Generative Models for Classification

Supervised learning in practice

Preprocessing
Explore & prepare data

Data Visualization and Exploration

Identify patterns that can be leveraged for learning

Data Cleaning

- Missing data
- Noisy data
- Erroneous data

Scaling (Standardization)

Prepare data for use in scale-dependent algorithms.

Feature Extraction

Dimensionality reduction eliminates redundant information

Model training

1 Supervised Learning Models: Linear models and KNN
(enough to get started using supervised learning)

Select model options

5 Other algorithms and concepts:
- **Generative vs discriminative models**
- Parametric vs nonparametric models
- Model ensembles
- Feature/representation learning (neural networks, deep learning)

4 How to control model overfit: regularization strategies for model refinement

Iteratively fine tune the model

Performance evaluation

Make a prediction on validation data

2 Evaluating model performance and comparing models

Classification

3 How to make decisions using models

Regression

MSE, explained variance, R^2

Classifiers

Covered so far

K-Nearest Neighbors

Perceptron

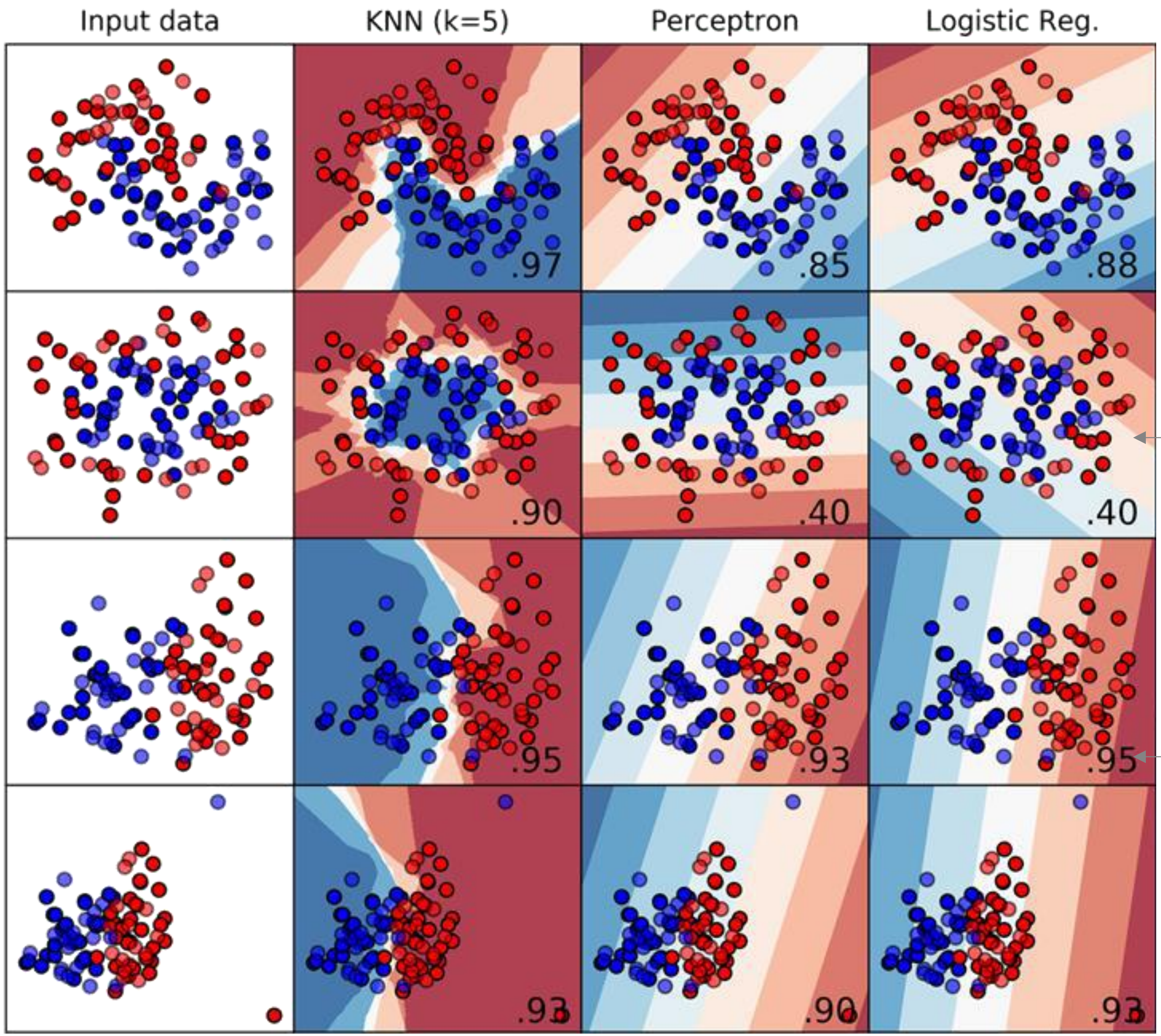
Logistic Regression

Linear Discriminant Analysis

Quadratic Discriminant Analysis

Naïve Bayes

Along the way…
Revisiting Bayes’ Rule
Projections from higher dimensions
Multivariate normal distributions



Comparison of classifiers we have seen so far

The color gradient shows the confidence scores

Test data accuracy

Bayes' rule in the context of classification

Bayes' Rule

Likelihood Prior

$$P(C|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|C)P(C)}{P(\boldsymbol{X})}$$

Posterior

Evidence

$\boldsymbol{X}$ Features

C Class label

i.e. $C \in \{c_0, c_1\}$ for the binary case

Bayes' Decision Rule:

choose the most probable class given the data

If $P(C_i = c_1|\boldsymbol{X}_i) > P(C_i = c_0|\boldsymbol{X}_i)$ then $\hat{y} = c_1$

otherwise $\hat{y} = c_0$

- If the distributions are correct, this decision rule is **optimal**
- Rarely do we have enough information to use this in practice

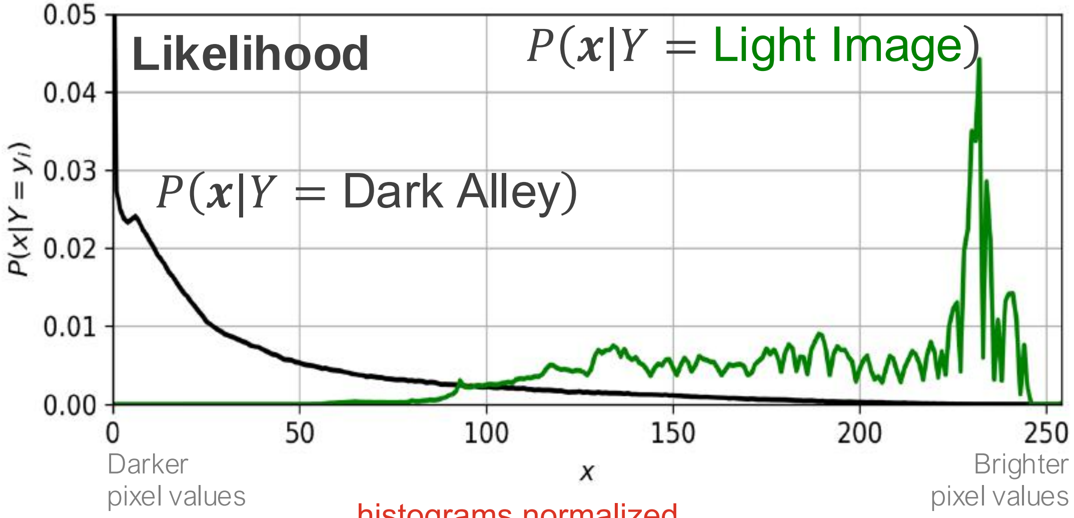
Class 1: Light Image

Class 0: Dark Image

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

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

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

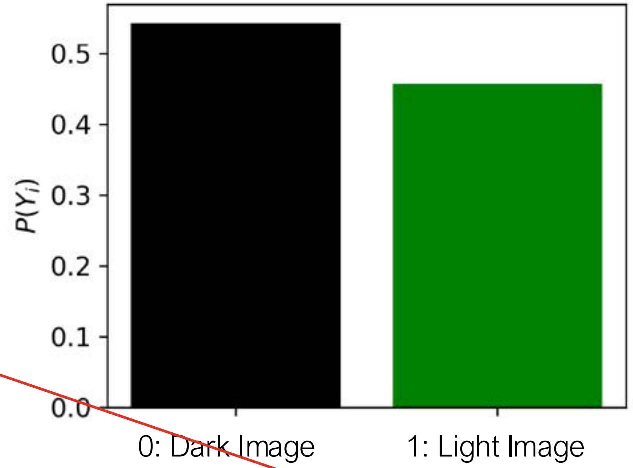


histograms normalized

Class 1: Light Image y_1

Class 0: Dark Image y_0

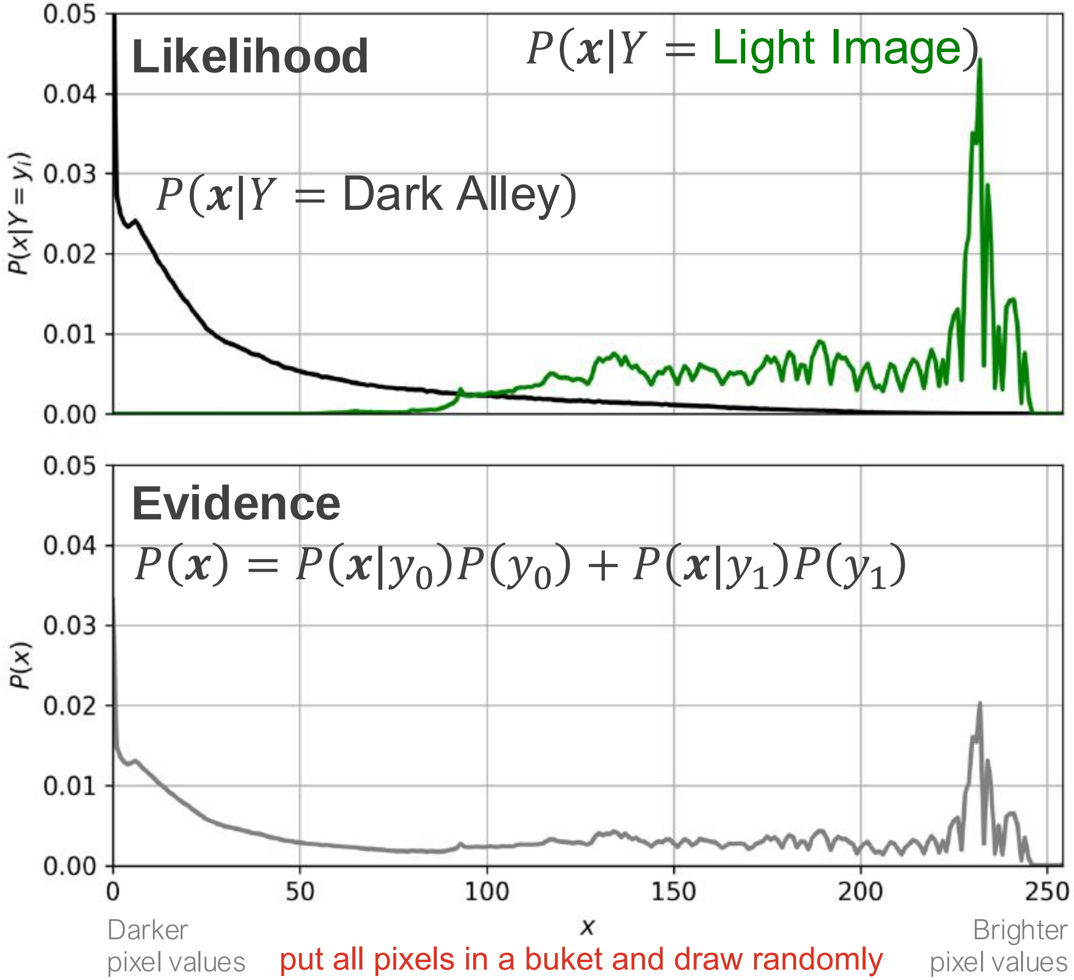
Prior: $P(Y = y_i)$



distribution of the pixel in the image

Bayes' Rule

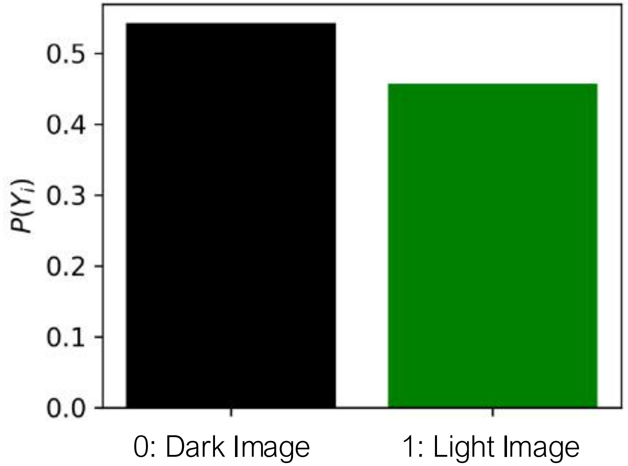
$$\underbrace{P(Y = y_i|\boldsymbol{x})}_{\text{Posterior}} = \frac{\overbrace{P(\boldsymbol{x}|Y = y_i)}^{\text{Likelihood}}\overbrace{P(Y = y_i)}^{\text{Prior}}}{\underbrace{P(\boldsymbol{x})}_{\text{Evidence}}}$$



Class 1: Light Image y_1

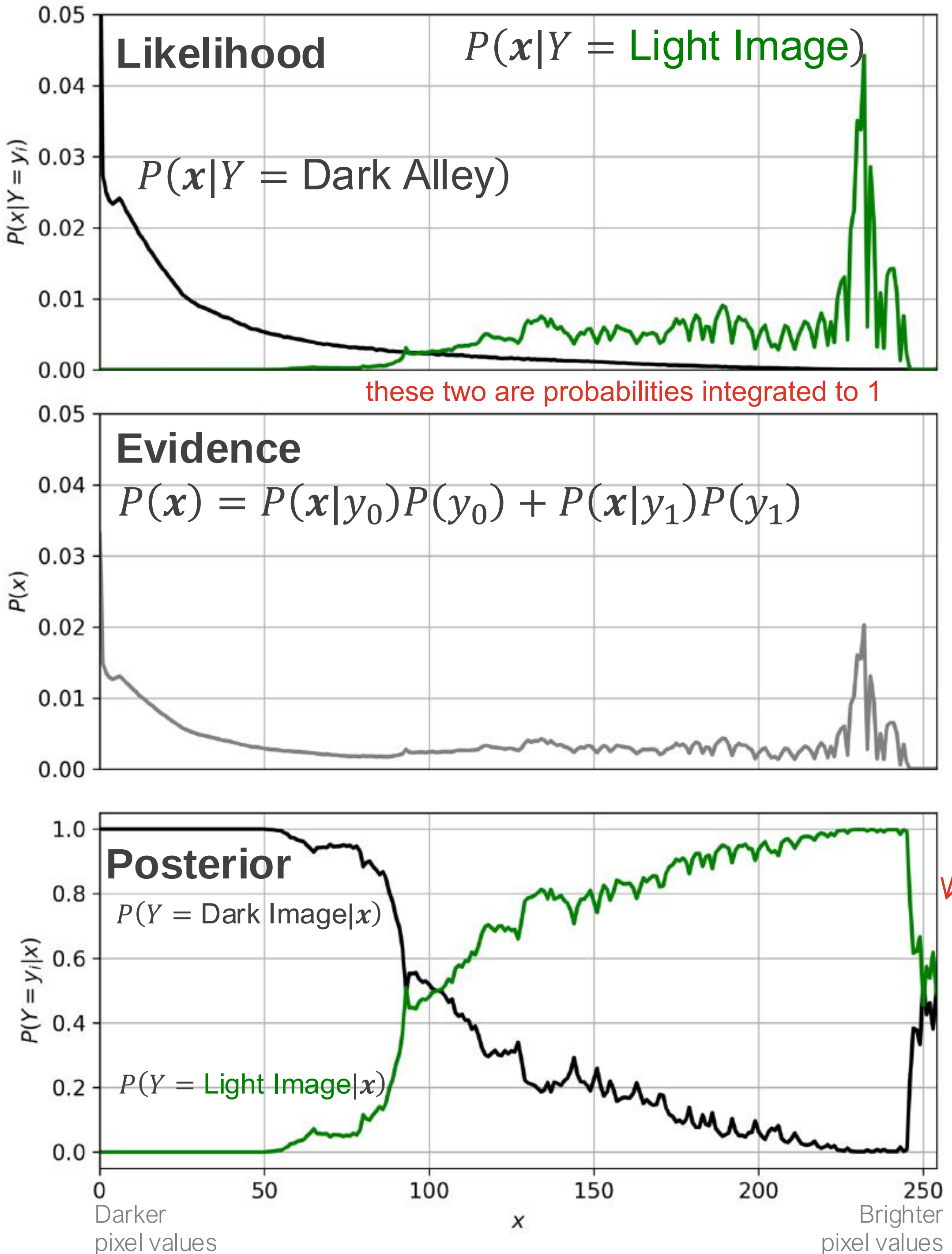
Class 0: Dark Image y_0

Prior: $P(Y = y_i)$



Bayes' Rule

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



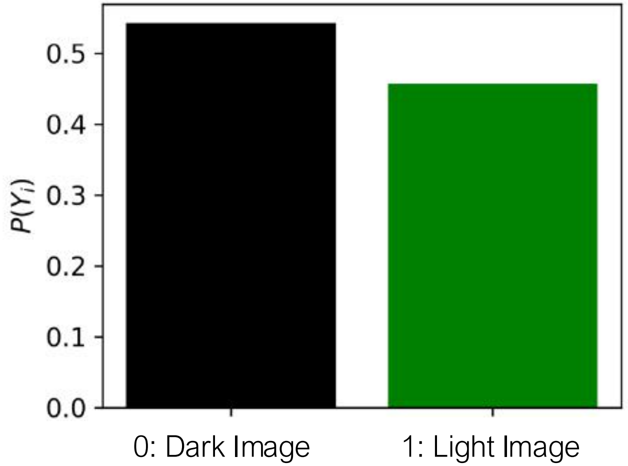
Class 1: Light Image y_1

Class 0: Dark Image y_0

integrate between two classes --> 1

integrate of one curve does not integrate to 1

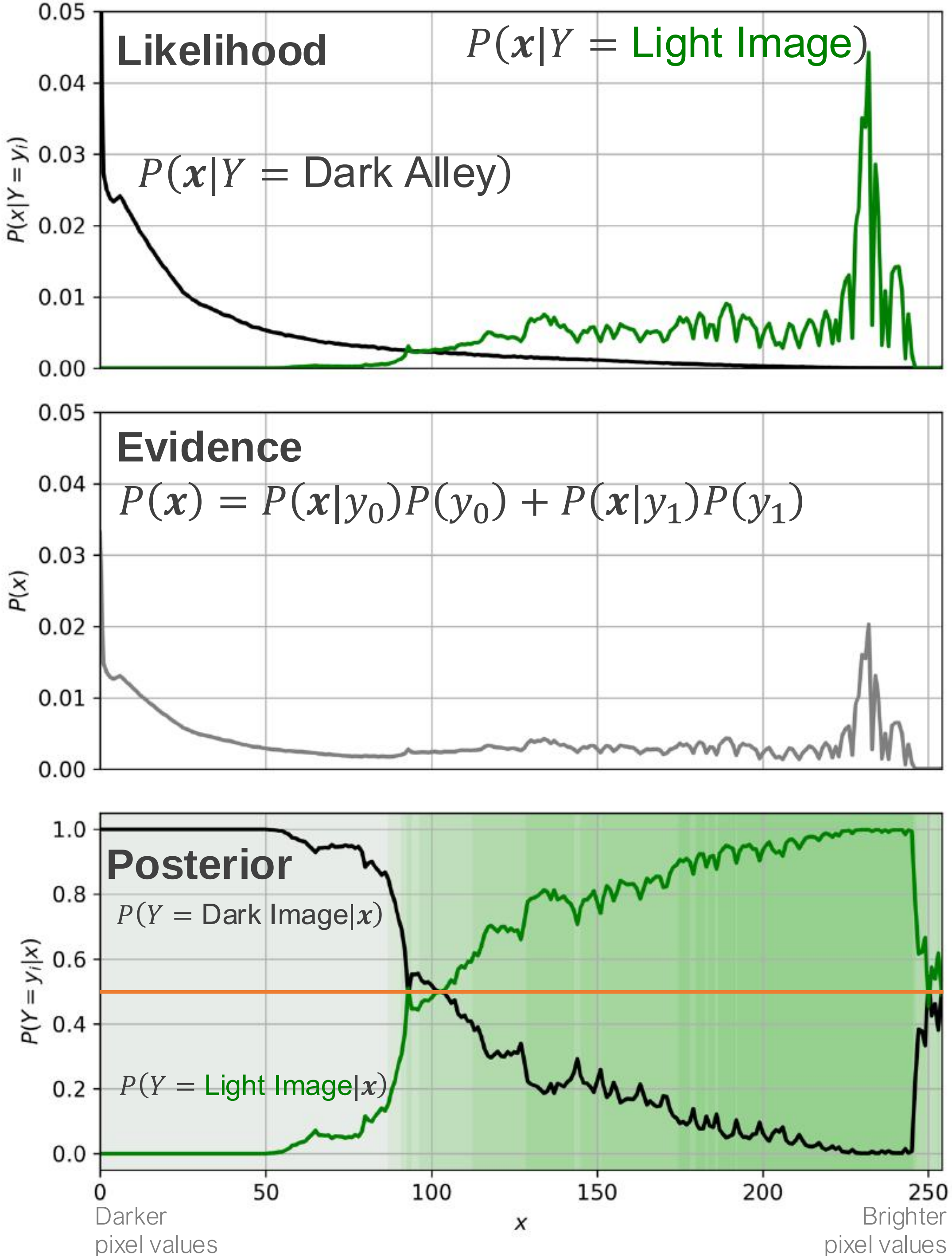
Prior: $P(Y = y_i)$



[integrate P(x) over x] = 1
[integrate P(y) over y] = 1

Bayes' Rule

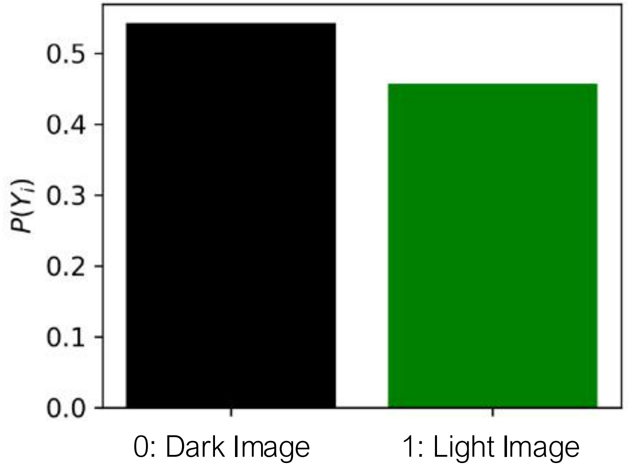
$$\underbrace{\boxed{P(Y} = y_i|\boldsymbol{x})}_{\text{Posterior}} = \frac{\underbrace{\boxed{P(\boldsymbol{x}}|Y = y_i)}_{\text{Likelihood}}\underbrace{\boxed{P(Y} = y_i)}_{\text{Prior}}}{\underbrace{\boxed{P(\boldsymbol{x})}}_{\text{Evidence}}}$$



Class 1: Light Image y_1

Class 0: Dark Image y_0

Prior: $P(Y = y_i)$

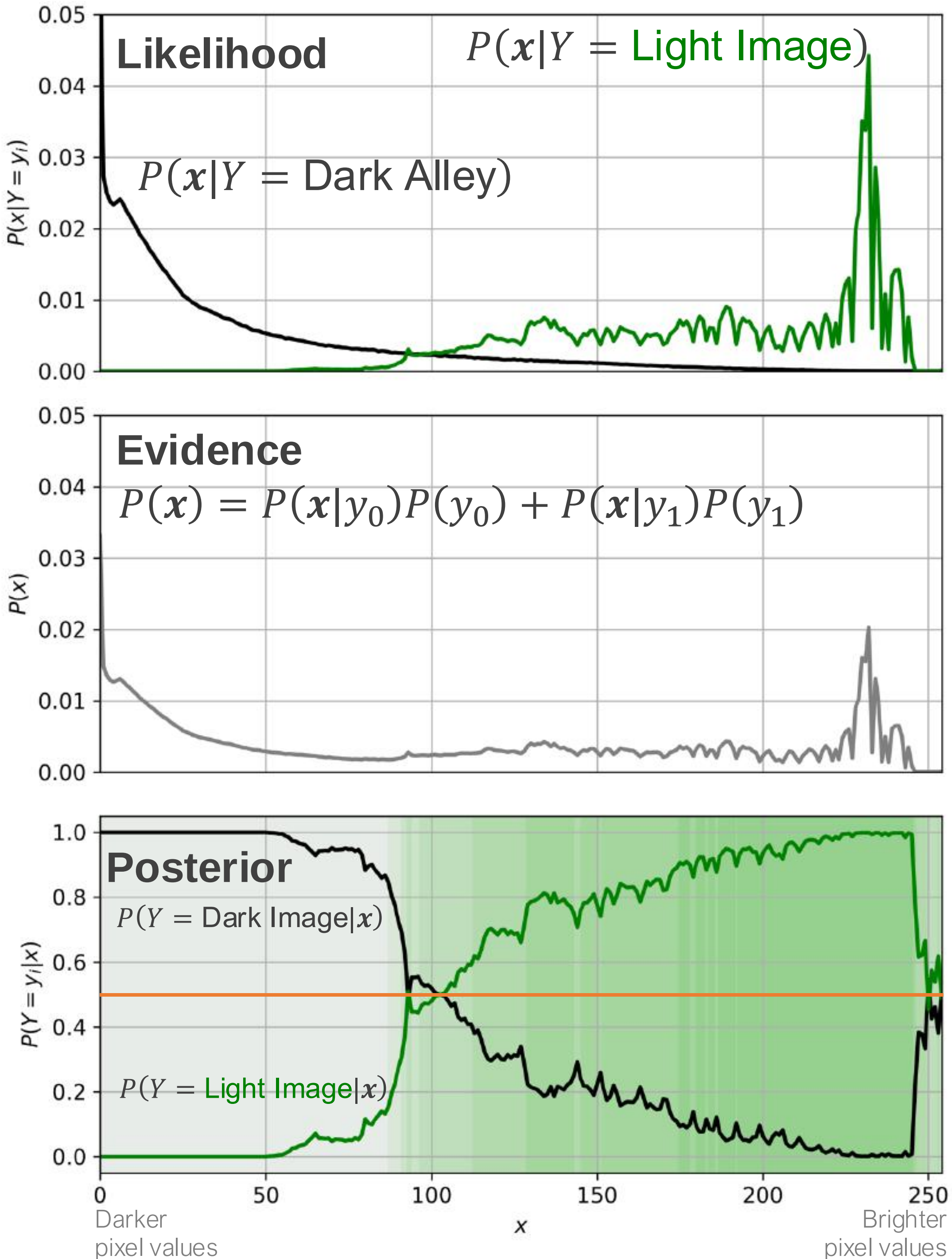


pick the bigger one

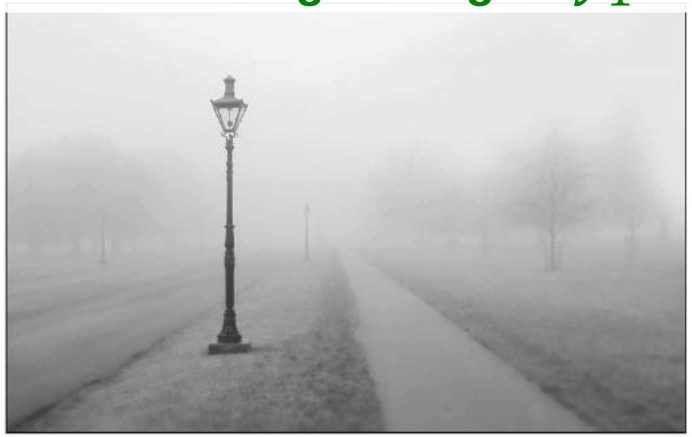
Decision rule:

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

else Dark Image



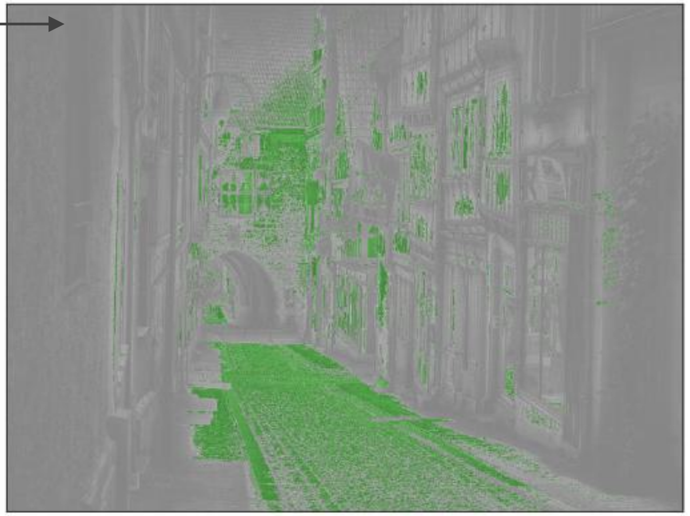
Class 1: Light Image y_1



Class 0: Dark Image y_0

Green = classified as from Light Image

Grey = classified as from Dark Image

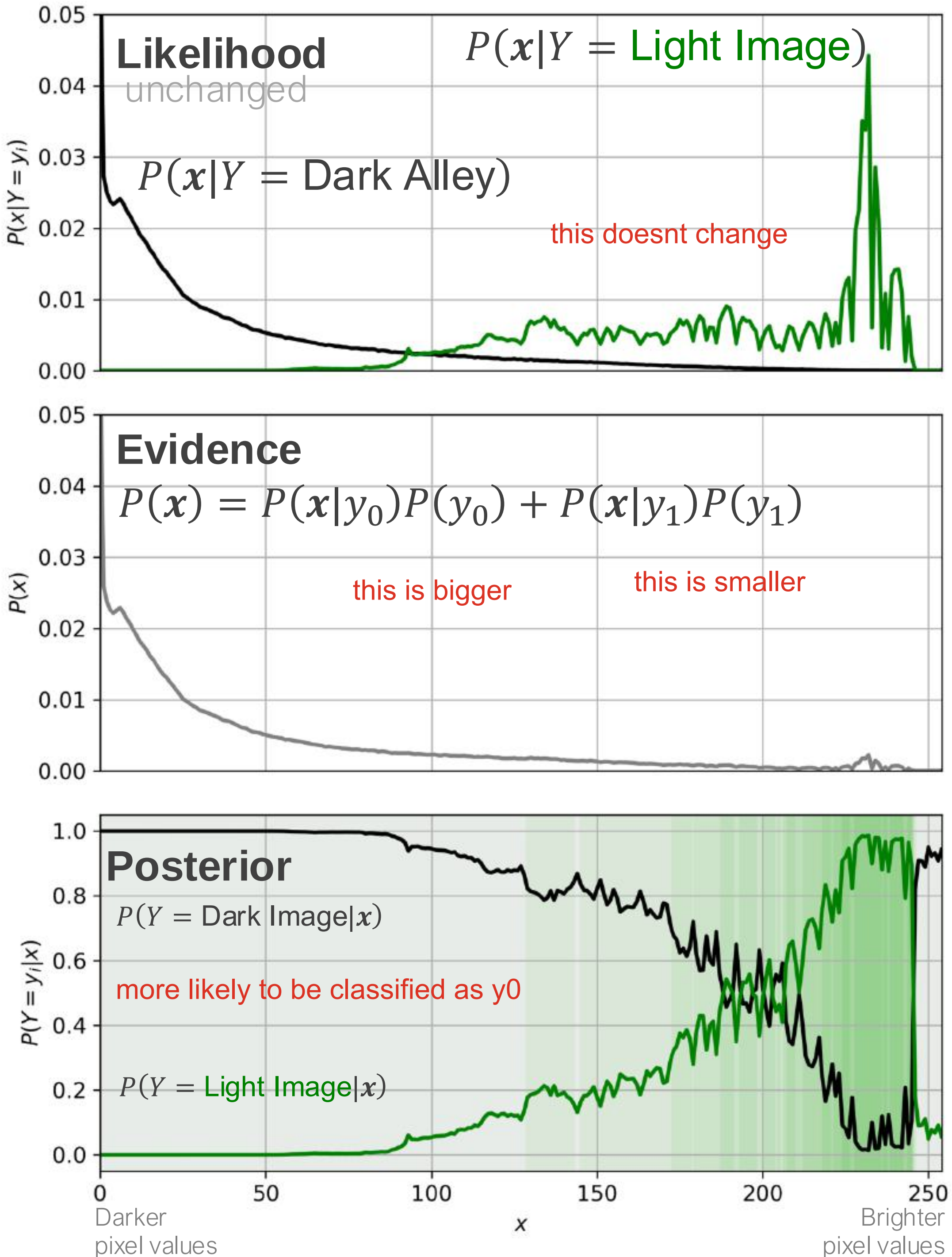


Classifying each of the individual pixels as being either from **Light Image** or **Dark Image** results in classification above

Decision rule:

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

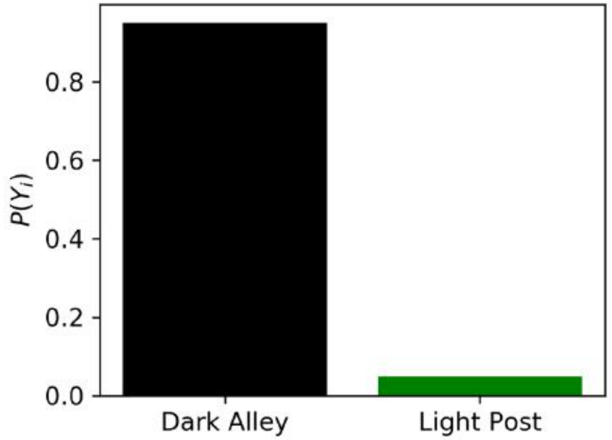
else Dark Image



Class 1: Light Image y_1

Class 0: Dark Image y_0

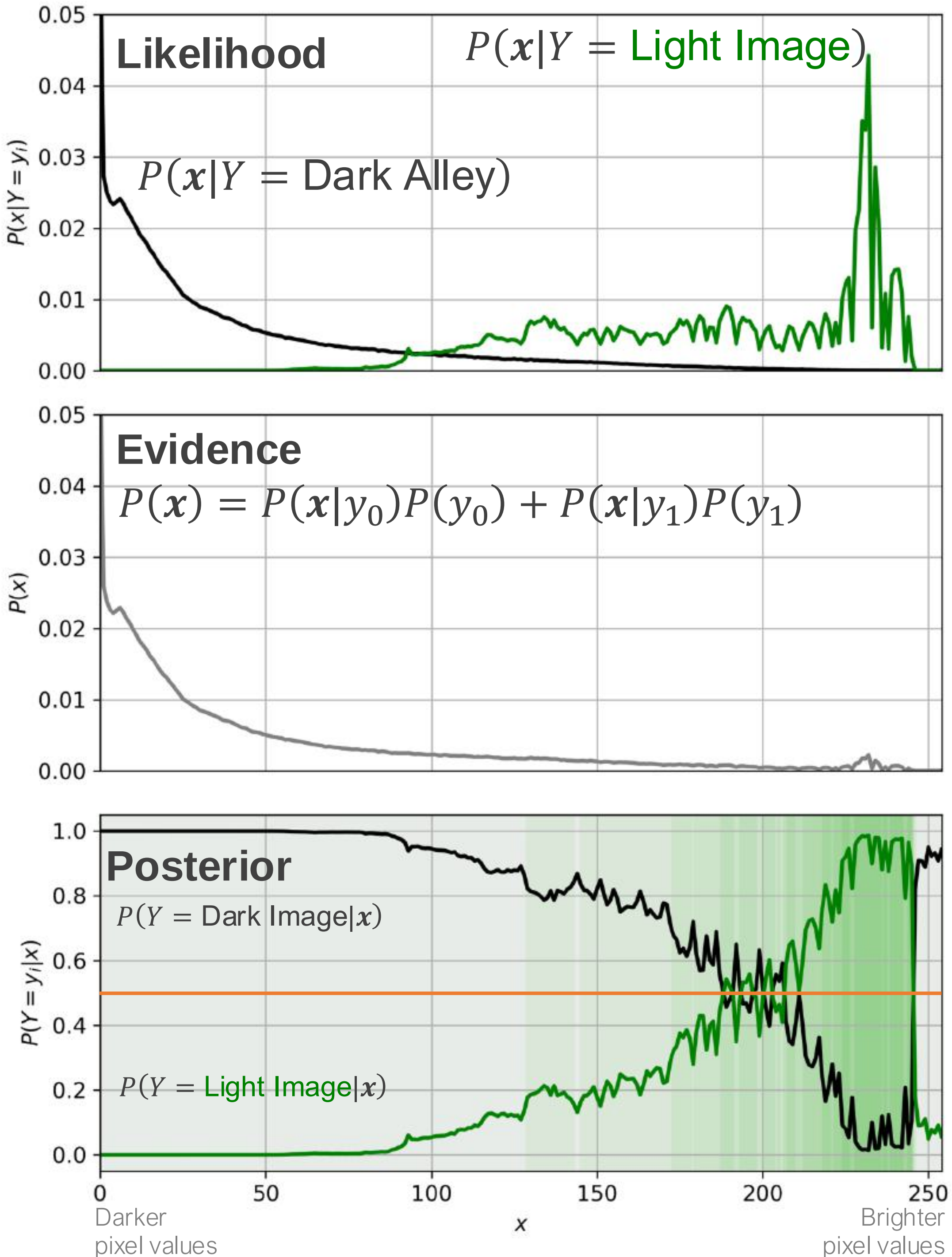
Prior: $P(Y = y_i)$



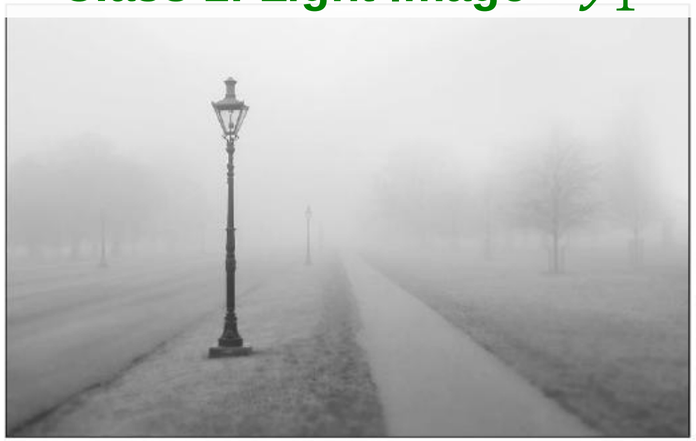
Let's assume the sampling of pixels occurred more from the **Dark Image**

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 Image y_1

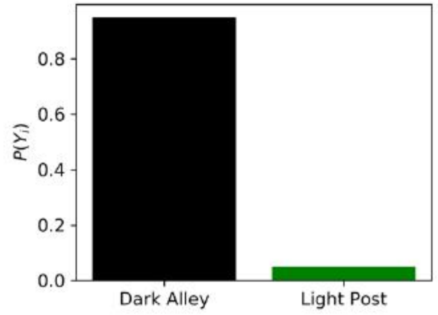


Class 0: Dark Image y_0

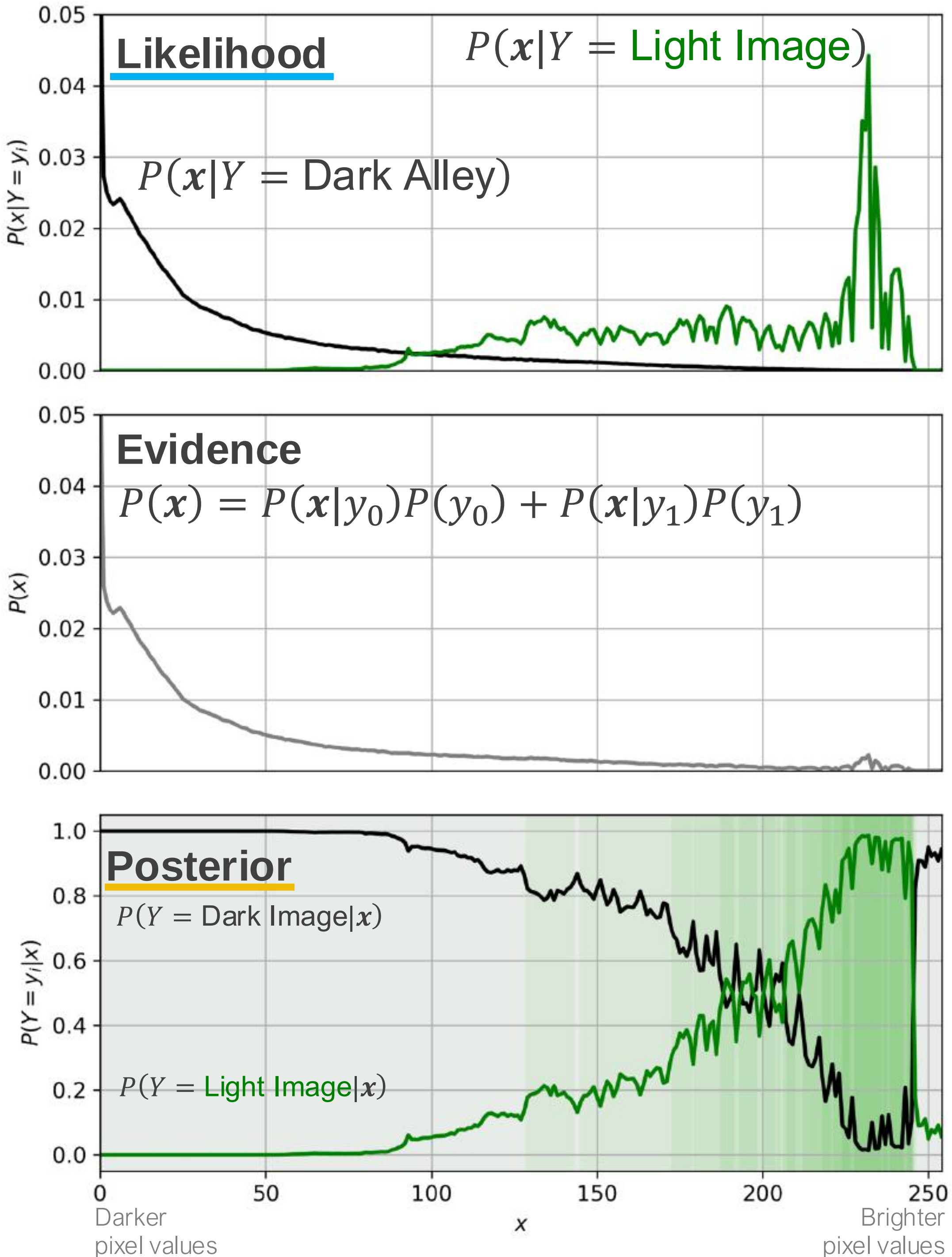
Green = classified as from Light Image

Grey = classified as from Dark Image

Prior: $P(Y = y_i)$



Assuming we the sampling of pixels occurred more from the **Dark Image**



Generative models model the **likelihood**

- These can also be used to generate synthetic data
- Often good performance when sample size is small

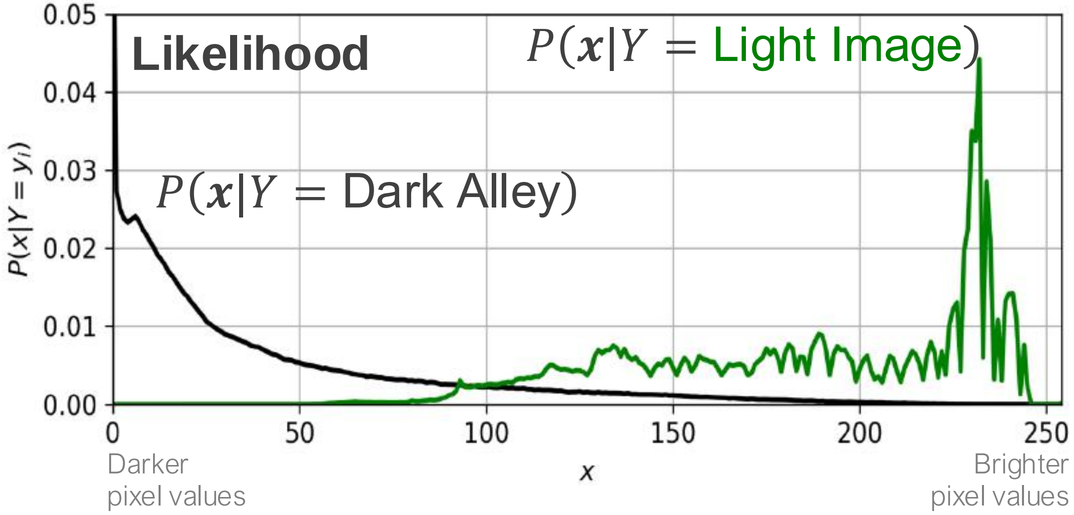
Examples: linear discriminant analysis, naïve Bayes, hidden Markov models, Guassian mixture models, Generative Adversarial Networks

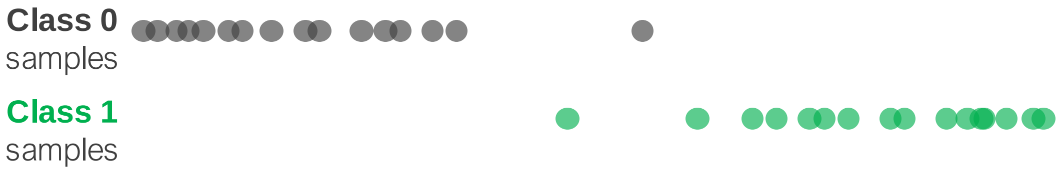
$$\underbrace{P(Y = y_i|\boldsymbol{x})}_{\text{Posterior}} = \frac{\overbrace{P(\boldsymbol{x}|Y = y_i)}^{\text{Likelihood}}\overbrace{P(Y = y_i)}^{\text{Prior}}}{\underbrace{P(\boldsymbol{x})}_{\text{Evidence}}}$$

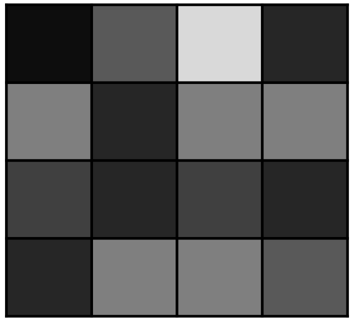
Discriminative models model the **posterior**

- Or they just directly estimate labels without a probabilistic interpretation, $f(\boldsymbol{x}) \rightarrow \boldsymbol{y}$
- Often better performance for large sample sizes

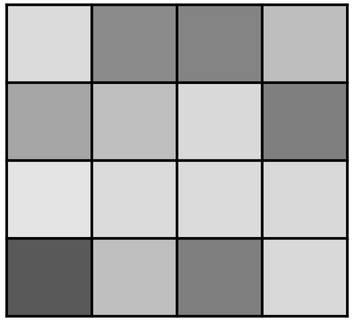
Examples: logistic regression, support vector machines, neural networks, k nearest neighbors







Synthetic "Image" sampled from **Class 0**



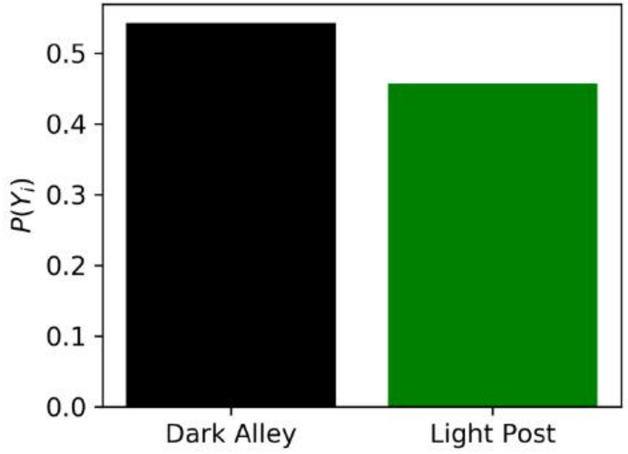
Synthetic "Image" sampled from **Class 1**

How to "Generate" Data?

Class 1: Light Image y_1

Class 0: Dark Image y_0

Prior: $P(Y = y_i)$



Bayes' Rule

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

Generative modeling for classification

Assume we have c different classes

For a new sample, classify it as the class with the **largest posterior** $P(y = i|\boldsymbol{x})$

$i = 1, \dots, c$

Generative modeling for classification

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

If $P(y = i|\boldsymbol{x}) > P(y = j|\boldsymbol{x})$ for $i \neq j$, then we classify feature $\boldsymbol{x}$ to class i

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

$$= \frac{P(\boldsymbol{x}|y = i)P(y = i)}{\sum_{i=1}^{c} P(\boldsymbol{x}|y = i)P(y = i)}$$

Bayes' Rule: $\underbrace{P(Y|\boldsymbol{X})}_{\text{Posterior}} = \frac{\overbrace{P(\boldsymbol{X}|Y)}^{\text{Likelihood}}\overbrace{P(Y)}^{\text{Prior}}}{\underbrace{P(\boldsymbol{X})}_{\text{Evidence}}}$

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 define the discriminant function as:

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

If we know the **true likelihood and prior** for our data, this process yields our **Bayes' classifier** (minimum misclassification error classifier)

Generative modeling for classification

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

We **rarely** know our **true likelihood** for our data so we need to assume a form for the distributions and approximate

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

   Gaussian → Linear and Quadratic Discriminant Analysis
   Gaussian mixture models
   Nonparametric density estimates

   If we assume **independent features** → **Naïve Bayes**

(remember $d_i(x)$ proportional to the posterior)

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

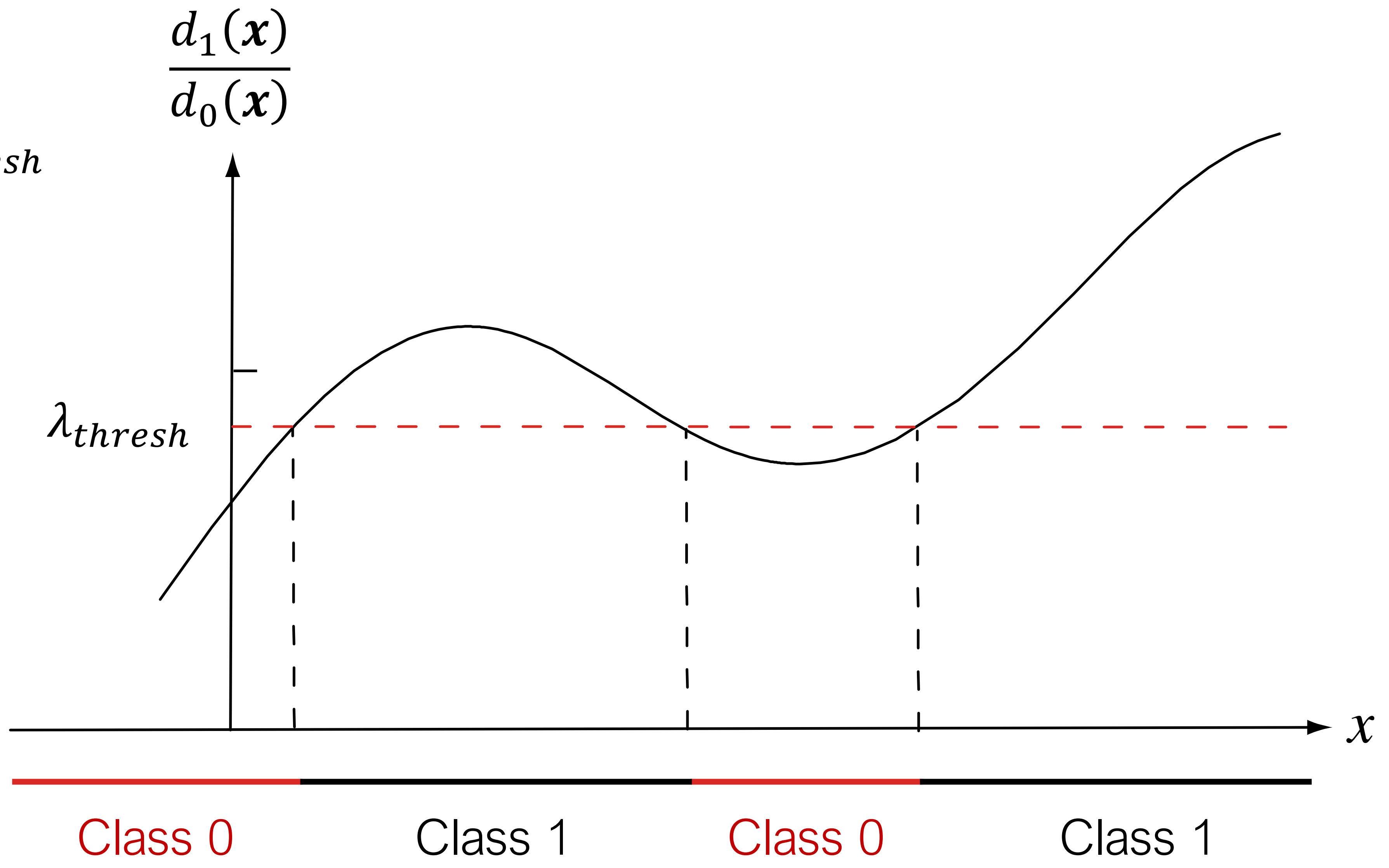
   Applies to both binary and multiclass problems

Discriminant Function: 2 classes

Decision rule:

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

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 $$\frac{d_1(\boldsymbol{x})}{d_0(\boldsymbol{x})} = \frac{P(\boldsymbol{x}|y=1)P(y=1)}{P(\boldsymbol{x}|y=0)P(y=0)} > 1$$ Assign class 1, else class 0

Assumes these likelihoods are normal

$N(\mu_1, \Sigma_1)$

We can write our decision rule as a function of likelihoods (**likelihood ratio**)

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

$N(\mu_0, \Sigma_0)$

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:

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

Likelihood ratio:

$N(\mu_1, \Sigma_1)$

$N(\mu_0, \Sigma_0)$

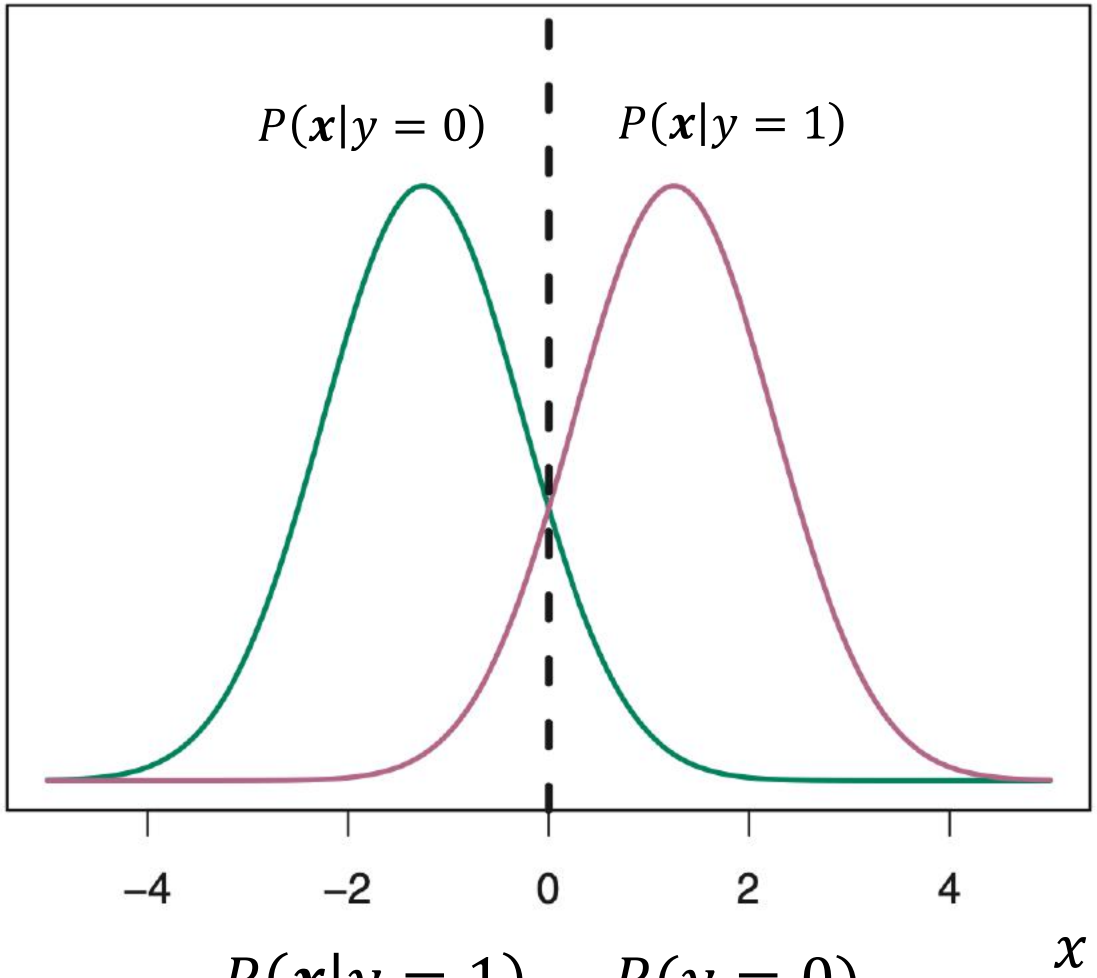
If we assume the class conditional distributions are **Gaussian**, this represents

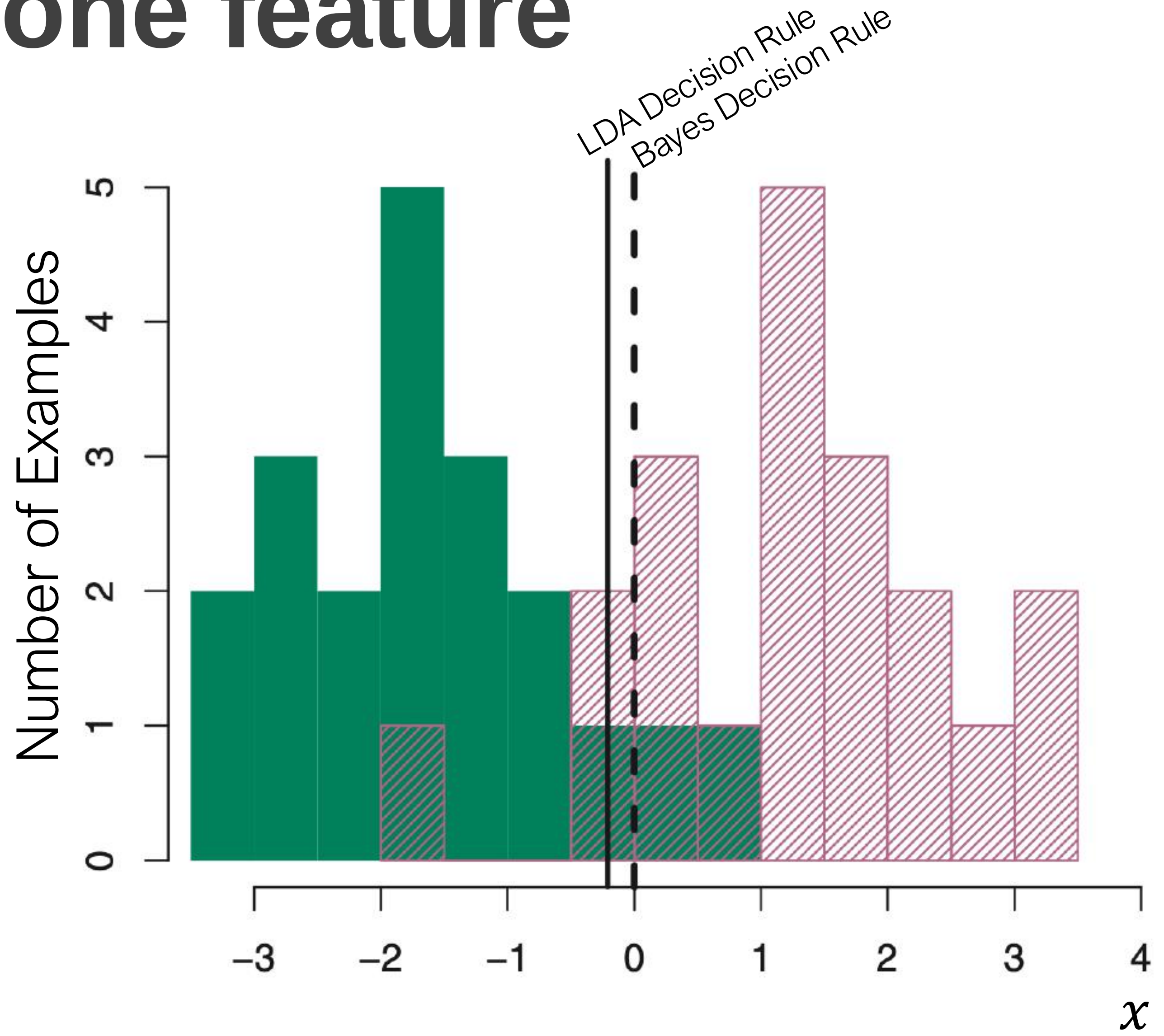
Quadratic Discriminant Analysis

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

Linear Discriminant Analysis

Simple example with one feature





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

Figures from James et al. - Introduction to Statistical Learning

Covariance matrix

$$\mathbf{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{iD} \end{bmatrix}$$

Vector of observation i

$$x_{ij}$$

Observation index / Predictor index

Predictors

$$\mathbf{X} = \begin{array}{c} \\ \text{Observations} \end{array} \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{ND} \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \rightarrow [D \times D]$$

$$[D \times 1][1 \times D]$$

Mean of each predictor

If $\bar{x}_j = 0$ for all j

This will be the case IF the data are standardized

$$\boldsymbol{\Sigma} = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{1D} \\ \Sigma_{21} & \Sigma_{22} & \cdots & \Sigma_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{D1} & \Sigma_{D2} & \cdots & \Sigma_{DD} \end{bmatrix}$$

$$\Sigma_{jk} = \frac{1}{N}\sum_{i=1}^{N}(x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)$$

$$= \operatorname{cov}(X_j, X_k)$$

$$= E[(X_j - \mu_j)(X_k - \mu_k)]$$

$$\Sigma_{jk} = \frac{1}{N}\sum_{i=1}^{N} x_{ij}x_{ik}$$

$$= \frac{1}{N}\mathbf{x}_j^T\mathbf{x}_k$$

$$= E[X_j X_k]$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \Sigma_{12} & \cdots & \Sigma_{1D} \\ \Sigma_{21} & \sigma_2^2 & \cdots & \Sigma_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{D1} & \Sigma_{D2} & \cdots & \sigma_D^2 \end{bmatrix}$$

$$\sigma_j^2 = \frac{1}{N}\sum_{i=1}^{N}(x_{ij} - \bar{x}_j)^2$$

$$= E[(X_j - \mu_j)^2]$$

$$\boldsymbol{\Sigma} = \frac{1}{N}\mathbf{X}^{\mathrm{T}}\mathbf{X}$$

Covariance and Correlation

differences please!

Relationship between covariance and correlation

$$\mathrm{corr}(X,Y) = \frac{\mathrm{cov}(X,Y)}{\sqrt{\mathrm{var}(X)\mathrm{var}(Y)}}$$

When $\mathrm{var}(X) = \mathrm{var}(Y) = 1$, then:

$$\mathrm{corr}(X,Y) = \mathrm{cov}(X,Y)$$

If each of the features have been standardized, this means this matrix is:

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1D} \\ \rho_{21} & 1 & \cdots & \rho_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{D1} & \rho_{D2} & \cdots & 1 \end{bmatrix}$$

Covariance Matrix Examples

diagonal only --> no correlation

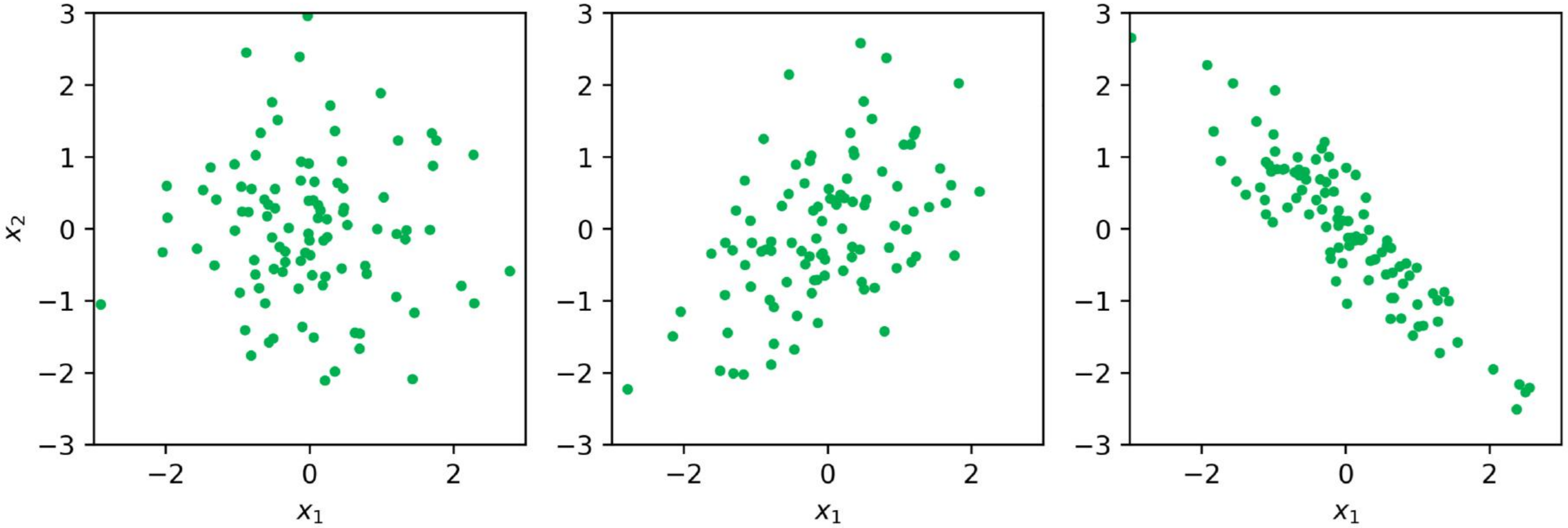
$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

some correlation

$$\Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

strong correlation

$$\Sigma = \begin{bmatrix} 1 & -0.9 \\ -0.9 & 1 \end{bmatrix}$$



Key model differences

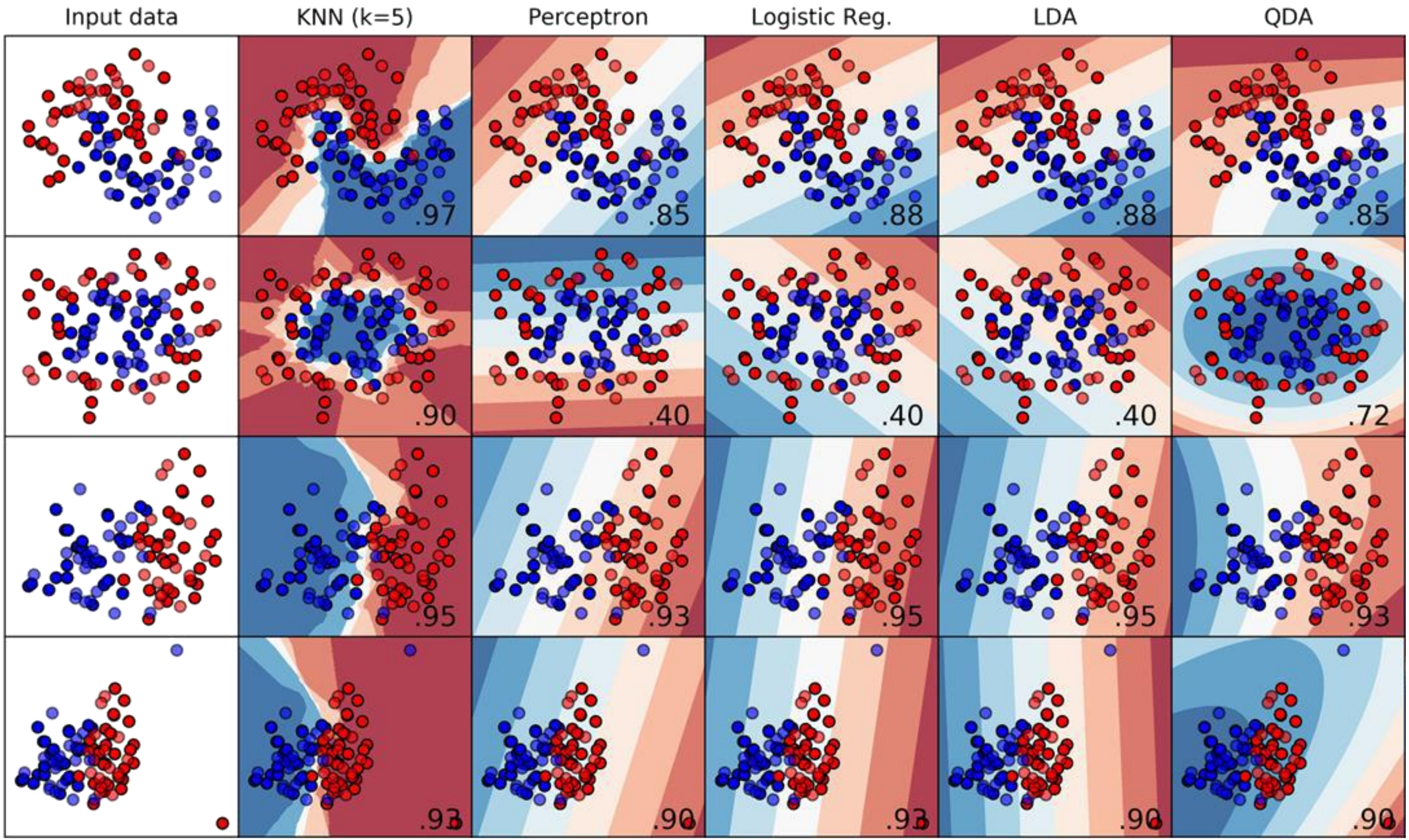
Quadratic Discriminant Analysis (QDA) $\boldsymbol{\Sigma}_0 \neq \boldsymbol{\Sigma}_1$ Each class (e.g., 0 or 1) may have a **unique** covariance matrix

Linear Discriminant Analysis (LDA) $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1$ Every class (e.g., 0 or 1) has an **identical** covariance matrix

Naïve Bayes with Gaussian Likelihoods

$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_D^2 \end{bmatrix}$$

Every class (e.g., $i = 0$ or 1) has a **diagonal** covariance matrix


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

Joint vs Marginal Densities 边际

The marginal densities don't factor in relationships between features assume features are independent

What if the joint density is too hard to estimate?

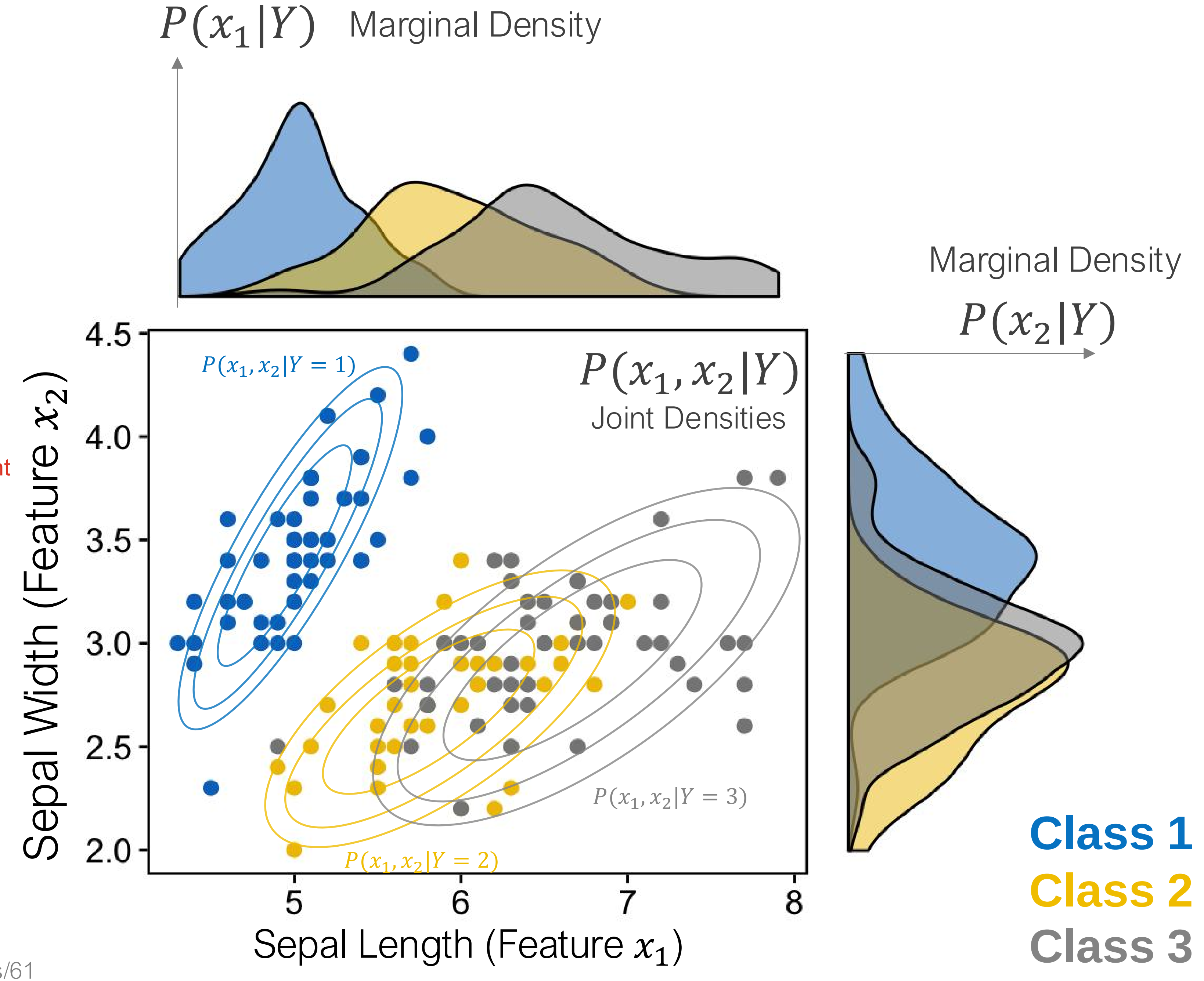


Image adapted from: https://github.com/daattali/ggExtra/issues/61

Naïve 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 discriminant function
(proportional to posterior distribution)

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

Write out the full expression with all the terms in $\boldsymbol{x}$
(assume p predictors/features)

$$d_i(x_1, x_2, \dots, x_p) = P\left(x_1, x_2, \dots, x_p \middle| y = i\right)P(y = i)$$

Assumption: Given the class, the features are independent

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

product of each marginal densities

Predict the class with the highest discriminant function
(i.e. posterior probability)

Naïve Bayes

We assign the class that has the largest discriminant (i.e. posterior probability)

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

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 for the likelihood $P(x_j | y = i)$

Naïve Bayes: Gaussian example

We assign the class that has the largest discriminant (i.e. posterior probability)

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

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

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

Naïve Bayes: Parameters

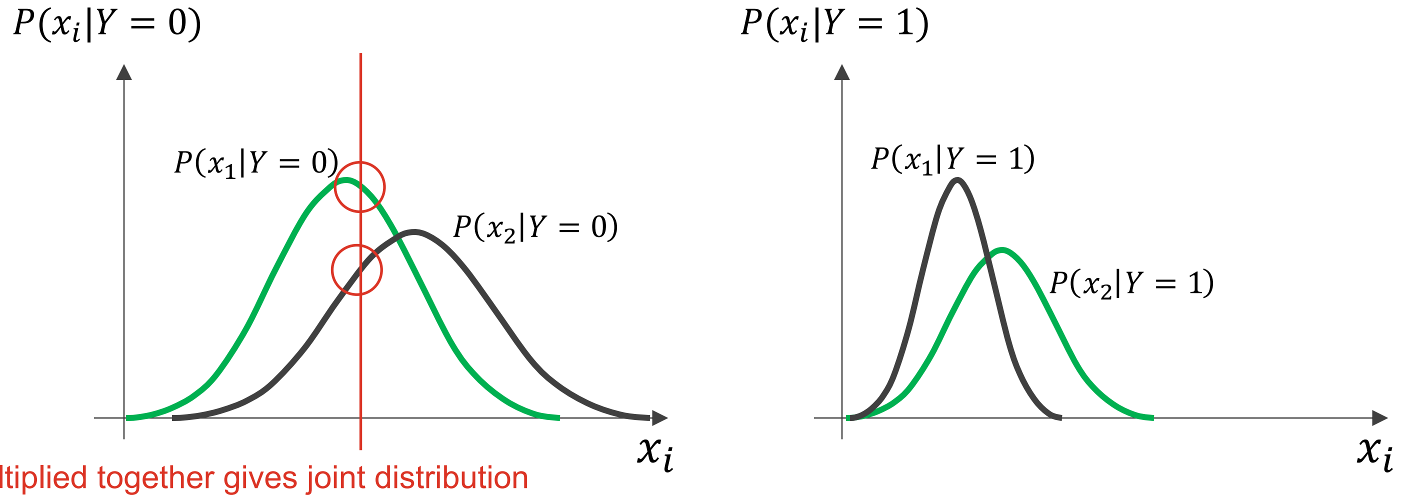
p predictors, c classes

For each predictor, x_i, and class, y_j:

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

Total parameters = $2cp$

Gaussian Naïve Bayes ($p = 2$)



Without the Naïve Bayes independence 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 = $c\left(p + \frac{p^2 - p}{2} + p\right)$

Multivariate Gaussian ($p = 2$)

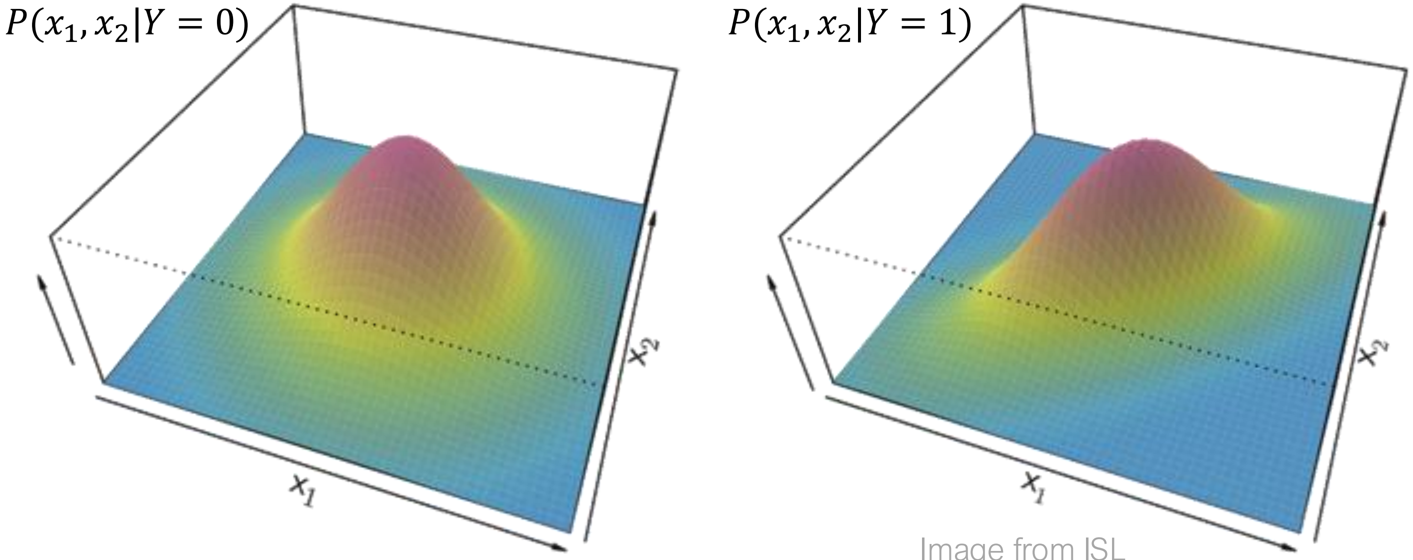
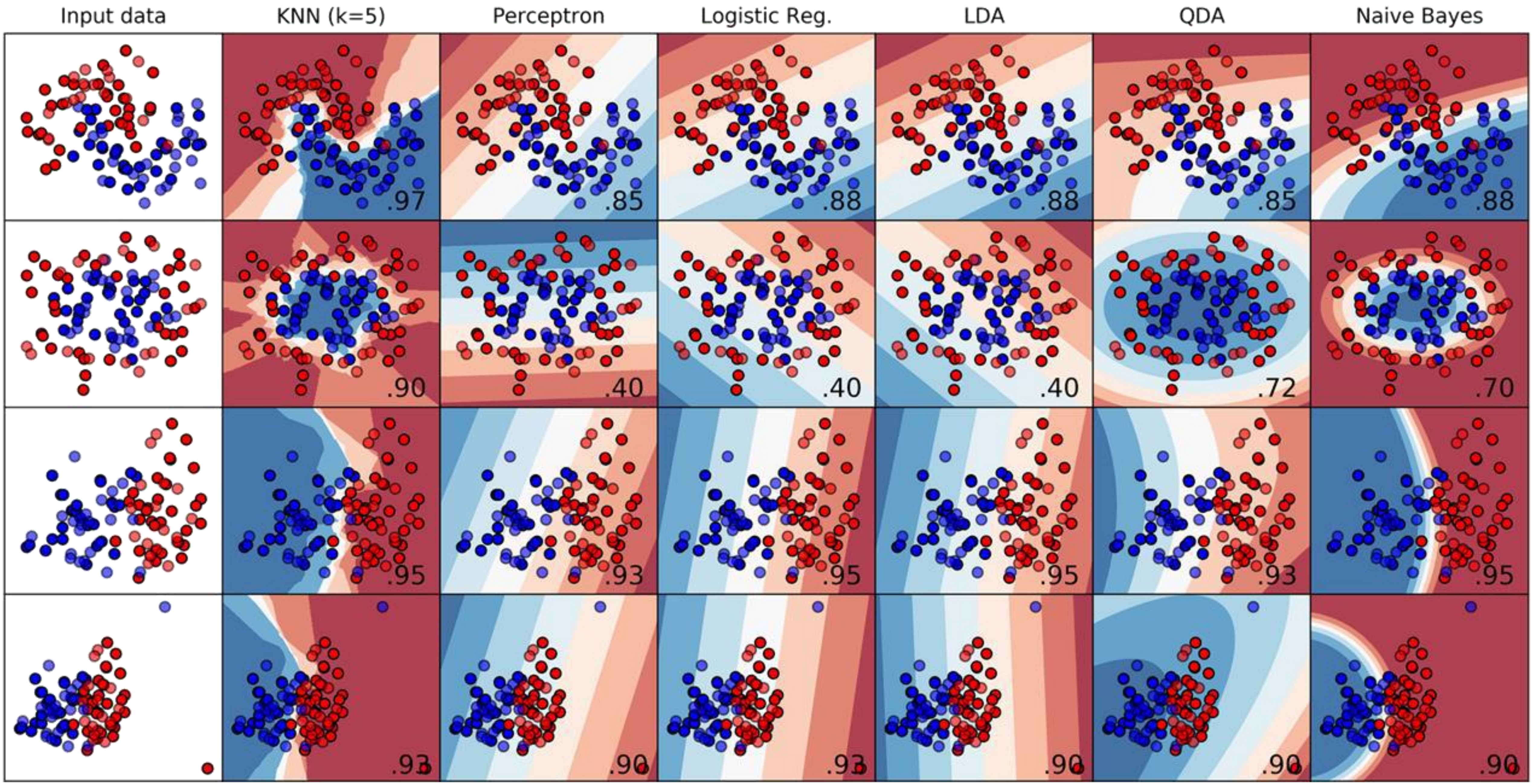


Image from ISL


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

Perceptron

Logistic Regression

Linear Discriminant Analysis

Quadratic Discriminant Analysis

Naïve Bayes

Have closed-form solutions

Apply to multiclass problems

No hyperparameters

Fast to train

Requires small amounts of training data

Only choice is the form of P(X|Y)
(otherwise no parameter choices)

Fast to train

Examples of deep generative models…

Variational Auto Encoders
Normalizing Flows
Generative Adversarial Networks (GANs)
Diffusion Models

Face Synthesis

Image Synthesis ([link]())

Karras et al. 2018, NVIDIA: Progressive growing of GANS for improved quality, stability, and variation

These images are all synthetic

Synthetic Generation

Karras, T., Laine, S. and Aila, T., 2019. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4401-4410).

Synthetic Images

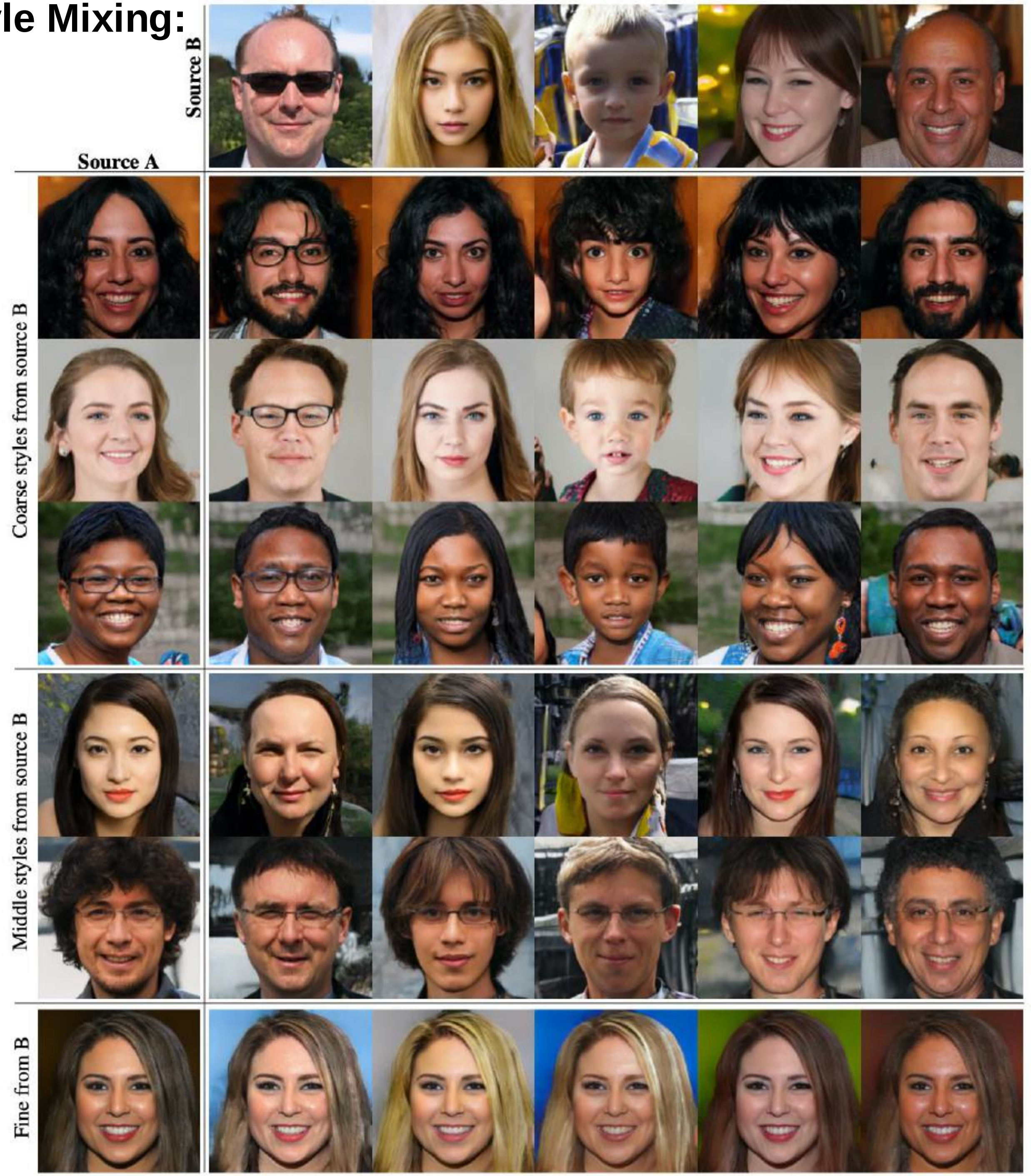
Style Mixing:

From source B:

Pose, general hair style, face shape, eyeglasses

Hair style, eyes open/closed

Color scheme and microstructure



Supervised learning in practice

Preprocessing
Explore & prepare data

Data Visualization and Exploration

Identify patterns that can be leveraged for learning

Data Cleaning

- Missing data
- Noisy data
- Erroneous data

Scaling (Standardization)

Prepare data for use in scale-dependent algorithms.

Feature Extraction

Dimensionality reduction eliminates redundant information

Model training

1 Supervised Learning Models: Linear models and KNN
(enough to get started using supervised learning)

Select model options

5 Other algorithms and concepts:
- **Generative vs discriminative models**
- Parametric vs nonparametric models
- Model ensembles
- Feature/representation learning (neural networks, deep learning)

4 How to control model overfit: regularization strategies for model refinement

Iteratively fine tune the model

Performance evaluation

Make a prediction on validation data

2 Evaluating model performance and comparing models

Classification

3 How to make decisions using models

Regression
MSE, explained variance, R^2