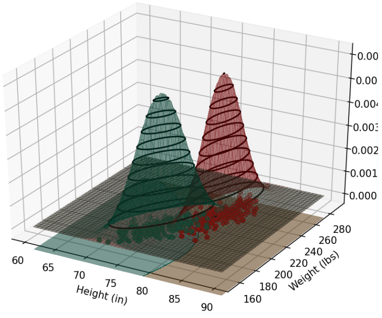


CSE 151A

Intro to Machine Learning

Lecture 04 – Part 01

Bayes with Multiple Features

Recap

- Bayes Classifier: predict y_i that maximizes $P(Y = y_i \mid X = x)$
- We have to estimate these probabilities.

Recap

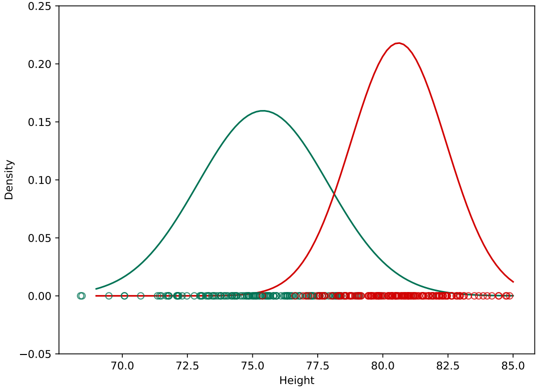
- Approach #1: Estimate $P(Y = y_i \mid X = x)$ using k neighbors.

- Approach #2: Use Bayes' Rule to write:

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

  Estimate $P(X = x \mid Y = y_i)$ using histograms or by fitting Gaussians.

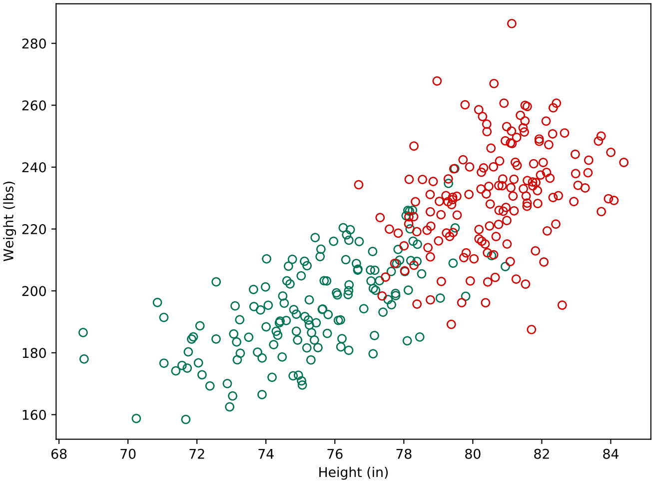
Recap



$P(Y = \text{guard} \mid X = x)P(Y = \text{guard})$ and $P(Y = \text{forward} \mid X = x)P(Y = \text{forward})$

Today

- How do we use more than one feature?
- Example: predict using height *and* weight.

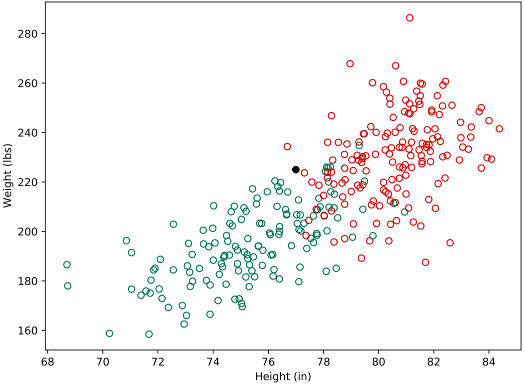

280
260
240
220
200
180
160
Weight (lbs)
68
70
72
74
76
78
80
82
84
Height (in)

Bayes in ≥ 2 Dimensions

- Instead of
$$P(Y = y_i | X = x)$$
we have
$$P(Y = y_i | \vec{X} = \vec{x})$$
- $\vec{x}$ is the **feature vector**. Here: (height, weight)^T

Approach #1

- Can estimate $P(Y = y_i \mid X = x_i)$ using k neighbors.



Approach #2: Generative Modeling

- Use Bayes' Rule:

$$P(Y = y_i \mid \vec{X} = \vec{x}) = \frac{P(\vec{X} = \vec{x} \mid Y = y_i)P(Y = y_i)}{P(\vec{X} = \vec{x})}$$

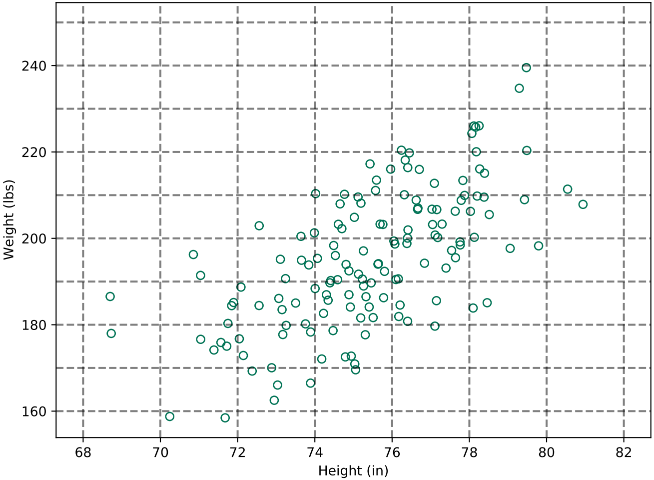
- Estimate $P(\vec{X} = \vec{x} \mid Y = y_i)$ and $P(Y = y_i)$.

Estimating Density

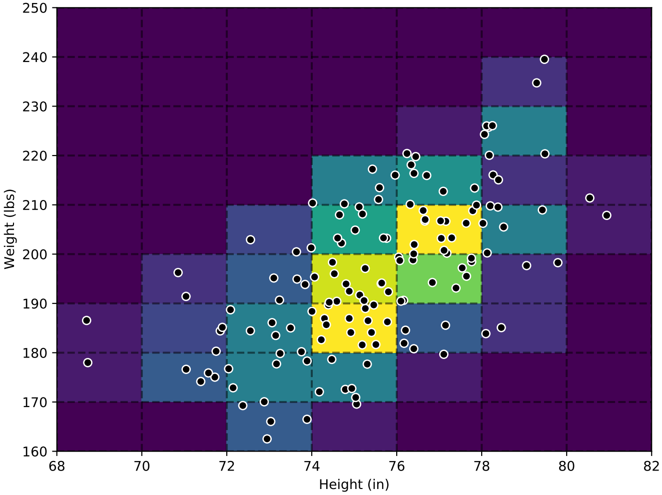
- We need to estimate $P(\vec{X} = \vec{x} \mid Y = y_i)$ for each class $y_1, \ldots, y_k$.

- See two methods: histograms and Gaussians.

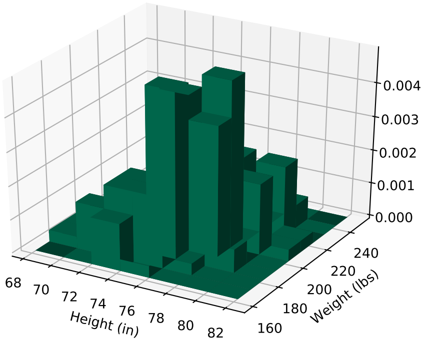
Estimating with Histograms



Estimating with Histograms



Estimating with Histograms



Predicting with Histograms

To predict the class of an input $\vec{x}$:

1. Use histograms to estimate $P(\vec{X} = \vec{x} \mid Y = y_i)$ for each class independently.

2. Predict the class y_i maximizing

$$P(\vec{X} = \vec{x} \mid Y = y_i)P(Y = y_i)$$

Histogram Estimators in d > 2

- With 1 feature, each bin is an interval.
- With 2 features, each bin is a rectangle.
- With 3 features, each bin is a cuboid (box).
- With >4 features, each bin is a **hypercuboid**.

Curse of Dimensionality

- We need enough bins to "cover" the input space.
- **Problem**: Number of bins is exponential in d.
- Example: split each dimension into 10 pieces.

Example

- In 2-d: $10^2 = 100$ bins.

Example

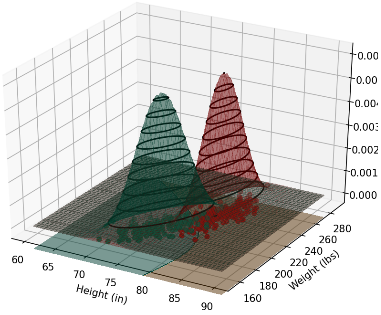
- In 2-d: $10^2 = 100$ bins.
- In 100-d: 10^{100} bins.
- More bins than atoms in universe.
- Highly likely that no bin has more than a few points.

Histogram Estimators

- Histogram density estimators are very general.
- But suffer heavily from **curse of dimensionality.**
- Then again, so do most things.

Up next...

- What about fitting Gaussians?



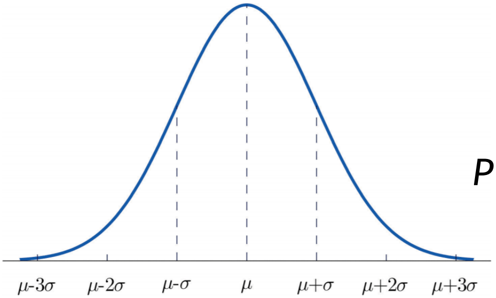
CSE 151A
Intro to Machine Learning

Lecture 04 – Part 02
Multivariate Gaussians

Multivariate Gaussians

- In 1 dimension, a Gaussian seemed to describe distribution of heights.
- Does a **multivariate** Gaussian describe distribution of heights and weights?

“Deriving” Multivariate Gaussians



$$X \sim \mathcal{N}(\mu, \sigma^2)$$

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(x-\mu)^2/\sigma^2}$$

Setting #1

- Suppose we have d independent random variables $X_1, \ldots, X_d$.

- Assume that each is Gaussian; different mean, but **same** variance:

$$X_1 \sim \mathcal{N}(\mu_1, \sigma^2), \quad X_2 \sim \mathcal{N}(\mu_2, \sigma^2), \ldots, \quad X_d \sim \mathcal{N}(\mu_d, \sigma^2).$$

Setting #1

- What is $P(x_1, x_2, \dots, x_d)$?

- Since we assumed $X_1, \dots, X_d$ are independent:

$$P(x_1, x_2, \dots, x_d) = P(x_1)P(x_2)\cdots P(x_d)$$

$$= \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(x-\mu_1)^2/\sigma^2}\right) \cdot \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(x-\mu_2)^2/\sigma^2}\right) \cdots \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(x-\mu_d)^2/\sigma^2}\right)$$

Setting #1

- What is $P(x_1, x_2, \dots, x_d)$?

- Since we assumed $X_1, \dots, X_d$ are independent:

$$P(x_1, x_2, \dots, x_d) = P(x_1)P(x_2)\cdots P(x_d)$$

$$= \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(x-\mu_1)^2/\sigma^2}\right) \cdot \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(x-\mu_2)^2/\sigma^2}\right) \cdots \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(x-\mu_d)^2/\sigma^2}\right)$$

$$= \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{(x_1-\mu_1)^2 + (x_2-\mu_2)^2 + \dots + (x_d-\mu_d)^2}{2\sigma^2}\right)$$

Setting #1

- What is $P(x_1, x_2, \dots, x_d)$?

- Since we assumed $X_1, \dots, X_d$ are independent:

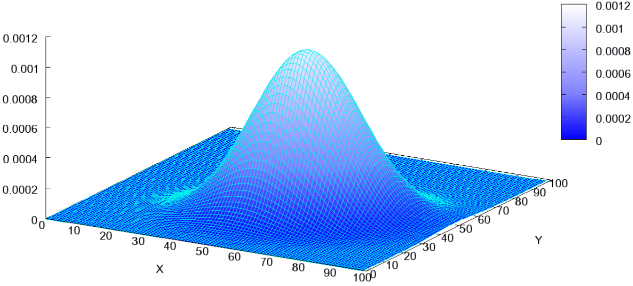
$$P(x_1, x_2, \dots, x_d) = P(x_1)P(x_2)\cdots P(x_d)$$

$$= \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(x-\mu_1)^2/\sigma^2}\right) \cdot \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(x-\mu_2)^2/\sigma^2}\right) \cdots \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(x-\mu_d)^2/\sigma^2}\right)$$

$$= \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{(x_1 - \mu_1)^2 + (x_2 - \mu_2)^2 + \dots + (x_d - \mu_d)^2}{2\sigma^2}\right)$$

$$= \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{\|\vec{x} - \vec{\mu}\|^2}{2\sigma^2}\right)$$

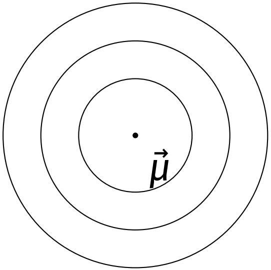
Setting #1


0.0012
0.001
0.0008
0.0006
0.0004
0.0002
0
0
10
20
30
40
50
60
70
80
90
100
X
0
10
20
30
40
50
60
70
80
90
100
Y
0.0012
0.001
0.0008
0.0006
0.0004
0.0002
0

Setting #1: Spherical Gaussians

$$P(\vec{x}) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{1}{2}\frac{\|\vec{x}-\vec{\mu}\|^2}{\sigma^2}\right)$$

- Contours are (hyper)spheres.
- Every slice through middle gives same Gaussian.



Setting #2

- Still assume $X_1, \ldots, X_d$ are independent, Normal.
- But they now have different variances:

$$X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2), \quad X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2), \ldots, \quad X_d \sim \mathcal{N}(\mu_d, \sigma_d^2).$$

Setting #2

$$P(x_1, x_2, \ldots, x_d) = P(x_1)P(x_2)\cdots P(x_d)$$

$$= \left(\frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{1}{2}(x-\mu_1)^2/\sigma_1^2}\right) \cdot \left(\frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{1}{2}(x-\mu_2)^2/\sigma_2^2}\right) \cdots \left(\frac{1}{\sqrt{2\pi\sigma_d^2}} e^{-\frac{1}{2}(x-\mu_d)^2/\sigma_d^2}\right)$$

Setting #2

$$P(x_1, x_2, \ldots, x_d) = P(x_1)P(x_2)\cdots P(x_d)$$

$$= \left(\frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{1}{2}(x-\mu_1)^2/\sigma_1^2}\right) \cdot \left(\frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{1}{2}(x-\mu_2)^2/\sigma_2^2}\right) \cdots \left(\frac{1}{\sqrt{2\pi\sigma_d^2}} e^{-\frac{1}{2}(x-\mu_d)^2/\sigma_d^2}\right)$$

$$= \frac{1}{(2\pi)^{d/2}\sigma_1 \cdot \sigma_2 \cdots \sigma_d} \exp\left(-\frac{1}{2}\left[\frac{(x_1-\mu_1)^2}{\sigma_1^2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2} + \ldots + \frac{(x_d-\mu_d)^2}{\sigma_d^2}\right]\right)$$

Setting #2

- Define

$$C = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \sigma_d^2 \end{pmatrix}$$

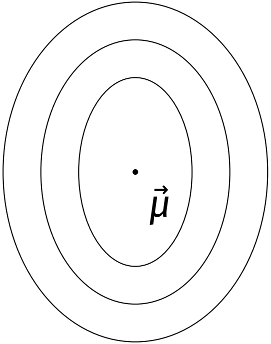
- Then:

$$P(\vec{x}) = \frac{1}{(2\pi)^{d/2}|C|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu})^T C^{-1} (\vec{x} - \vec{\mu})\right)$$

Setting #2: Diagonal Gaussians

$$P(\vec{x}) = \frac{1}{(2\pi)^{d/2}|C|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\vec{x}-\vec{\mu})^T C^{-1}(\vec{x}-\vec{\mu})\right)$$

- Contours are axis-aligned (hyper)ellipses.
- C is the **covariance matrix**.
  - Diagonal.
  - Entries are variances.



Setting #3: General Gaussians

- We have assumed that $X_1, \dots, X_d$ are independent.

- Now assume that they're not. Define **covariance**:

$$\mathrm{Cov}(X_i, X_j) = \mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)]$$

- **Note**:

$$\mathrm{Var}(\mathrm{X_i}) = \mathrm{Cov}(X_i, X_j)$$

Setting #3: General Gaussians

- Now the **covariance matrix** has off-diagonal elements:

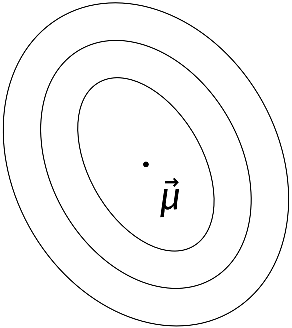
$$C = \begin{pmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \cdots & \mathrm{Cov}(X_1, X_d) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \cdots & \mathrm{Cov}(X_2, X_d) \\ \cdots & \cdots & \cdots & \cdots \\ \mathrm{Cov}(X_d, X_1) & \mathrm{Cov}(X_d, X_2) & \cdots & \mathrm{Var}(X_d) \end{pmatrix}$$

- Since $\mathrm{Cov}(X_i, X_j) = \mathrm{Cov}(X_j, X_i)$, C is symmetric.

Setting #3: General Gaussians

$$P(\vec{x}) = \frac{1}{(2\pi)^{d/2}|C|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu})^T C^{-1}(\vec{x} - \vec{\mu})\right)$$

Contours are general (hyper)ellipses.
C need not be diagonal.



Fitting Multivariate Gaussians

- Given vectors $\vec{x}^{(1)}, \ldots, \vec{x}^{(n)}$, fit a Gaussian.
- First, choose assumptions. Spherical? Diagonal? General (no assumptions)?
- In each case,

$$\vec{\mu} = \frac{1}{n} \sum_{i=1}^{n} \vec{x}^{(i)}$$

Fitting Spherical Gaussians

- Only one variance: σ^2.

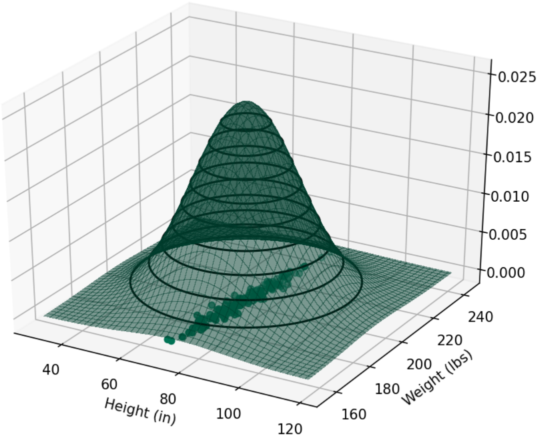
- In 1 dimension:

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2$$

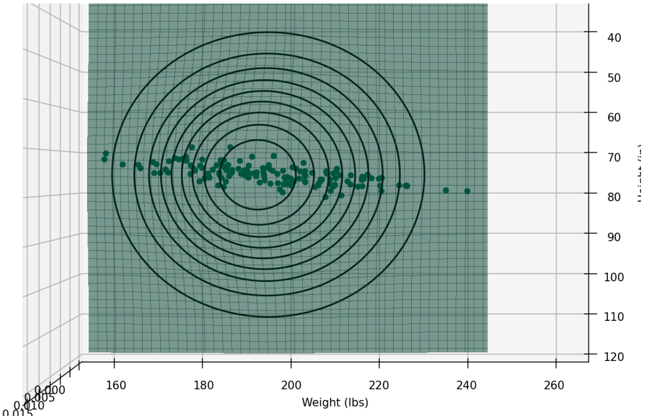
- In d dimensions:

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} \| \vec{x}^{(i)} - \vec{\mu} \|^2$$

Fitting Spherical Gaussians



Fitting Spherical Gaussians



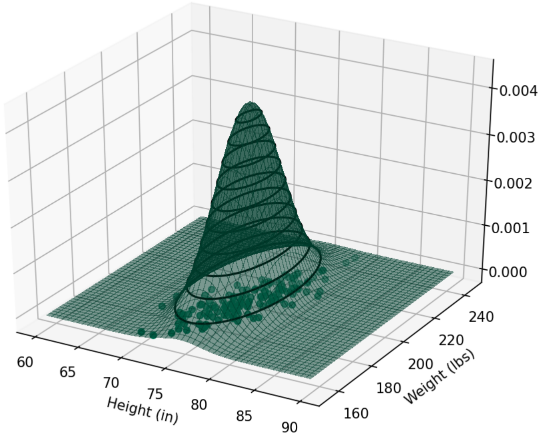
Fitting Diagonal Gaussians

- Variance for each axis: σ_1^2 and σ_2^2.

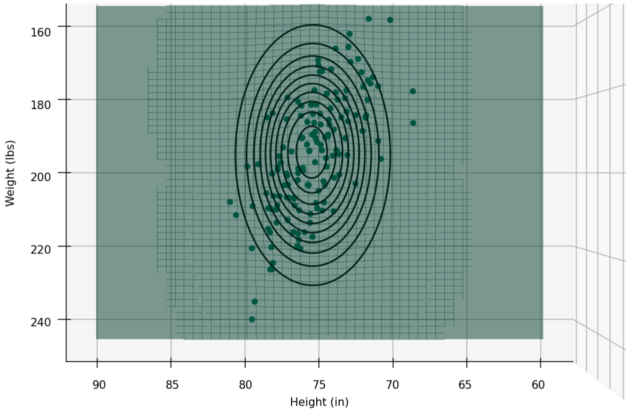
- Example:

$$\sigma_1^2 = \text{variance of heights}$$
$$\sigma_2^2 = \text{variance of weights}$$

Fitting Diagonal Gaussians



Fitting Diagonal Gaussians



Fitting General Gaussians

- Must compute covariance for each pair of dimensions.

- Empirical covariance:

$$C_{ij} = \left(\frac{1}{n} \sum_{k=1}^{n} \vec{x}_i^{(k)} \vec{x}_j^{(k)} \right) - \mu_i \mu_j$$

Computing the Covariance Matrix

Step 1. Make matrix with heights in first column, weights in second:

$$\begin{pmatrix} \text{height } 1 & \text{weight } 1 \\ \text{height } 2 & \text{weight } 2 \\ \ldots & \ldots \\ \text{height } n & \text{weight } n \end{pmatrix}$$

Computing the Covariance Matrix

Step 2. Subtract mean height, mean weight from each column. Call this matrix X:

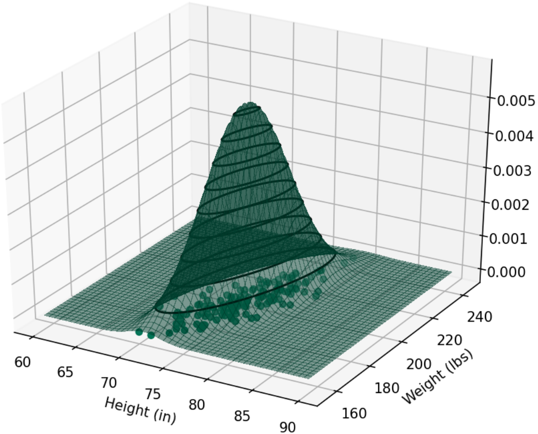
$$X = \begin{pmatrix} \text{height 1} - \text{mean height} & \text{weight 1} - \text{mean weight} \\ \text{height 2} - \text{mean height} & \text{weight 2} - \text{mean weight} \\ \dots & \dots \\ \text{height } n - \text{mean height} & \text{weight } n - \text{mean weight} \end{pmatrix}$$

Computing the Covariance Matrix

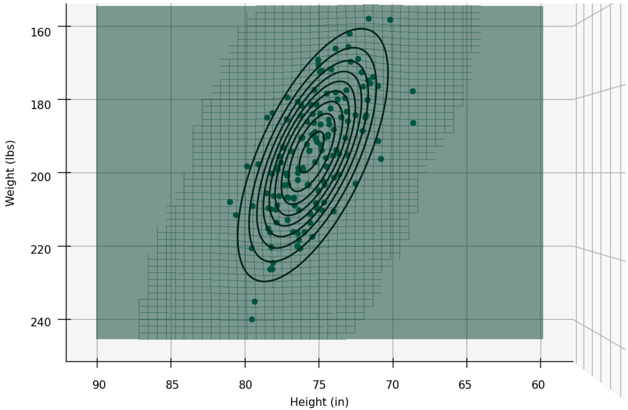
The empirical covariance matrix is then:

$$C = \frac{1}{n} X^T X$$

Fitting General Gaussians

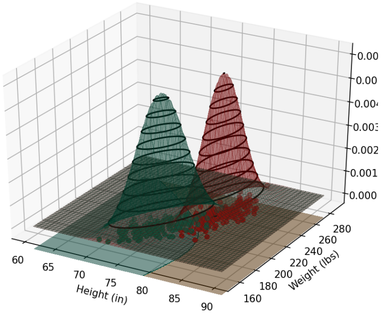


Fitting General Gaussians



Up next...

Making predictions using these fitted Gaussians.



CSE 151A

Intro to Machine Learning

Lecture 04 – Part 03

Discriminant Analysis

Bayes Classifier with MV Gaussians

1. Fit Gaussian for $P(\vec{X} \mid Y = y_i)$ for each class, y_i.

2. For new point, predict y_i maximizing:

$$P(\vec{X} = \vec{x} \mid Y = y_i)P(Y = y_i)$$

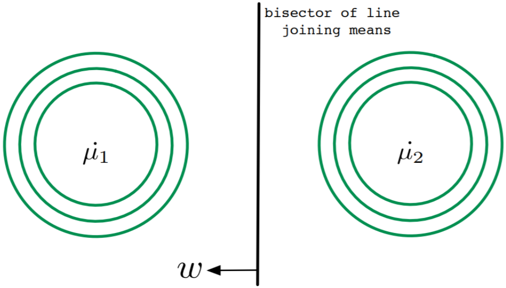
Decision Boundary

- For every point in space, we have a classification.

- The **decision boundary**: surface between different classifications.
  - On one side, prediction is y_1;
  - on the other, prediction is y_2.

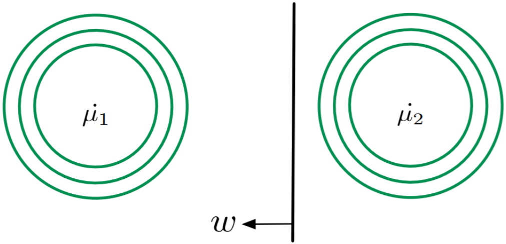
Setting #1

- Assume:
  - only two classes (binary classification)
  - covariance matrices identical, spherical



Setting #1

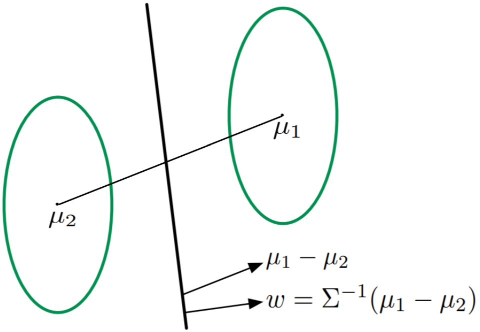
- If $P(Y = y_1) > P(Y = y_2)$:



Choose class 1 if $\vec{w} \cdot \frac{(\vec{\mu}_1 - \vec{\mu}_2)}{\sigma^2} \geq \theta$.

Setting #2

- Assume:
  - only two classes (binary classification)
  - covariance matrices identical, non-diagonal



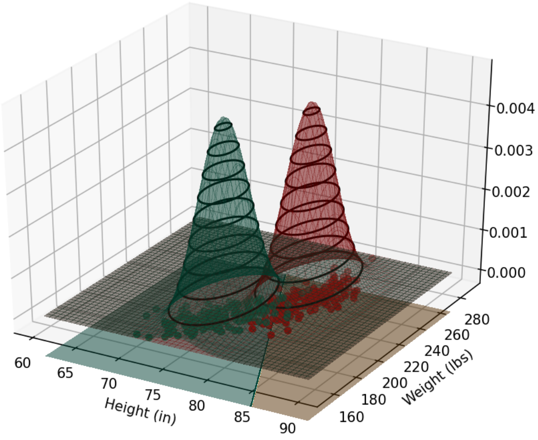
Predict class 1 if $\vec{x} \cdot \vec{w} \geq \theta$.

Example

- Use to predict position given height and weight.
- How do we get one covariance matrix?
- Don't lump data together...
- Instead, compute covariance matrix for each class, perform weighted average:

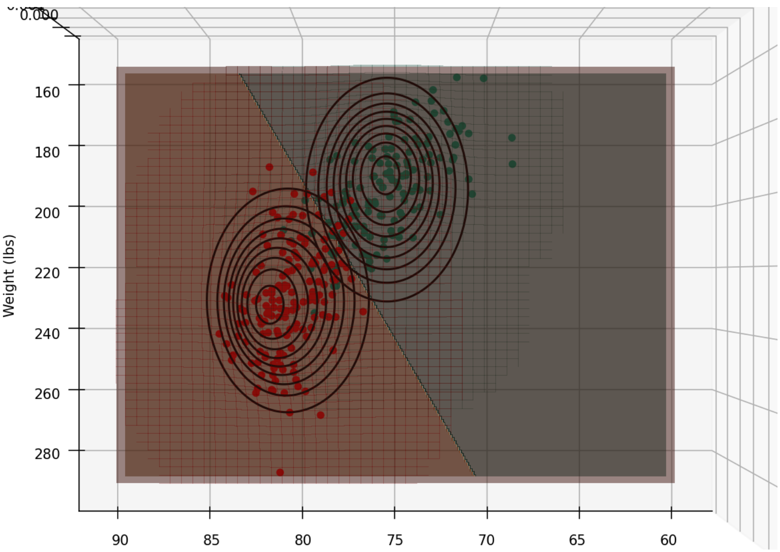
$$C = \frac{n_1 C_1 + n_2 C_2}{n_1 + n_2}$$

Example



0.004
0.003
0.002
0.001
0.000
60
65
70
75
80
85
90
Height (in)
160
180
200
220
240
260
280
Weight (lbs)

Example

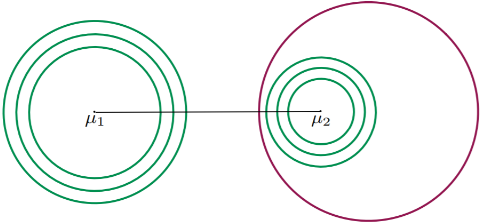


Linear Discriminant Analysis

- When covariance matrices are equal, decision boundary is linear.
- This procedure is called **linear discriminant analysis** (LDA).

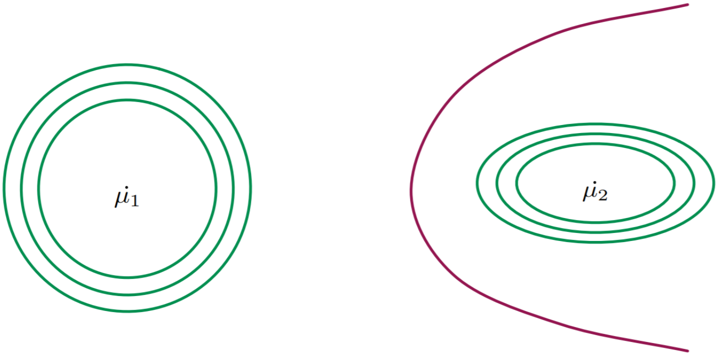
Setting #3

- Assume:
  - only two classes (binary classification)
  - covariance matrices C_1, C_2 different, non-diagonal

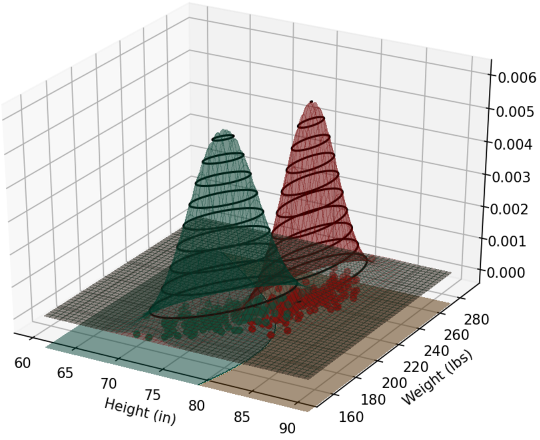


Setting #3

- Assume:
  - only two classes (binary classification)
  - covariance matrices C_1, C_2 different, non-diagonal

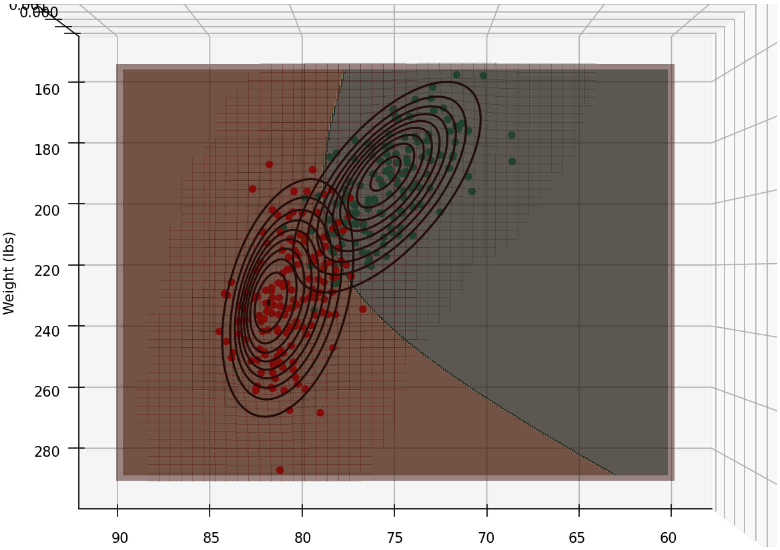


Example



0.006
0.005
0.004
0.003
0.002
0.001
0.000
60
65
70
75
80
85
90
Height (in)
160
180
200
220
240
260
280
Weight (lbs)

Example



Quadratic Discriminant Analysis

- When covariance matrices are equal, decision boundary is quadratic (ellipsoidal, paraboloidal, hyperoboloidal).

- This procedure is called **quadratic discriminant analysis** (QDA).

In practice...

- LDA and QDA can work well.
- A full covariance requires estimating $\Theta(d^2)$ parameters.
- Gaussian assumption may be poor.