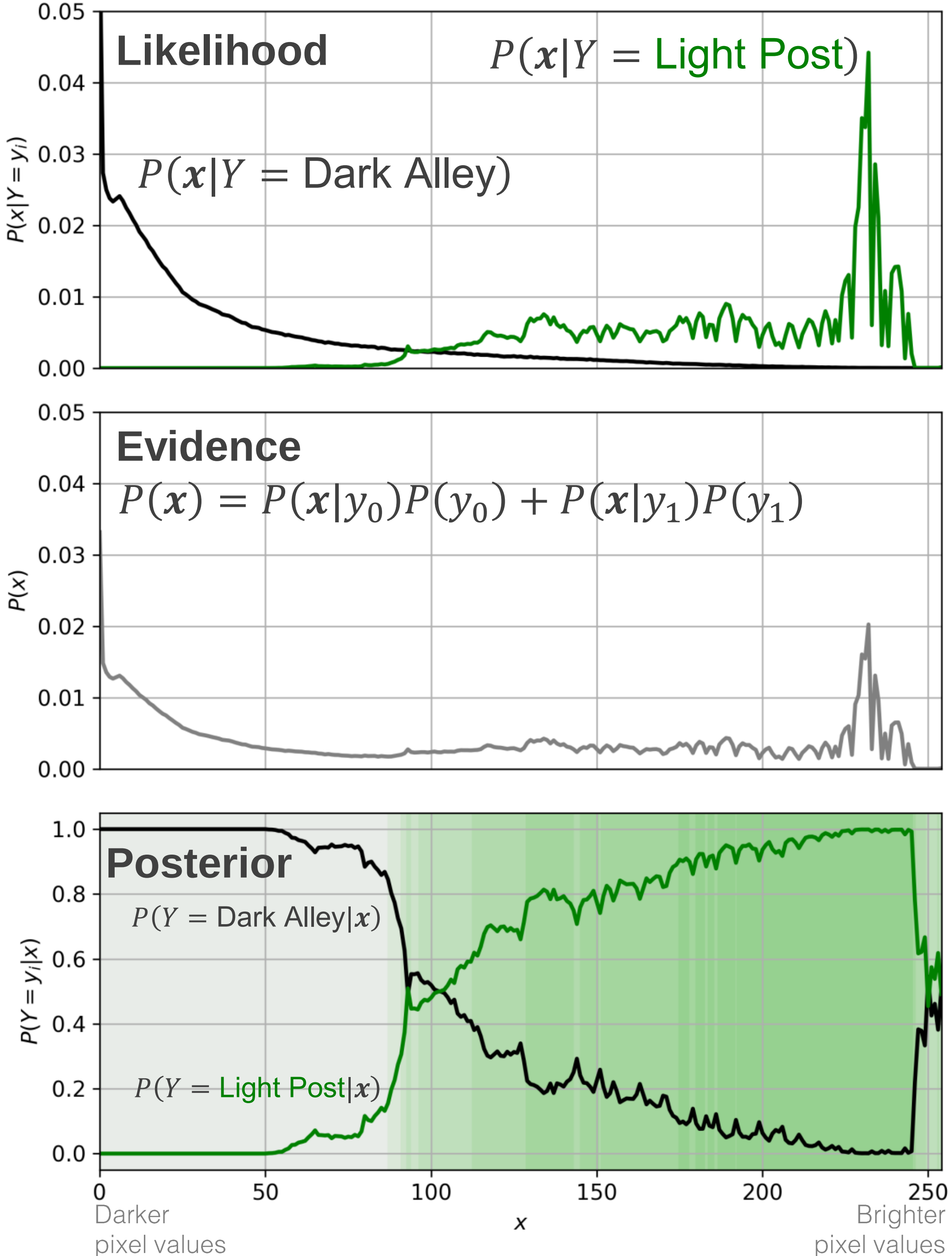
Prior: $P(Y = y_i)$



Decision rule:

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

else Dark Alley

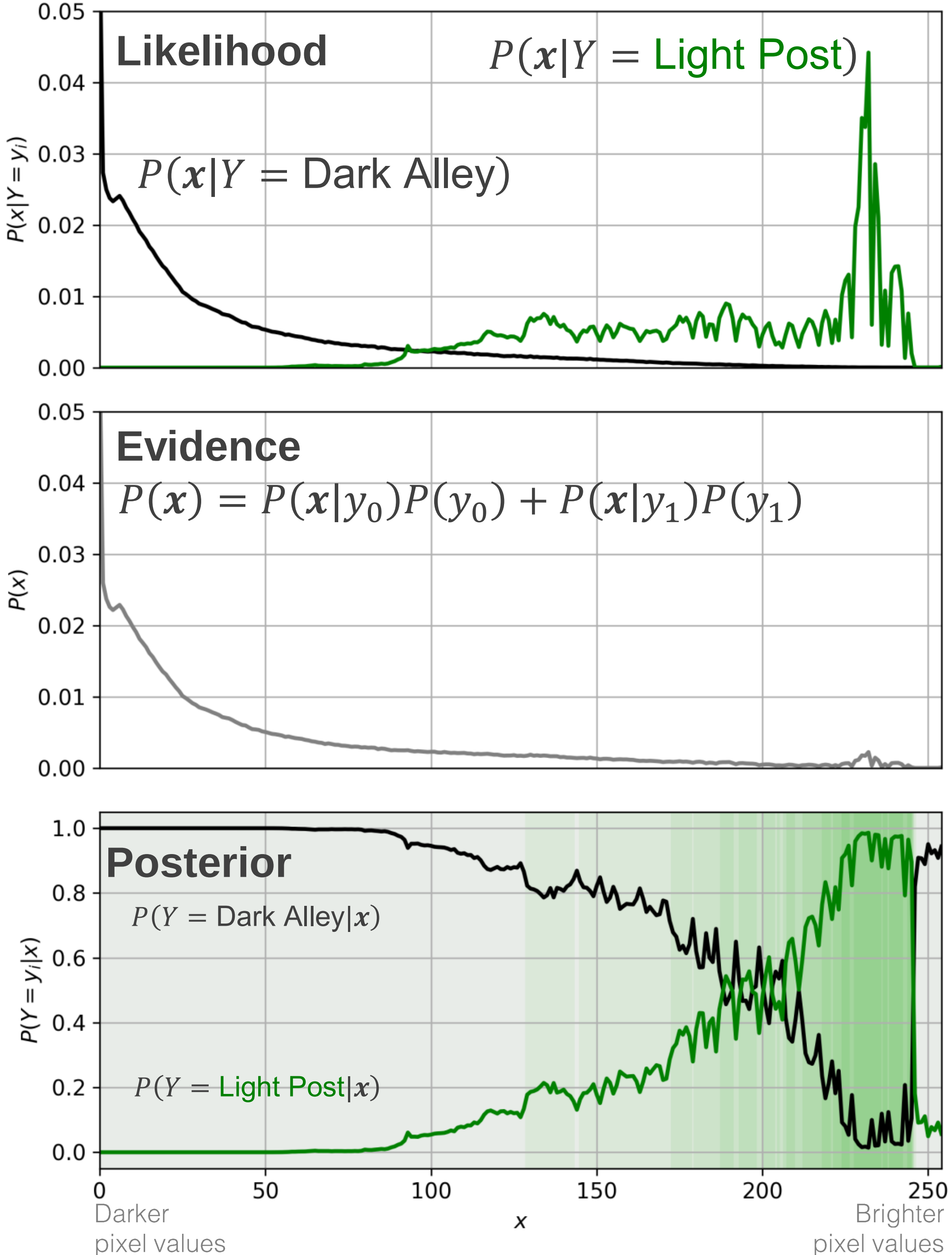


Classifying each of the individual pixels as being either from **Light Post** or **Dark Alley** results in classification above

Decision rule:

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

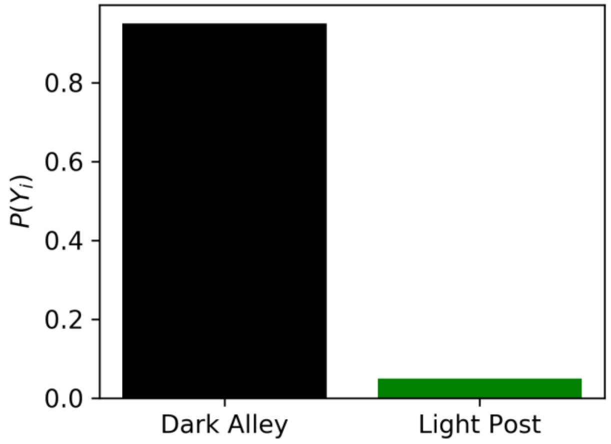
else Dark Alley



Class 1: Light Post y_1

Class 0: Dark Alley y_0

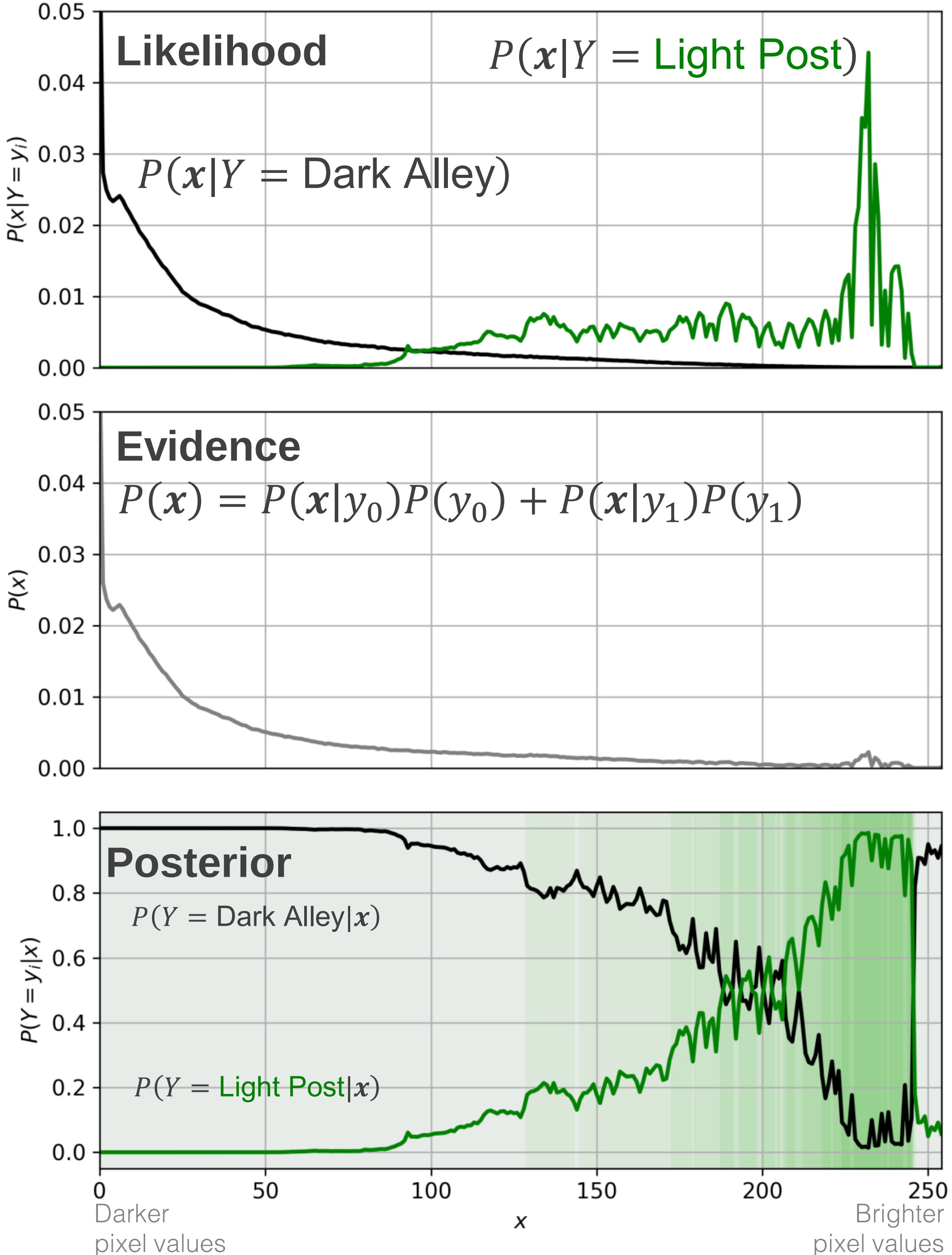
Prior: $P(Y = y_i)$



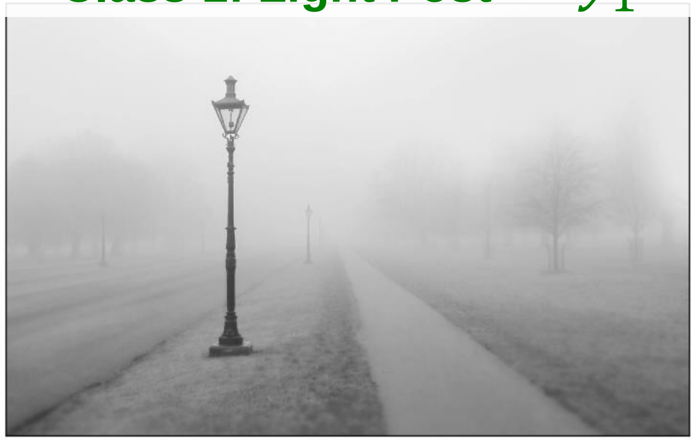
Let's assume we had reason to believe that the random sampling of pixels favored the **Dark Alley**

Bayes' Rule

$$\underbrace{P(Y = y_i|\boldsymbol{x})}_{\text{Posterior}} = \frac{\overbrace{P(\boldsymbol{x}|Y = y_i)}^{\text{Likelihood}}\overbrace{P(Y = y_i)}^{\text{Prior}}}{\underbrace{P(\boldsymbol{x})}_{\text{Evidence}}}$$



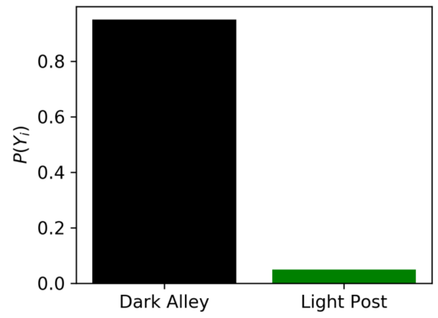
Class 1: Light Post y_1



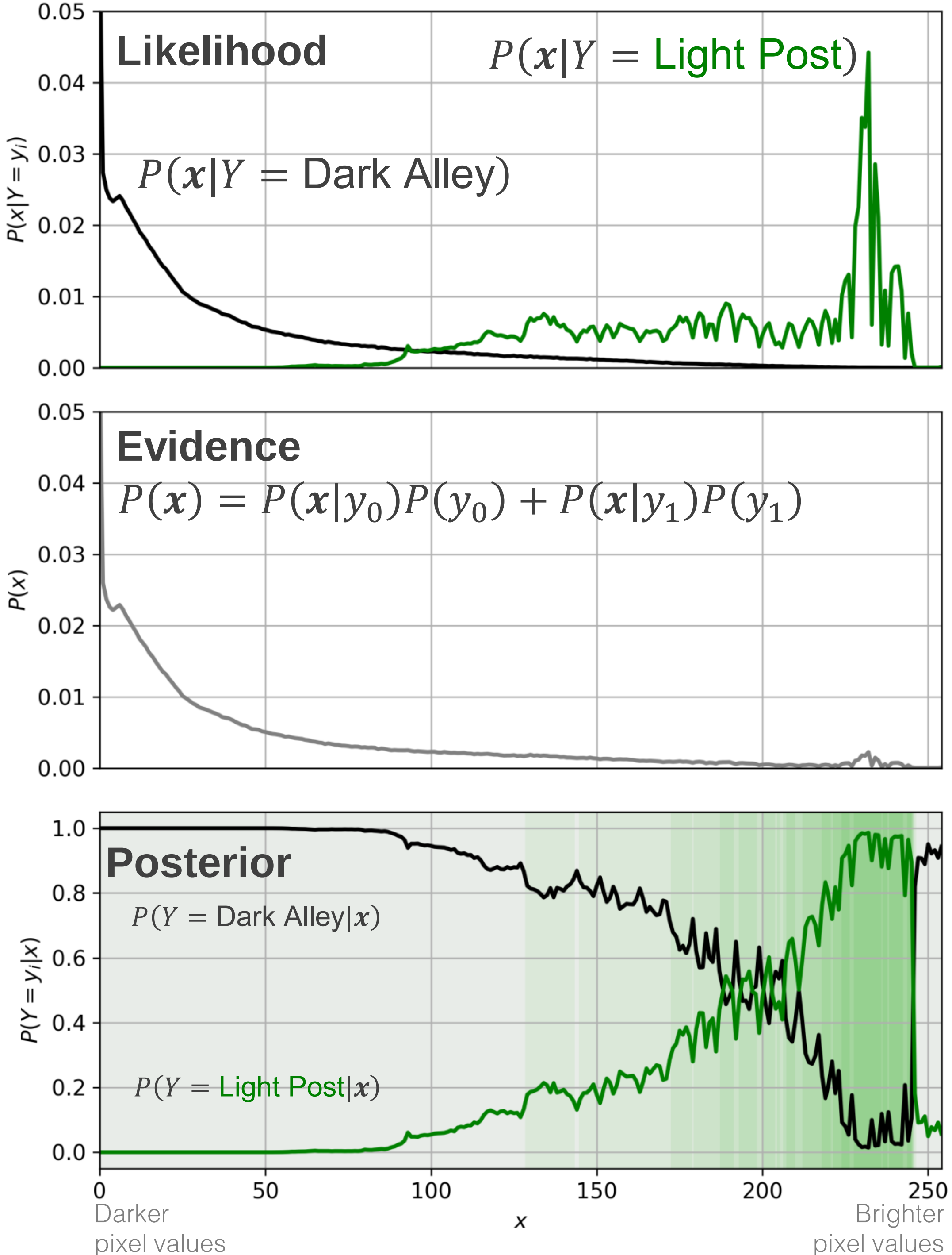
Class 0: Dark Alley y_0

Green = classified as from Light Post
Grey = classified as from Dark Alley

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

$$\underbrace{P(Y = y_i|\boldsymbol{x})}_{\text{Posterior}} = \frac{\overbrace{P(\boldsymbol{x}|Y = y_i)}^{\text{Likelihood}}\overbrace{P(Y = y_i)}^{\text{Prior}}}{\underbrace{P(\boldsymbol{x})}_{\text{Evidence}}}$$

Discriminative models model the **posterior**

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

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

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

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

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

$\boldsymbol{\Sigma}_i =$ covariance matrix for class i

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

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

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

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

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

$$\alpha (\boldsymbol{m}_2 - \boldsymbol{m}_1) \underbrace{(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T \boldsymbol{w}}_{\text{scalar } m_2 - m_1 \text{, call this } \gamma} = \beta \boldsymbol{S}_W \boldsymbol{w}$$

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

Property of matrix transposition:

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

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

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