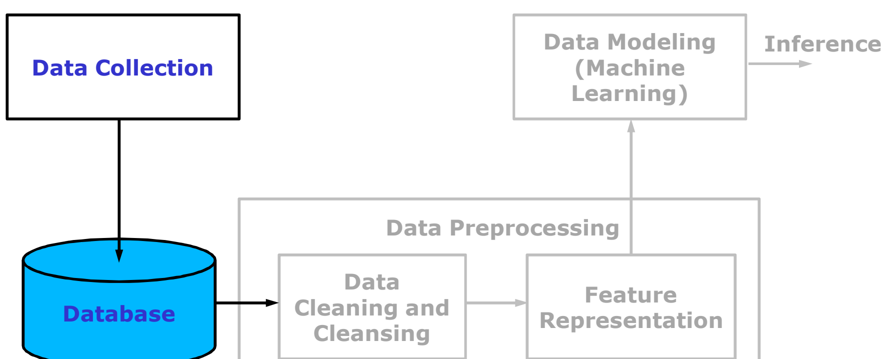
Data Modeling

Data Science

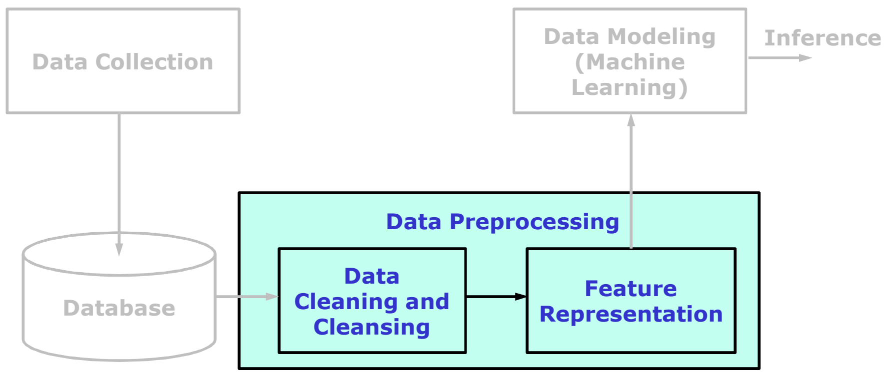
- Multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insight from structured and unstructured data
- Central concept is gaining insight from data
- Machine learning uses data to extract knowledge



2

Data Science

- Multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insight from structured and unstructured data
- Central concept is gaining insight from data
- Machine learning uses data to extract knowledge



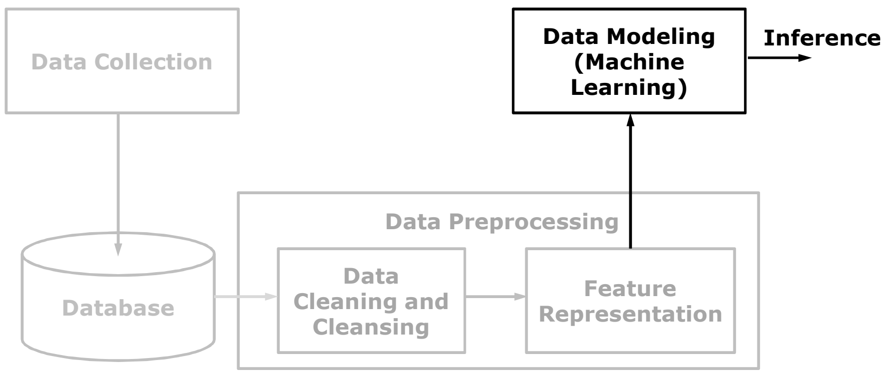
3

Descriptive Data Analytics

- It helps us to study the general characteristics of data and identify the presence of noise or outliers
- Data characteristics:
  - Central tendency of data
    - Centre of the data
    - Measuring mean, median and mode
  - Dispersion of data
    - The degree to which numerical data tend to spread
    - Measuring range, quartiles, interquartile range (IQR), the five-number summery and standard deviation
- Descriptive analytics are the backbone of reporting

Data Science

- Multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insight from structured and unstructured data
- Central concept is gaining insight from data
- Machine learning uses data to extract knowledge



5

Predictive Data Analytics

- It is used to identify the trends, correlations and causation by learning the patterns from data
- Study and construction of algorithms that can learn from data and make predictions on data
- It involve tasks like
  - Classification:
    - E.g.: predicting the presence or absence of disease or
    - the classification of disease according to symptoms
  - Regression: Numeric prediction
    - E.g.: predicting the landslide or
    - predicting the rainfall
  - Clustering:
    - E.g.: grouping the similar items to be sold or
    - grouping the people from the same region
- Learning from data

6

Pattern Classification

Classification

- Problem of identifying to which of a set of categories a new observation belongs
- Predicts categorical labels
- Example:
  - Assigning a given email to the "spam" or "non-spam" class
  - Assigning a diagnosis (disease) to a given patient based on observed characteristics of the patient
- Classification is a two step process
  - Step1: Building a classifier (data modeling)
    - Learning from data (training phase)
  - Step2: Using classification model for classification
    - Testing phase

8

Step1: Building a Classification Model (Training Phase)

- A classifier is built describing a predetermined set of data classes
- This is a learning step (or training phase)
- Training phase: A classification algorithm builds the classifier by analysing or learning from a training data set made up of tuples (samples) and their class labels
- In the context of machine learning, data tuples can be referred to as samples, examples, instance, data vectors, data points

9

Step1: Building a Classification Model (Training Phase)

- Suppose a training data consist of N tuples (or data vectors) described by d-attributes (d -dimensions)

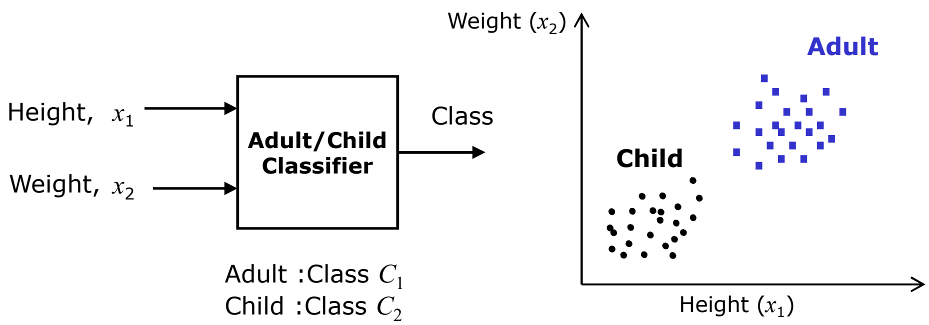
$$\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^{N}, \mathbf{x}_n \in \mathbb{R}^d$$

- Each tuple (or data vector) is assumed to belong to a predefined class
  - Class is determined by another attribute ($(d+1)^{th}$ attribute) called the class label attribute
  - Class label attribute is discrete-valued and unordered
  - It is a *categorical (nominal)* in that each value serves as a category or class
- Individual tuples (or data vectors) making up training set are referred as training tuples or training samples or training examples or training data vectors

10

2-class Classification

- Example: Classifying a person as child or adult



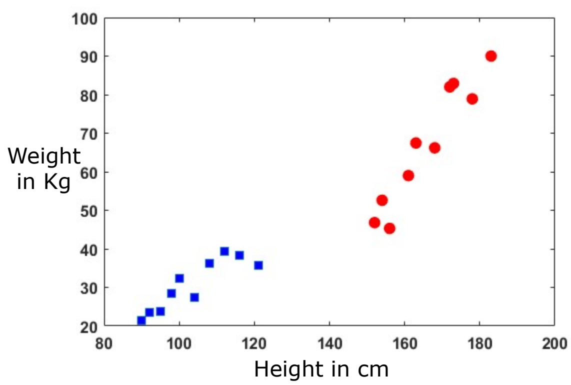
$$\mathbf{x} = [x_1 \;\; x_2]^\mathsf{T}$$

11

Illustration of Training Set: Adult-Child

Height	Weight	Class
90	21.5	0
95	23.67	0
100	32.45	0
116	38.21	0
98	28.43	0
108	36.32	0
104	27.38	0
112	39.28	0
121	35.8	0
92	23.56	0
152	46.8	1
178	78.9	1
163	67.45	1
173	82.9	1
154	52.6	1
168	66.2	1
183	90	1
172	82	1
156	45.3	1
161	59	1

- Number of training examples (N) = 20
- Dimension of a training example = 2
- Class label attribute is 3rd dimension
- Class:
  - Child (0)
  - Adult (1)



12

Illustration of Training Set – Iris (Flower) Data

Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Class
5.1	3.5	1.4	0.2	1
4.9	3	1.4	0.2	1
4.7	3.2	1.3	0.2	1
7	3.2	4.7	1.4	2
6.4	3.2	4.5	1.5	2
6.9	3.1	4.9	1.5	2
6.3	3.3	6	2.5	3
5.8	2.7	5.1	1.9	3
7.1	3	5.9	2.1	3
5.7	2.8	4.1	1.3	2
7.3	2.9	6.3	1.8	3
7.3	2.9	6.3	1.8	3
5.3	3.7	1.5	0.2	1
4.9	2.4	3.3	1	2
5	3.5	1.6	0.6	1
6.3	3.3	4.7	1.6	2
5.8	2.7	3.9	1.2	2
5.8	2.8	5.1	2.4	3
4.4	3	1.3	0.2	1
6.2	3.4	5.4	2.3	3

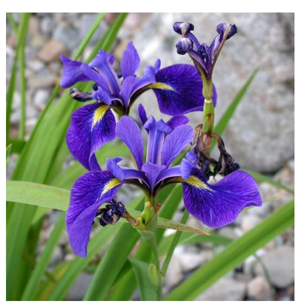
- Number of training examples (N) = 20
- Dimension of a training example = 4
- Class label attribute is 5th dimension
- Class:
  - Iris Setosa (1)
  - Iris Versicolour (2)
  - Iris Virginica (3)

13

Illustration of Training Set – Iris (Flower) Data

Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Class
5.1	3.5	1.4	0.2	1
4.9	3	1.4	0.2	1
4.7	3.2	1.3	0.2	1
7	3.2	4.7	1.4	2
6.4	3.2	4.5	1.5	2
6.9	3.1	4.9	1.5	2
6.3	3.3	6	2.5	3

1: Iris Setosa



2: Iris Versicolour

3: Iris Virginica

14

Step1: Building a Classification Model (Training Phase)

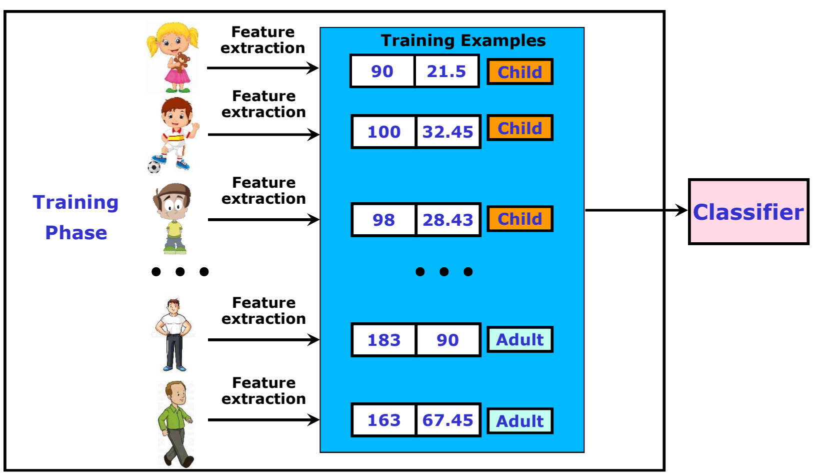
- Training phase or learning phase is viewed as the learning of a mapping or function that can predict the associated class label of a given training example

$$y_n = f(\mathbf{x}_n)$$

  - $\mathbf{x}_n$ is the n^{th} training example and y_n is the associated class label
- Supervised learning:
  - Class label for each training example is provided
  - In supervised learning, each example is a *pair* consisting of an input example (typically a vector) and a desired output value

15

Step1: Building a Classification Model (Training Phase)



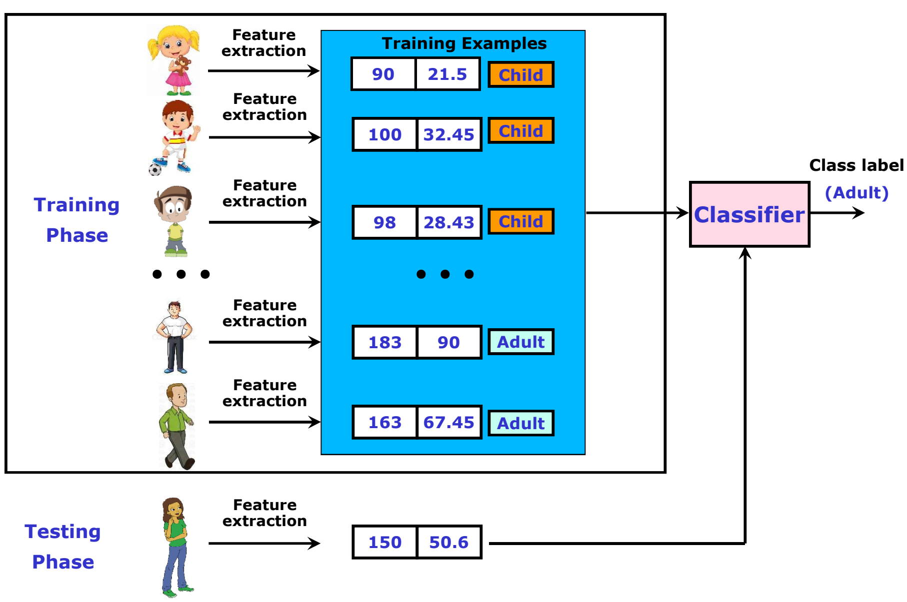
16

Step2: Classification (Testing Phase)

- Trained model is used for classification
- Predictive accuracy of the classifier is estimated
- Accuracy of a classifier:
  - Accuracy of a classifier on a test set is percentage of test examples that are correctly classified by the classifier
  - The associated class label of each test example (ground truth) is compared with the learned classifier's class prediction for that example
- Generalization ability of trained model: Performance of trained models on new (test) data
- Target of learning techniques: Good generalization ability

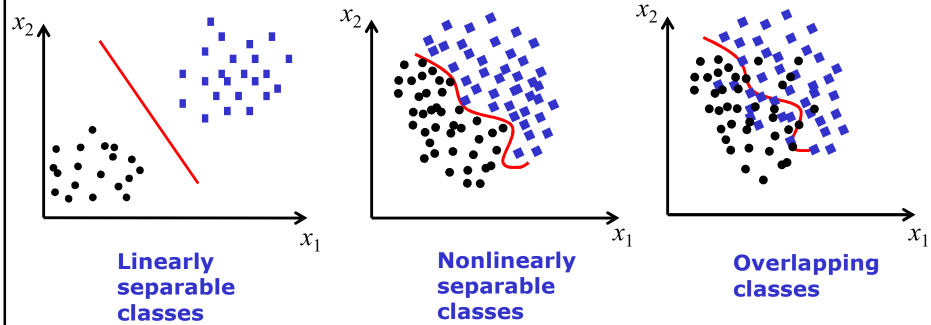
17

Step2: Classification (Testing Phase)



18

Pattern Classification Problems



19

- 1, 2, 3, 4, 5, ?, …, 24, 25, 26, 27, ?
- 1, 3, 5, 7, 9, ?, …, 25, 27, 29, 31, ?
- 2, 3, 5, 7, 11, ?, …, 29, 31, 37, 41, ?
- 1, 4, 9, 16, 25, ?, …, 121, 144, 169, ?
- 1, 2, 4, 8, 16, 32, ?,…, 1024, 2048, 4096, ?
- 1, 1, 2, 3, 5, 8, ?, …, 55, 89, 144, 233, ?
- 1, 1, 2, 4, 7, 13, ?, 44, 81, 149, 274, 504, ?
- 3, 5, 12, 24, 41, ?, …., 201, 248, 300, 357, ?
- 1, 6, 19, 42, 59, ?, …, 95, 117, 156, 191, ?

20

- 1, 2, 3, 4, 5, 6, …, 24, 25, 26, 27, 28
- 1, 3, 5, 7, 9, 11, …, 25, 27, 29, 31, 33
- 2, 3, 5, 7, 11, 13, …, 29, 31, 37, 41, 43
- 1, 4, 9, 16, 25, 36, …, 121, 144, 169, 196
- 1, 2, 4, 8, 16, 32, 64,…, 1024, 2048, 4096, 8192
- 1, 1, 2, 3, 5, 8, 13, …, 55, 89, 144, 233, 377
- 1, 1, 2, 4, 7, 13, 24, 44, 81, 149, 274, 504, 927
- 3, 5, 12, 24, 41, 63, ….., 201, 248, 300, 357, 419
  (2, 7, 12, 17, 22, 27, 32, 37, 42, 47, 52, 57, 62)
- 1, 6, 19, 42, 59, ?, …, 95, 117, 156, 191, ?

- **Pattern: Any regularity or structure in data or source of data**
- **Pattern Analysis: Automatic discovery of patterns in data**

21

Image Classification

Tiger

Giraffe

Horse

Bear

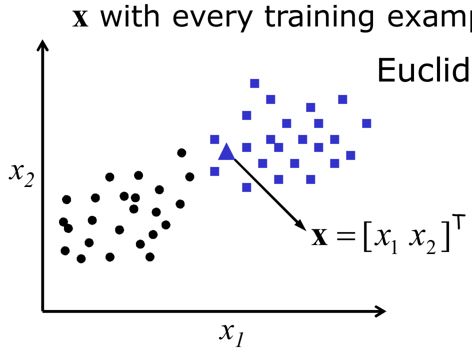
22

Scene Image Classification

Tall building	Inside city	Street	Highway	Coast	Open country	Mountain	Forest

Nearest-Neighbour Method

- Training data with N samples: $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$,
  $\mathbf{x}_n \in \mathbb{R}^d$ and $y_n \in \{1, 2, \ldots, M\}$
  - d: dimension of input example
  - M: Number of classes
- Step 1: Compute Euclidian distance for a test example $\mathbf{x}$ with every training examples, $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$, …, $\mathbf{x}_N$

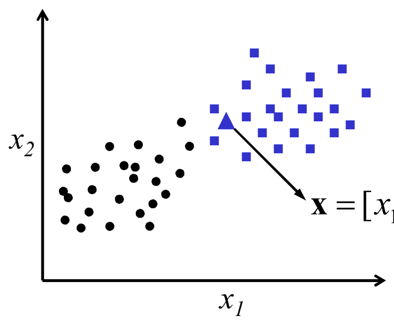


$$\text{Euclidean distance} = \|\mathbf{x}_n - \mathbf{x}\|$$
$$= \sqrt{(\mathbf{x}_n - \mathbf{x})^\top (\mathbf{x}_n - \mathbf{x})}$$
$$= \sqrt{\sum_{i=1}^{d} (x_{ni} - x_i)^2}$$

Nearest-Neighbour Method

- Training data: $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$,

  $\mathbf{x}_n \in \mathbb{R}^d$ and $y_n \in \{1, 2, \ldots, M\}$
  - d: dimension of input example
  - M: Number of classes
- Step 1: Compute Euclidian distance for a test example $\mathbf{x}$ with every training examples, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_N$



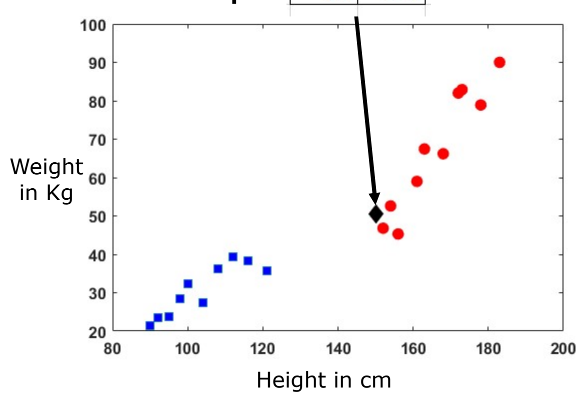
- Step 2: Sort the examples in the training set in the ascending order of the distance to $\mathbf{x}$
- Step 3: Assign the class of the training example with the minimum distance to the test example, $\mathbf{x}$

25

Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

Height	Weight	Class	ED
90	21.5	0	66.68
95	23.67	0	61.24
100	32.45	0	53.19
116	38.21	0	36.19
98	28.43	0	56.53
108	36.32	0	44.36
104	27.38	0	51.53
112	39.28	0	39.65
121	35.8	0	32.56
92	23.56	0	63.99
152	46.8	1	4.294
178	78.9	1	39.81
163	67.45	1	21.28
173	82.9	1	39.65
154	52.6	1	4.472
168	66.2	1	23.82
183	90	1	51.39
172	82	1	38.34
156	45.3	1	8.006
161	59	1	13.84

Test Example: 150 50.6



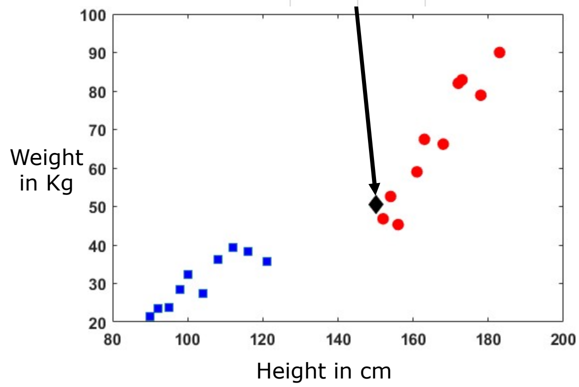
- Step 1: Compute Euclidian distance (ED) will each training examples

26

Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

Height	Weight	Class	ED
90	21.5	0	66.68
95	23.67	0	61.24
100	32.45	0	53.19
116	38.21	0	36.19
98	28.43	0	56.53
108	36.32	0	44.36
104	27.38	0	51.53
112	39.28	0	39.65
121	35.8	0	32.56
92	23.56	0	63.99
152	46.8	1	4.294
178	78.9	1	39.81
163	67.45	1	21.28
173	82.9	1	39.65
154	52.6	1	4.472
168	66.2	1	23.82
183	90	1	51.39
172	82	1	38.34
156	45.3	1	8.006
161	59	1	13.84

Test Example: 150 50.6



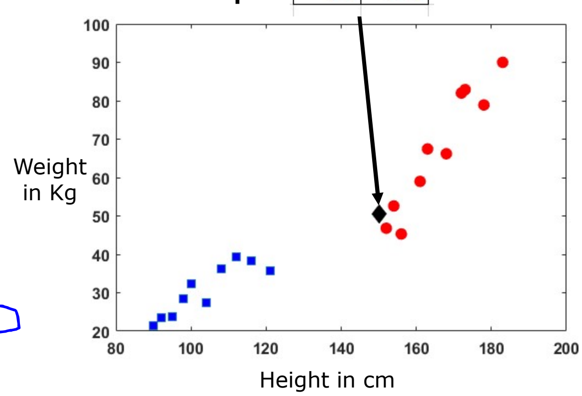
- Step 2: Sort the examples in the training set in the ascending order of the distance to test example

27

Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

Height	Weight	Class	ED
90	21.5	0	66.68
95	23.67	0	61.24
100	32.45	0	53.19
116	38.21	0	36.19
98	28.43	0	56.53
108	36.32	0	44.36
104	27.38	0	51.53
112	39.28	0	39.65
121	35.8	0	32.56
92	23.56	0	63.99
152	46.8	1	4.294
178	78.9	1	39.81
163	67.45	1	21.28
173	82.9	1	39.65
154	52.6	1	4.472
168	66.2	1	23.82
183	90	1	51.39
172	82	1	38.34
156	45.3	1	8.006
161	59	1	13.84

Test Example: 150 50.6



- Step 3: Assign the class of the training example with the minimum distance to the test example
  - Class: Adult

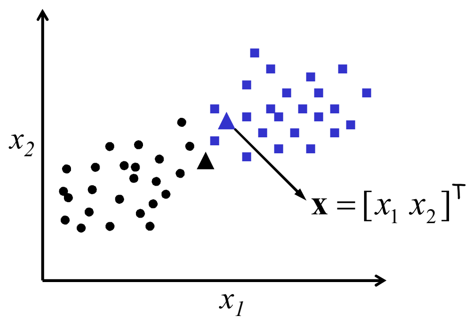
28

Nearest-Neighbour Method

- Training data: $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$,

$$\mathbf{x}_n \in \mathbb{R}^d \text{ and } y_n \in \{1, 2, \ldots, M\}$$

  - d: dimension of input example
  - M: Number of classes
- Step 1: Compute Euclidian distance for a test example $\mathbf{x}$ with every training examples, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_N$



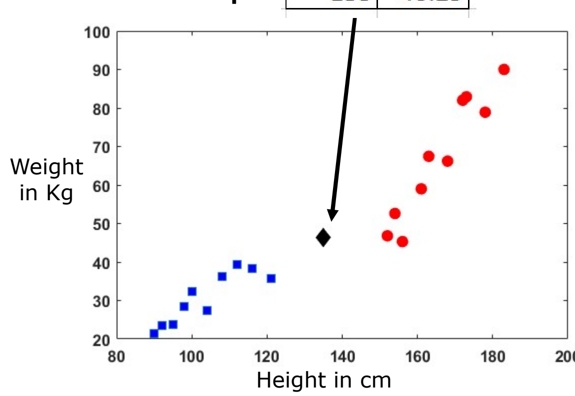
- Step 2: Sort the examples in the training set in the ascending order of the distance to $\mathbf{x}$
- Step 3: Assign the class of the training example with the minimum distance to the test example, $\mathbf{x}$

29

Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

Height	Weight	Class	ED
90	21.5	0	51.38
95	23.67	0	45.95
100	32.45	0	37.64
116	38.21	0	20.65
98	28.43	0	41.09
108	36.32	0	28.78
104	27.38	0	36.31
112	39.28	0	24.04
121	35.8	0	17.49
92	23.56	0	48.64
152	46.8	1	17.01
178	78.9	1	53.97
163	67.45	1	35.1
173	82.9	1	52.77
154	52.6	1	20.02
168	66.2	1	38.54
183	90	1	64.92
172	82	1	51.42
156	45.3	1	21.02
161	59	1	28.94

Test Example: | 135 | 46.29 |



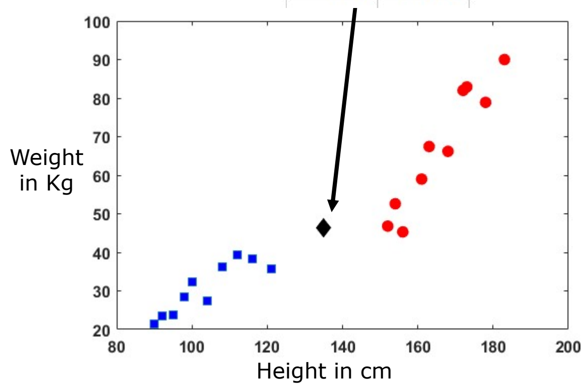
- Step 1: Compute Euclidian distance (ED) will each training examples

30

Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

Height	Weight	Class	ED
90	21.5	0	51.38
95	23.67	0	45.95
100	32.45	0	37.64
116	38.21	0	20.65
98	28.43	0	41.09
108	36.32	0	28.78
104	27.38	0	36.31
112	39.28	0	24.04
121	35.8	0	17.49
92	23.56	0	48.64
152	46.8	1	17.01
178	78.9	1	53.97
163	67.45	1	35.1
173	82.9	1	52.77
154	52.6	1	20.02
168	66.2	1	38.54
183	90	1	64.92
172	82	1	51.42
156	45.3	1	21.02
161	59	1	28.94

Test Example: | 135 | 46.29 |



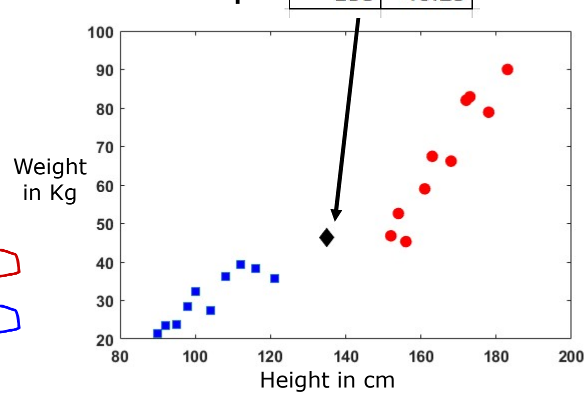
- Step 2: Sort the examples in the training set in the ascending order of the distance to test example

31

Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

Height	Weight	Class	ED
90	21.5	0	51.38
95	23.67	0	45.95
100	32.45	0	37.64
116	38.21	0	20.65
98	28.43	0	41.09
108	36.32	0	28.78
104	27.38	0	36.31
112	39.28	0	24.04
121	35.8	0	17.49
92	23.56	0	48.64
152	46.8	1	17.01
178	78.9	1	53.97
163	67.45	1	35.1
173	82.9	1	52.77
154	52.6	1	20.02
168	66.2	1	38.54
183	90	1	64.92
172	82	1	51.42
156	45.3	1	21.02
161	59	1	28.94

Test Example: | 135 | 46.29 |

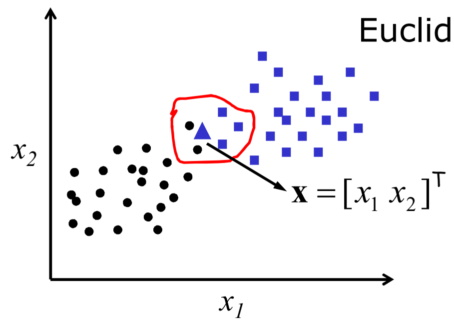


- Step 3: Assign the class of the training example with the minimum distance to the test example
  - Class: Adult

32

K-Nearest Neighbours (K-NN) Method

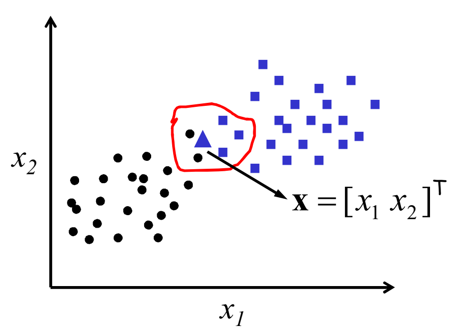
- Consider the class labels of the K training examples nearest to the test example
- Step 1: Compute Euclidian distance for a test example $\mathbf{x}$ with every training examples, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_N$



$$\text{Euclidean distance} = \|\mathbf{x}_n - \mathbf{x}\|$$
$$= \sqrt{(\mathbf{x}_n - \mathbf{x})^{\mathsf{T}}(\mathbf{x}_n - \mathbf{x})}$$
$$= \sqrt{\sum_{i=1}^{d}(x_{ni} - x_i)^2}$$

K-Nearest Neighbours (K-NN) Method

- Consider the class labels of the K training examples nearest to the test example
- Step 1: Compute Euclidian distance for a test example $\mathbf{x}$ with every training examples, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_N$

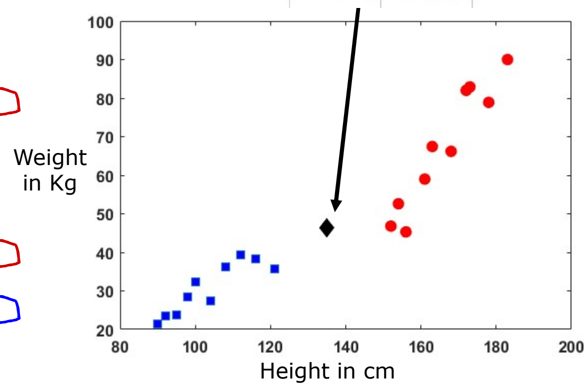


- Step 2: Sort the examples in the training set in the ascending order of the distance to $\mathbf{x}$
- Step 3: Choose the first K examples in the sorted list
  - K is the number of neighbours for text example
- Step 4: Test example is assigned the most common class among its K neighbours

Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

Height	Weight	Class	ED
90	21.5	0	51.38
95	23.67	0	45.95
100	32.45	0	37.64
116	38.21	0	20.65
98	28.43	0	41.09
108	36.32	0	28.78
104	27.38	0	36.31
112	39.28	0	24.04
121	35.8	0	17.49
92	23.56	0	48.64
152	46.8	1	17.01
178	78.9	1	53.97
163	67.45	1	35.1
173	82.9	1	52.77
154	52.6	1	20.02
168	66.2	1	38.54
183	90	1	64.92
172	82	1	51.42
156	45.3	1	21.02
161	59	1	28.94

Test Example: 135 | 46.29



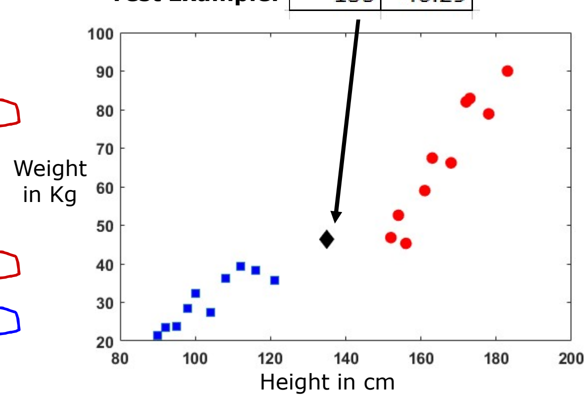
- Consider K=5
- Step 3: Choose the first K=5 examples in the sorted list

35

Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

Height	Weight	Class	ED
90	21.5	0	51.38
95	23.67	0	45.95
100	32.45	0	37.64
116	38.21	0	20.65
98	28.43	0	41.09
108	36.32	0	28.78
104	27.38	0	36.31
112	39.28	0	24.04
121	35.8	0	17.49
92	23.56	0	48.64
152	46.8	1	17.01
178	78.9	1	53.97
163	67.45	1	35.1
173	82.9	1	52.77
154	52.6	1	20.02
168	66.2	1	38.54
183	90	1	64.92
172	82	1	51.42
156	45.3	1	21.02
161	59	1	28.94

Test Example: 135 | 46.29



- Consider K=5
- Step 4: Test example is assigned the most common class among its K neighbours
  - Class: Adult

36

Determining K, Number of Neighbours

- This is determined experimentally
- Starting with K=1, test set is used to estimate the accuracy of the classifier
- This process is repeated each time by incrementing K to allow for more neighbour
- The K value that gives the maximum accuracy may be selected
- Preferably the value of K should be an odd number.

37

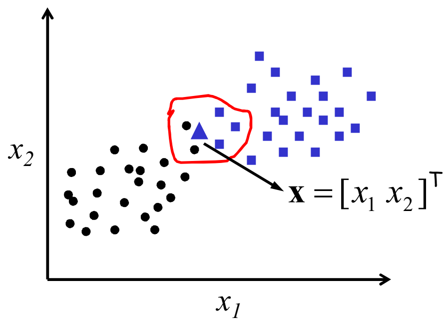
Data Normalization

- Since the distance measure is used, K-NN classifier require normalising the values of each attribute
- Normalising the training data:
  - Compute the minimum and maximum values of each of the attributes in the training data
  - Store the minimum and maximum values of each of the attributes
  - Perform the min-max normalization on training data set'
- Normalizing the test data:
  - Use the stored minimum and maximum values of each of the attributes from training set to normalise the test examples
- NOTE: Ensure that test examples are not causing out-of-bound error

38

K-Nearest Neighbours (K-NN) Method

- Consider the class labels of the K training examples nearest to the test example
- Step 1: Compute Euclidian distance for a test example $\mathbf{x}$ with every training examples, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_N$



- Step 2: Sort the examples in the training set in the ascending order of the distance to $\mathbf{x}$
- Step 3: Choose the first K examples in the sorted list
  - K is the number of neighbours for text example
- Step 4: Test example is assigned the most common class among its K neighbours

39

Learning from Data

- 1, 2, 3, 4, 5, ?, …, 24, 25, 26, 27, ?
- 1, 3, 5, 7, 9, ?, …, 25, 27, 29, 31, ?
- 2, 3, 5, 7, 11, ?, …, 29, 31, 37, 41, ?
- 1, 4, 9, 16, 25, ?, …, 121, 144, 169, ?
- 1, 2, 4, 8, 16, 32, ?,…, 1024, 2048, 4096, ?
- 1, 1, 2, 3, 5, 8, ?, …, 55, 89, 144, 233, ?
- 1, 1, 2, 4, 7, 13, ?, 44, 81, 149, 274, 504, ?
- 3, 5, 12, 24, 41, ?, …., 201, 248, 300, 357, ?
- 1, 6, 19, 42, 59, ?, …, 95, 117, 156, 191, ?

41

- 1, 2, 3, 4, 5, 6, …, 24, 25, 26, 27, 28
- 1, 3, 5, 7, 9, 11, …, 25, 27, 29, 31, 33
- 2, 3, 5, 7, 11, 13, …, 29, 31, 37, 41, 43
- 1, 4, 9, 16, 25, 36, …, 121, 144, 169, 196
- 1, 2, 4, 8, 16, 32, 64,…, 1024, 2048, 4096, 8192
- 1, 1, 2, 3, 5, 8, 13, …, 55, 89, 144, 233, 377
- 1, 1, 2, 4, 7, 13, 24, 44, 81, 149, 274, 504, 927
- 3, 5, 12, 24, 41, 63, ….., 201, 248, 300, 357, 419
  (2, 7, 12, 17, 22, 27, 32, 37, 42, 47, 52, 57, 62)
- 1, 6, 19, 42, 59, ?, …, 95, 117, 156, 191, ?

- **Pattern: Any regularity or structure in data or source of data**
- **Pattern Analysis: Automatic discovery of patterns in data**

42

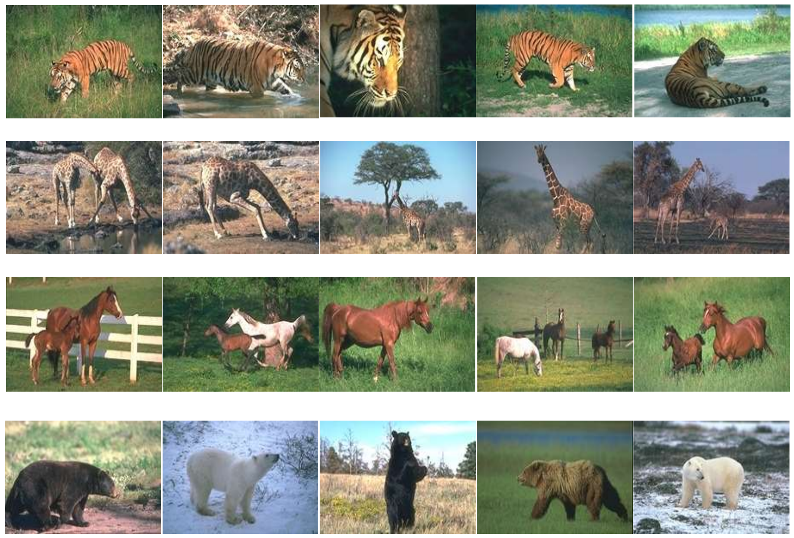
Image Classification

Tiger

Giraffe

Horse

Bear



43

Scene Image Classification

Tall building | Inside city | Street | Highway | Coast | Open country | Mountain | Forest

44

Machine Learning for Pattern Recognition

- Learning: Acquiring new knowledge or modifying the existing knowledge
- Knowledge: Familiarity with information present in data
- Learning by machines for pattern analysis: Acquisition of knowledge from data to discover patterns in data
- Data-driven techniques for learning by machines: Learning from examples (Training of models)
- Generalization ability of learning machines: Performance of trained models on new (test) data
- Target of learning techniques: Good generalization ability
- Learning techniques: Estimation of parameters of models

45

Lazy Learning : Learning from Neighbours

- The K nearest neighbour classifier is an example of lazy learner
- Lazy learning waits until the last minute before doing any model construction to classify test example
- When the training examples are given, a lazy learner simply stores them and waits until it is given a test example
- When it sees the test example, then it classify based on its similarity to the stored training examples
- Since the lazy learns stores the training examples or instances, they also called instance based learners
- Disadvantages:
  - Making classification or prediction is computationally intensive
  - Require efficient huge storage techniques when the training samples are huge

46

Data Preparation for the Classification

- Divide the data into training set and test set
  - Example:
    - Training data contain 70% of samples from each class
    - Test data contain remaining 30% of samples from each class

Data Preparation for the Classification using K-Nearest Classifier

- Suppose the data set has 3000 samples
- Each sample is belonging to one of the 3 classes
- Suppose each class has 1000 samples
  - Step1: From class1, 70% i.e. 700 samples considered as training samples and remaining 30% i.e. 300 samples are considered as test samples
  - Step2: From class2, 70% i.e. 700 samples considered as training samples and remaining 30% i.e. 300 samples are considered as test samples
  - Step3: From class3, 70% i.e. 700 samples considered as training samples and remaining 30% i.e. 300 samples are considered as test samples
  - Step4: Combine training examples from each class
    - Training set now contain 700+700+700=2100 samples
  - Step5: Combine test examples from each class
    - Test set now contain 300+300+300=900 samples

Performance Evaluation for Classification

Confusion Matrix

		Actual Class	
Predicted Class		Class1 (Positive)	Class2 (Negative)
	Class1 (Positive)	True Positive	False Positive
	Class2 (Negative)	False Negative	True Negative

- True Positive: Number of test samples correctly predicted as positive class.
- True Negative: Number of test samples correctly predicted as negative class.
- False Positive: Number of test samples predicted as positive class but actually belonging to negative class.
- False Negative: Number of test samples predicted as negative class but actually belonging to positive class.

Confusion Matrix

	Actual Class		
Predicted Class		Class1 (Positive)	Class2 (Negative)
	Class1 (Positive)	True Positive	False Positive
	Class2 (Negative)	False Negative	True Negative

Total test samples in class1

51

Confusion Matrix

	Actual Class		
Predicted Class		Class1 (Positive)	Class2 (Negative)
	Class1 (Positive)	True Positive	False Positive
	Class2 (Negative)	False Negative	True Negative

Total test samples in class2

52

Confusion Matrix

Predicted Class	Actual Class		
		Class1 (Positive)	Class2 (Negative)
	Class1 (Positive)	True Positive	False Positive
	Class2 (Negative)	False Negative	True Negative

Total test samples predicted as class1

Confusion Matrix

Predicted Class	Actual Class		
		Class1 (Positive)	Class2 (Negative)
	Class1 (Positive)	True Positive	False Positive
	Class2 (Negative)	False Negative	True Negative

Total test samples predicted as class2

Accuracy

- $$\text{Accuracy}(\%) = \frac{\text{Number of samples correctly classified (TP+TN)}}{\text{Total number of samples used for testing}} * 100$$

	Actual Class		
Predicted Class		Class1 (Positive)	Class2 (Negative)
	Class1 (Positive)	True Positive	False Positive
	Class2 (Negative)	False Negative	True Negative

55

Confusion Matrix - Multiclass

	Actual Class			
Predicted Class		Class1	Class2	Class3
	Class1	C11	C21	C31
	Class2	C12	C22	C32
	Class2	C13	C23	C33

- True Positive: Number of test samples correctly predicted as positive class (C11).
- True Negative: Number of test samples correctly predicted as negative class (C22+C33).
- False Positive: Number of test samples predicted as positive class but actually belonging to negative class (C21+C31)
- False Negative: Number of test samples predicted as negative class but actually belonging to positive class (C12+C13)

56

Confusion Matrix - Multiclass

		Actual Class			
		Class1	Class2	Class3	
Predicted Class	Class1	C11	C21	C31	Total samples predicted as class1
	Class2	C12	C22	C32	Total samples predicted as class2
	Class2	C13	C23	C33	Total samples predicted as class3
Total		Total samples in class1	Total samples in class2	Total samples in class3	

Total samples used for testing

57

Accuracy of Multiclass Classification

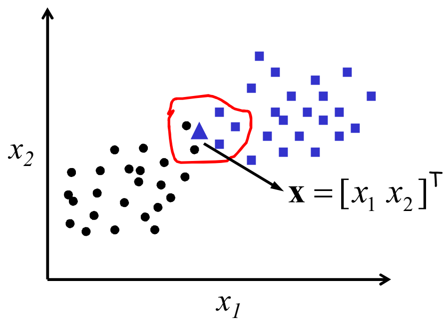
- $$\text{Accuracy}(\%) = \frac{\text{Number of samples correctly classified (TP+TN)}}{\text{Total number of samples used for testing}} * 100$$

		Actual Class		
		Class1	Class2	Class3
Predicted Class	Class1	C11	C21	C31
	Class2	C12	C22	C32
	Class2	C13	C23	C33

58

K-Nearest Neighbours (K-NN) Method

- Consider the class labels of the K training examples nearest to the test example
- Step 1: Compute Euclidian distance for a test example $\mathbf{x}$ with every training examples, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_N$



- Step 2: Sort the examples in the training set in the ascending order of the distance to $\mathbf{x}$
- Step 3: Choose the first K examples in the sorted list
  - K is the number of neighbours for text example
- Step 4: Test example is assigned the most common class among its K neighbours

59

Reference Templates Method

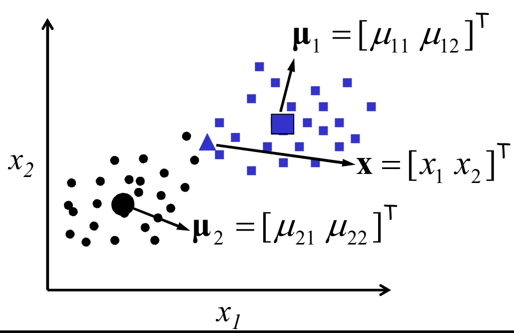
- Each class is represented by its reference templates
  - Mean of each data points of each class as reference template
- The class of the nearest reference template (mean) is assigned to the test pattern

$$\text{Euclidean distance} = \|\mathbf{x} - \boldsymbol{\mu}_i\|$$

$\boldsymbol{\mu}_i$: Mean vector of class i

$$= \sqrt{(\mathbf{x} - \boldsymbol{\mu}_i)^\mathsf{T}(\mathbf{x} - \boldsymbol{\mu}_i)}$$

$$= \sqrt{\sum_{j=1}^{d}(x_j - \mu_{ij})^2}$$



- Learning: Estimating first order statistics (mean) from the data of each class

60

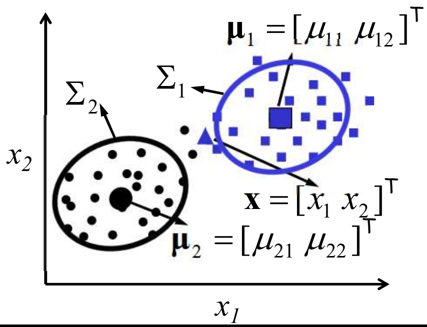
Modified Reference Templates Method

- Each class is represented by one or more reference templates
  - Mean and variance of data points of each class as reference template
- The class of the nearest reference templates is assigned to the test pattern

$$\text{Mahalanobis distance} = \frac{\|\mathbf{x}-\boldsymbol{\mu}_i\|}{\Sigma_i}$$

$\boldsymbol{\mu}_i$ & Σ_i : Mean vector and Covariance matrix of class i

$$= \sqrt{(\mathbf{x}-\boldsymbol{\mu}_i)^{\mathsf{T}}\Sigma_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)}$$

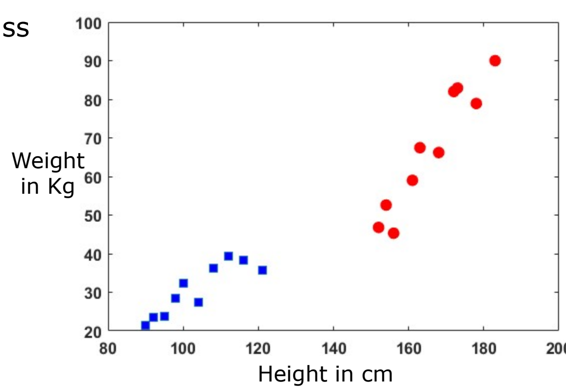


- Learning: Estimating
  - first order statistics (mean) and
  - Second order statistics (variance and covariance) from the data of each class

61

Probability Distribution

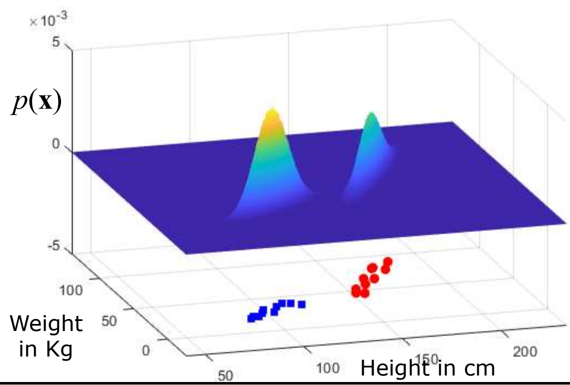
- Data of a class is represented by a probability distribution
- For a class whose data is considered to be forming a single cluster, it can be represented by a normal or Gaussian distribution
- Multivariate Gaussian distribution:
  - Adult-Child class



62

Probability Distribution

- Data of a class is represented by a probability distribution
- For a class whose data is considered to be forming a single cluster, it can be represented by a normal or Gaussian distribution
- Multivariate Gaussian distribution:
  - Adult-Child class
  - *Bivariate Gaussian distribution*
  - Each example is sampled from Gaussian distribution



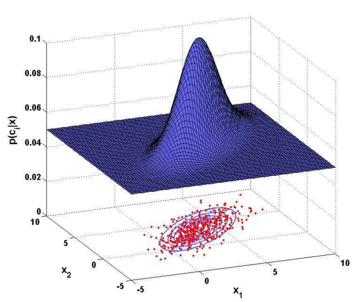
Multivariate Gaussian Distribution

- Data in d-dimensional space

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$= \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2} \underbrace{(\mathbf{x}-\boldsymbol{\mu})^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}_{\text{Mahalanobis distance}} \right)$$

  - $\boldsymbol{\mu}$ is the mean vector
  - $\boldsymbol{\Sigma}$ is the covariance matrix
- Bivariate Gaussian distribution: d=2



$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} E[x_1] \\ E[x_2] \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} E\left[(x_1-\mu_1)^2\right] & E[(x_1-\mu_1)(x_2-\mu_2)] \\ E[(x_2-\mu_2)(x_1-\mu_1)] & E\left[(x_2-\mu_2)^2\right] \end{bmatrix}$$

64

Bayes Classifier: Multivariate Data

- Let C_1, C_2, …, C_i, …, C_M be the M classes
  - Each class has N_i number of training examples
- Given: a test example $\mathbf{x}$
- Bayes decision rule:

Likelihood Prior

Posterior Probability of a class

$$P(C_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_i)P(C_i)}{P(\mathbf{x})}$$

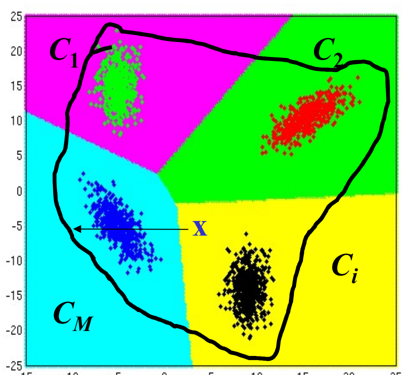
Evidence

  - Prior: Prior information of a class $P(C_i) = \frac{N_i}{N}$
    - where, N is total number of training examples
  - Evidence: Evidence/probability that $\mathbf{x}$ exists $p(\mathbf{x}) = \sum_{i=1}^{M} p(\mathbf{x} \mid C_i)P(C_i)$
    - Out of all the samples, what is the probability of the sample we are looking at
  - Likelihood follows the distribution of the data of a class

Class label for $\mathbf{x}$ = $\arg\max_i P(C_i \mid \mathbf{x}) \qquad i = 1, 2, \ldots, M$

65

Probability Theory and Bayes Rule



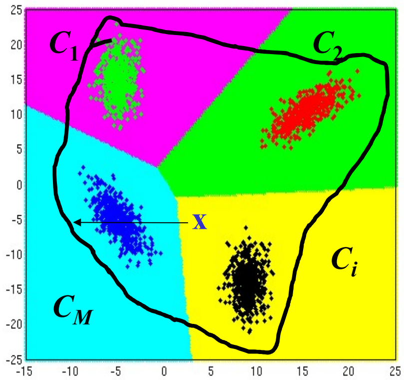
- The sample space is partitioned into C_1, C_2, …, C_i, …, C_M where each partitions are disjoint
  - Example:
    - Data space is sample space
    - Each class is my partitions
- Let $\mathbf{x}$ be an event defined in sample space
  - Example: A finite data points (training data) are the event $\mathbf{x}$
- $P(\mathbf{x})$: Total probability i.e. joint probability of $\mathbf{x}$ and C_i, $P(\mathbf{x}, C_i)$, for all i

$$P(\mathbf{x}) = \sum_{i=1}^{M} p(\mathbf{x}, C_i) = \sum_{i=1}^{M} p(\mathbf{x} \mid C_i)P(C_i)$$

- $P(\mathbf{x})$ is marginal probability – probability of $\mathbf{x}$ is obtained by marginalising over the events C_i

66

Probability Theory and Bayes Rule



- Conditional probability:

$$p(\mathbf{x} \mid C_i) = \frac{p(\mathbf{x}, C_i)}{P(C_i)} \qquad (1)$$

$$p(C_i \mid \mathbf{x}) = \frac{p(\mathbf{x}, C_i)}{P(\mathbf{x})} \qquad (2)$$

- From (1) and (2)

$$p(\mathbf{x} \mid C_i)P(C_i) = p(C_i \mid \mathbf{x})P(\mathbf{x})$$

- Bayes decision rule:

$$P(C_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_i)P(C_i)}{P(\mathbf{x})}$$

67

Bayes Classifier: Multivariate Data

- Data of a class is represented by a probability distribution
- Given: a test example $\mathbf{x}$
- Bayes decision rule:

Likelihood

Prior

Posterior Probability of a class

$$P(C_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_i)P(C_i)}{P(\mathbf{x})}$$

Evidence

  - Likelihood of a class follows the distribution of the data of a class
  - Computation of likelihood of a class depends on the distribution of the data and the parameters of that distribution
- Bayes decision rule can be given as $P(\boldsymbol{\theta}_i \mid \mathbf{x}) = \dfrac{p(\mathbf{x} \mid \boldsymbol{\theta}_i)P(C_i)}{P(\mathbf{x})}$
  - $\boldsymbol{\theta}_i$ is the parameters of the distribution of class C_i

68

Maximum Likelihood (ML) Method for Parameter Estimation

- Given: Training data for a class C_i: having N_i samples

$$\mathcal{D}_i = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_{Ni}\}, \mathbf{x}_n \in \mathbb{R}^d$$

- Data of a class is represented by parameter vector: $\boldsymbol{\theta}_i = [\theta_{i1}, \theta_{i2}, \ldots, \theta_{iK}]^{\mathsf{T}}$, of its distribution
- Unknown: $\boldsymbol{\theta}_i$
- Likelihood of training data (Total data likelihood) for a given $\boldsymbol{\theta}_i$:

$$p(\mathcal{D}_i \mid \boldsymbol{\theta}_i) = \prod_{n=1}^{N_i} p(\mathbf{x}_n \mid \boldsymbol{\theta}_i)$$

$$\mathcal{L}(\boldsymbol{\theta}_i) = \ln p(\mathcal{D}_i \mid \boldsymbol{\theta}_i) = \sum_{n=1}^{N_i} \ln p(\mathbf{x}_n \mid \boldsymbol{\theta}_i)$$

- Choose the parameters for which the total data likelihood (log likelihood) is maximum:

$$\boldsymbol{\theta}_{i_{\mathrm{ML}}} = \arg\max_{\boldsymbol{\theta}_i} \mathcal{L}(\boldsymbol{\theta}_i)$$

69

ML Method for Parameter Estimation of Multivariate Gaussian Distribution

- Given: Training data for a class C_i having N_i samples

$$\mathcal{D}_i = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_{Ni}\}, \mathbf{x}_n \in \mathbb{R}^d$$

- Data of a class is represented by parameter vector: $[\boldsymbol{\mu}_i \ \boldsymbol{\Sigma}_i]^{\mathsf{T}}$, of Gaussian distribution
- Unknown: $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$
- Likelihood of training data (Total data likelihood) for a given $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$:

$$p(\mathcal{D} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \prod_{n=1}^{N_i} p(\mathbf{x}_n \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

$$\mathcal{L}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \ln p(\mathcal{D}_i \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \sum_{n=1}^{N_i} \ln p(\mathbf{x}_n \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

- Choose the parameters for which the total data likelihood (log likelihood) is maximum:

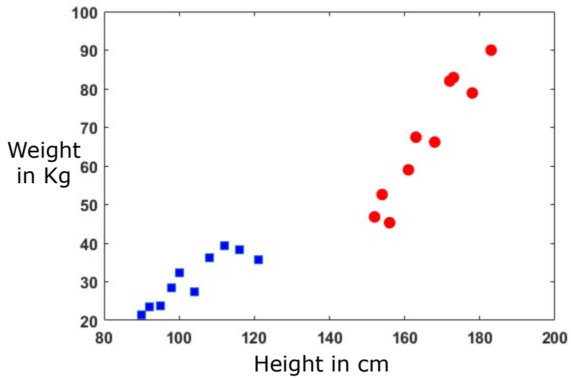
$$\boldsymbol{\mu}_{i_{\mathrm{ML}}}, \boldsymbol{\Sigma}_{i_{\mathrm{ML}}} = \arg\max_{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i} \mathcal{L}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

70

Illustration of ML Method: Training Set: Adult-Child

Height	Weight	Class
90	21.5	0
95	23.67	0
100	32.45	0
116	38.21	0
98	28.43	0
108	36.32	0
104	27.38	0
112	39.28	0
121	35.8	0
92	23.56	0
152	46.8	1
178	78.9	1
163	67.45	1
173	82.9	1
154	52.6	1
168	66.2	1
183	90	1
172	82	1
156	45.3	1
161	59	1

- Number of training examples (N) = 20
- Dimension of a training example = 2
- Class label attribute is 3rd dimension
- Class:
  - Child (0)
  - Adult (1)



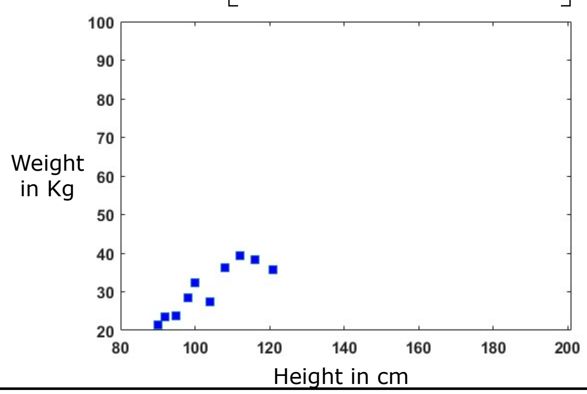
71

Illustration of ML Method: Child class

Height	Weight
90	21.5
95	23.67
100	32.45
116	38.21
98	28.43
108	36.32
104	27.38
112	39.28
121	35.8
92	23.56

- Number of training examples (N) = 20
- Dimension of a training example = 2
- Sample mean: **[103.6 30.66]**
- Sample covariance matrix**:**

$$\begin{bmatrix} \mathbf{109.3778} & \mathbf{61.3500} \\ \mathbf{61.3500} & \mathbf{43.5415} \end{bmatrix}$$



72

Illustration of ML Method: Child class

Height	Weight
90	21.5
95	23.67
100	32.45
116	38.21
98	28.43
108	36.32
104	27.38
112	39.28
121	35.8
92	23.56

- Covariance matrix value is fixed at **:**

$$\begin{bmatrix} \mathbf{109.3778} & \mathbf{61.3500} \\ \mathbf{61.3500} & \mathbf{43.5415} \end{bmatrix}$$

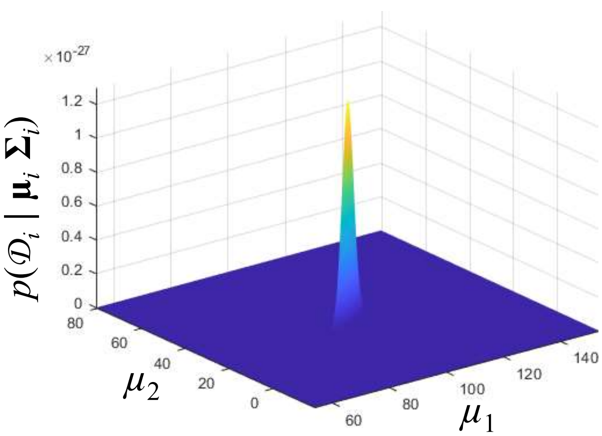
- Search the values for mean vector $\boldsymbol{\mu}=[\mu_1, \ \mu_2]^T$ that maximizes the total data likelihood
- Range of values for mean vectors to search:
  - 1000 equally sampled values from 53.6 to 153.6 for μ_1
  - 1000 equally sampled values from -20.66 to 80.66 for μ_2
- Compute the likelihood value for each of the 10,00,000 (1000 x 1000) values of the mean vectors

73

Illustration of ML Method: Child class

Height	Weight
90	21.5
95	23.67
100	32.45
116	38.21
98	28.43
108	36.32
104	27.38
112	39.28
121	35.8
92	23.56

- A maximum value for the likelihood is obtained for the value **[103.65 30.71]**
- This value is close to sample mean vector: **[103.6 30.66]**



ML Method for Parameter Estimation of Multivariate Gaussian Distribution

- Parameters of Gaussian distribution of class C_i : $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$
- Likelihood for a single example, $\mathbf{x}_n$:

$$p(\mathbf{x}_n \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_i|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_i)\right)$$

- Log likelihood for total training data of class C_i , $\mathcal{D}_i=\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$:

$$\mathcal{L}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \ln p(\mathcal{D}_i \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \ln \prod_{n=1}^{N_i} p(\mathbf{x}_n \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \sum_{n=1}^{N_i} \ln p(\mathbf{x}_n \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

$$= \sum_{n=1}^{N_i} -\frac{1}{2}\ln|\boldsymbol{\Sigma}_i| - \frac{d}{2}\ln 2\pi - \frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_i)$$

- Setting the derivatives of $\mathcal{L}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ w.r.t. $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ to zero, we get:

$$\boldsymbol{\mu}_{i_{\text{ML}}} = \frac{1}{N_i}\sum_{n=1}^{N_i} \mathbf{x}_n \qquad \boldsymbol{\Sigma}_{i_{\text{ML}}} = \frac{1}{N_i}\sum_{n=1}^{N_i} (\mathbf{x}_n - \boldsymbol{\mu}_{i_{\text{ML}}})(\mathbf{x}_n - \boldsymbol{\mu}_{i_{\text{ML}}})^\top$$

75

Bayes Classifier with Unimodal Gaussian Density – Training Process

- Let C_1, C_2, …, C_i, …, C_M be the M classes
- Let $\mathcal{D}_1$, $\mathcal{D}_2$, …, $\mathcal{D}_i$, …, $\mathcal{D}_M$ be the training data for M classes
- Estimate the parameters
  - $\boldsymbol{\theta}_1 = [\boldsymbol{\mu}_1\ \boldsymbol{\Sigma}_1]^\top$,
  - $\boldsymbol{\theta}_2 = [\boldsymbol{\mu}_2\ \boldsymbol{\Sigma}_2]^\top$,
  - …,
  - $\boldsymbol{\theta}_i = [\boldsymbol{\mu}_i\ \boldsymbol{\Sigma}_i]^\top$,
  - …,
  - $\boldsymbol{\theta}_M = [\boldsymbol{\mu}_M\ \boldsymbol{\Sigma}_M]^\top$ for each of the classes
- Number of parameters to be estimated for each class is dependent on dimensionality of the data space d
  - Number of parameters: $d + (d(d+1))/2$

76

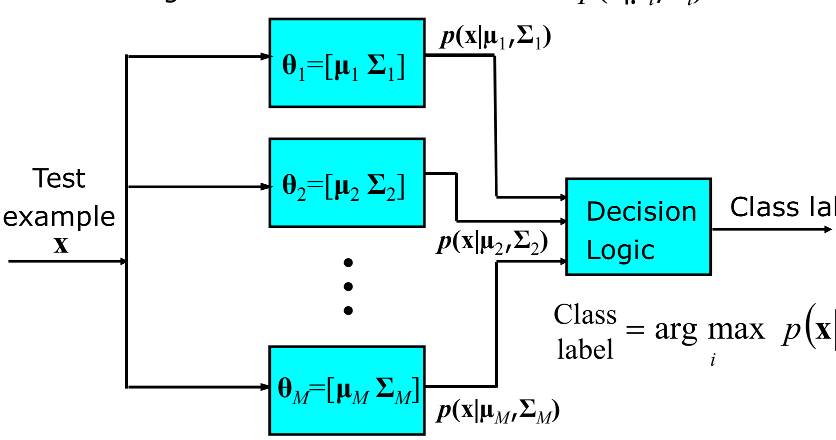
Bayes Classifier with Unimodal Gaussian Density – Training Process

- Let C_1, C_2, …, C_i, …, C_M be the M classes
- Let $\mathcal{D}_1$, $\mathcal{D}_2$, …, $\mathcal{D}_i$, …, $\mathcal{D}_M$ be the training data for M classes
- Compute sample mean vector and sample covariance matrix from training data of class 1, $\boldsymbol{\theta}_1 = [\boldsymbol{\mu}_1\ \boldsymbol{\Sigma}_1]^\top$
- Compute sample mean vector and sample covariance matrix from training data of class 2, $\boldsymbol{\theta}_2 = [\boldsymbol{\mu}_2\ \boldsymbol{\Sigma}_2]^\top$,
- …,
- Compute sample mean vector and sample covariance matrix from training data of class M, $\boldsymbol{\theta}_M = [\boldsymbol{\mu}_M\ \boldsymbol{\Sigma}_M]^\top$

77

Bayes Classifier with Unimodal Gaussian Density: Classification

- For a test example $\mathbf{x}$:
  - likelihood of $\mathbf{x}$ generated from each of the classes $p(\mathbf{x}|\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)$ is computed
  - Assign the label of class for which $p(\mathbf{x}|\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)$ is maximum



$$\text{Class label} = \arg\max_i \; p(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

78

Illustration of Bayes Classifier with Unimodal Gaussian Density : Adult(1)-Child(0) Classification

Height	Weight	Class
90	21.5	0
95	23.67	0
100	32.45	0
116	38.21	0
98	28.43	0
108	36.32	0
104	27.38	0
112	39.28	0
121	35.8	0
92	23.56	0
152	46.8	1
178	78.9	1
163	67.45	1
173	82.9	1
154	52.6	1
168	66.2	1
183	90	1
172	82	1
156	45.3	1
161	59	1

- Training Phase:
  - Compute sample mean vector and sample covariance matrix from training data of class 1 (Child)

$$\boldsymbol{\mu}_1 = [\mathbf{103.6000} \quad \mathbf{30.6600}]$$

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} \mathbf{109.3778} & \mathbf{61.3500} \\ \mathbf{61.3500} & \mathbf{43.5415} \end{bmatrix}$$

  - Compute sample mean vector and sample covariance matrix from training data of class 2 (Adult)

$$\boldsymbol{\mu}_2 = [\mathbf{166.0000} \quad \mathbf{67.1150}]$$

$$\boldsymbol{\Sigma}_2 = \begin{bmatrix} \mathbf{110.6667} & \mathbf{160.5278} \\ \mathbf{160.5278} & \mathbf{255.4911} \end{bmatrix}$$

79

Illustration of Bayes Classifier with Unimodal Gaussian Density : Adult(1)-Child(0) Classification

- Test phase: Classification
- Class1 (Child)

$$\boldsymbol{\mu}_1 = [\mathbf{103.6000} \quad \mathbf{30.6600}]$$

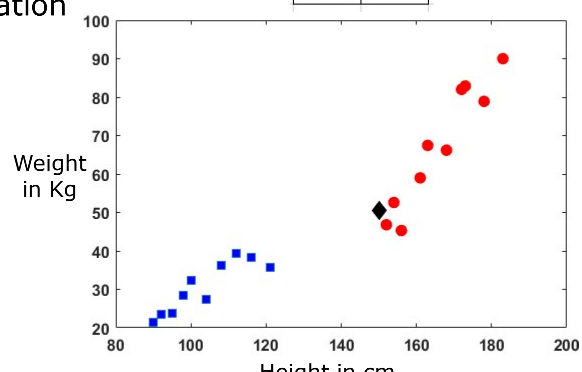
$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} \mathbf{109.3778} & \mathbf{61.3500} \\ \mathbf{61.3500} & \mathbf{43.5415} \end{bmatrix}$$

- Class 2 (Adult)

$$\boldsymbol{\mu}_2 = [\mathbf{166.0000} \quad \mathbf{67.1150}]$$

$$\boldsymbol{\Sigma}_2 = \begin{bmatrix} \mathbf{110.6667} & \mathbf{160.5278} \\ \mathbf{160.5278} & \mathbf{255.4911} \end{bmatrix}$$

Test Example, $\mathbf{x}$: | 150 | 50.6 |



- Compute likelihood of test sample, $\mathbf{x}$ with class 1 (Child)

$$p(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) = 3.5237\text{x}10^{-08}$$

- Compute likelihood of test sample, $\mathbf{x}$ with class 2 (Adult)

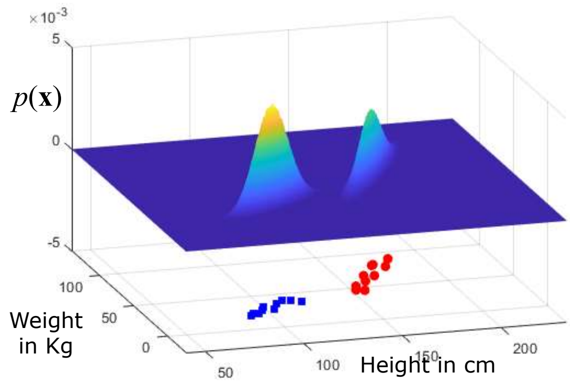
$$p(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) = 3.7177\text{x}10^{-04}$$

Class label of $\mathbf{x}$ = Adult

80

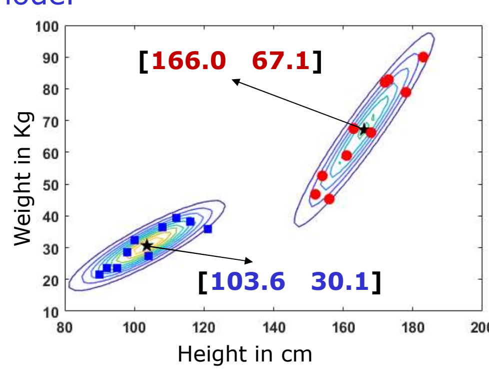
Summary: Bayes Classifier with Unimodal Gaussian Density

- The relation between examples and class can be captured in a statistical model
  - Bayes classifier
- Statistical model:
  - Unimodal Gaussian density
    - Univariate
    - Multivariate



Summary: Bayes Classifier with Unimodal Gaussian Density

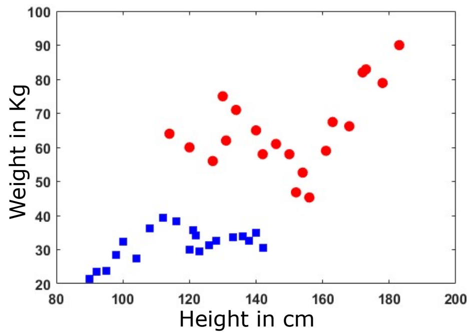
- The relation between examples and class can be captured in a statistical model
  - Bayes classifier
- Statistical model:
  - Unimodal Gaussian density
    - Univariate
    - Multivariate

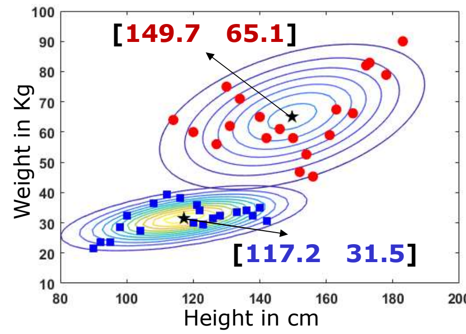


- The real world data need not be unimodal
  - The shape of the density can be arbitrary
  - Bayes classifier?
- Multimodal density function

82

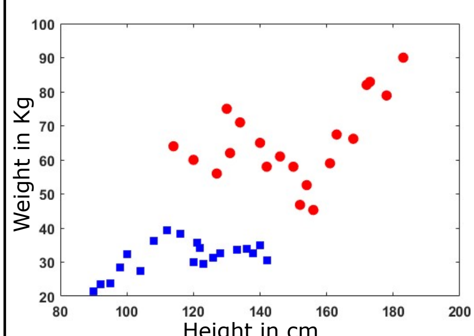
Adult-Child Data

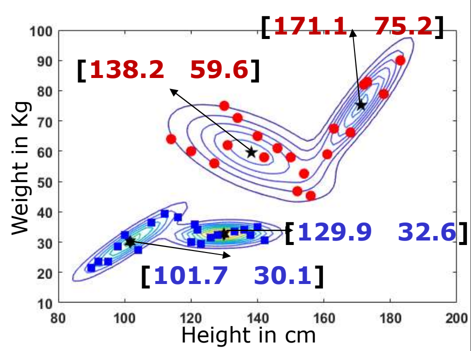




Multimodal Distribution: Adult-Child Data

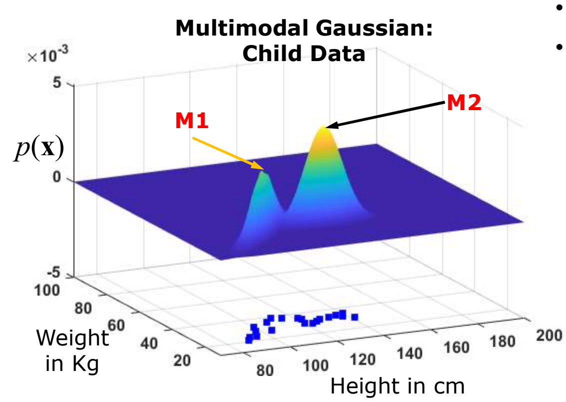
- For a class whose data is considered to have multiple clusters, the probability distribution is multimodal





Multimodal Distribution: Adult-Child Data

- For a class whose data is considered to have multiple clusters, the probability distribution is multimodal



- **M1**: Cluster 1 (mode 1)
- **M2**: Cluster 2 (mode 2)

85

Multimodal Gaussian Distribution: Gaussian Mixture Model

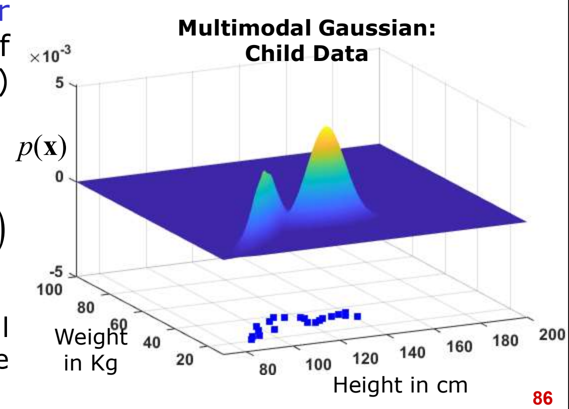
- Given: Training data for a class C_i: having N_i samples

$$\mathcal{D}_i = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_{N_i}\}, \ \mathbf{x}_n \in \mathbb{R}^d$$

- Gaussian mixture model (GMM): to represent a multimodal distribution
- GMM is a linear superposition of multiple (Q) *Gaussian components*:

$$p(\mathbf{x}|C_i) = \sum_{q=1}^{Q} w_q \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\right)$$

  - The overall envelope of the curve



86

Gaussian Mixture Model (GMM)

- GMM is a linear superposition of multiple Gaussians:

$$p(\mathbf{x}|C_i) = \sum_{q=1}^{Q} w_q \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\right)$$

- For a d-dimensional feature vector representation of data, the parameters of GMM are
  - Mixture coefficients, w_q , $q = 1,2,\dots, Q$
    - *Mixture weight or Strength of each clusters (or mixtures or modes)*
    - Property: $\sum_{q=1}^{Q} w_q = 1$
  - d-dimensional mean vector, $\boldsymbol{\mu}_q$, $q = 1,2,\dots, Q$
  - dxd size covariance matrices, $\boldsymbol{\Sigma}_q$, $q = 1,2,\dots, Q$
- Training process objective: To estimate the parameters of the GMM

87

Parameter Estimation of GMM: Incomplete Data Problem

- Given: Training data for a class C_i: having N_i samples

$$\mathcal{D}_i = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n, \dots, \mathbf{x}_{Ni}\}, \; \mathbf{x}_n \in \mathbb{R}^d$$

- Known: Training data is multimodal in nature
- Unknown: identity of the cluster (or mixture) of these training data points
- Incomplete data problem:
  - Given is only data points but not their identity (i.e. to which cluster it belongs)
  - Hidden (latent) information: Identity of data points to the cluster

88

Parameter Estimation of GMM: Incomplete Data Problem

- *If identity (latent information) is given, how to estimate parameters of GMM?*
- Apply maximum likelihood method to estimate the parameters of each of the q mixtures ($\boldsymbol{\mu}_q$ and $\boldsymbol{\Sigma}_q$)
- Mixture coefficients, w_q is computed as

$$w_q = \frac{N_{iq}}{N_i}$$

- N_{iq}: Number of data points in cluster q
- N_i: Number of data points in class C_i

- In practice, we do not have this information
- Goal of parameter estimation: To find the best possible values of parameters of GMM such that the total likelihood of data is maximized
  – Maximum likelihood method for training a GMM: ***Expectation-Maximization (EM) method***

89

Expectation-Maximization (EM) for GMMs

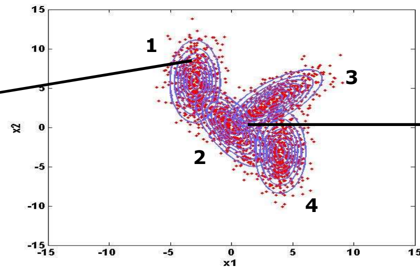
- An elegant and powerful method for finding the maximum likelihood solution for a model with latent variables
- Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters
  1. Initialize the means $\boldsymbol{\mu}_q$, covariances $\boldsymbol{\Sigma}_q$ and mixing coefficients w_q, and evaluate the initial value of the log likelihood
  2. **E-step**: Evaluate the responsibilities $\gamma_q(\mathbf{x})$ using the current parameter values

90

EM Method – Responsibility Term

- A quantity that plays an important role is the responsibility term, $\gamma_q(\mathbf{x})$
- It is given by
$$\gamma_q(\mathbf{x}) = \frac{w_q \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)}{\sum_{j=1}^{Q} w_j \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$
- w_q : mixture coefficient or prior probability of component q,
- $\gamma_q(\mathbf{x})$ gives the posterior probability of the component q for the observation $\mathbf{x}$

$\gamma_1(\mathbf{x}_n) = 0.99$
$\gamma_2(\mathbf{x}_n) = 0.01$
$\gamma_3(\mathbf{x}_n) = 0.00$
$\gamma_4(\mathbf{x}_n) = 0.00$



$\gamma_1(\mathbf{x}_m) = 0.08$
$\gamma_2(\mathbf{x}_m) = 0.42$
$\gamma_3(\mathbf{x}_m) = 0.34$
$\gamma_4(\mathbf{x}_m) = 0.16$

91

Expectation-Maximization (EM) for GMMs

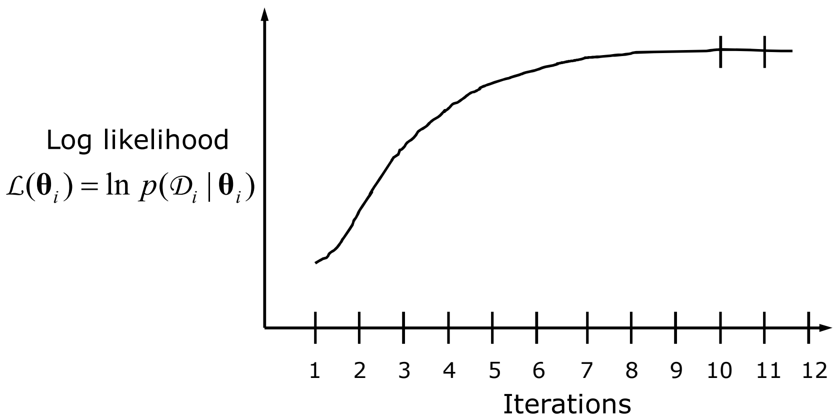
- Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters

1. Initialize the means $\boldsymbol{\mu}_q$, covariances $\boldsymbol{\Sigma}_q$ and mixing coefficients w_q, and evaluate the initial value of the log likelihood
2. **E-step**: Evaluate the responsibilities $\gamma_q(\mathbf{x})$ using the current parameter values
3. **M-step**: Re-estimate the parameters $\boldsymbol{\mu}_q^{new}$, $\boldsymbol{\Sigma}_q^{new}$ and w_q^{new} using the current responsibilities
4. Evaluate the log likelihood and check for convergence of the log likelihood
   - If the convergence criterion is not satisfied return to step 2

92

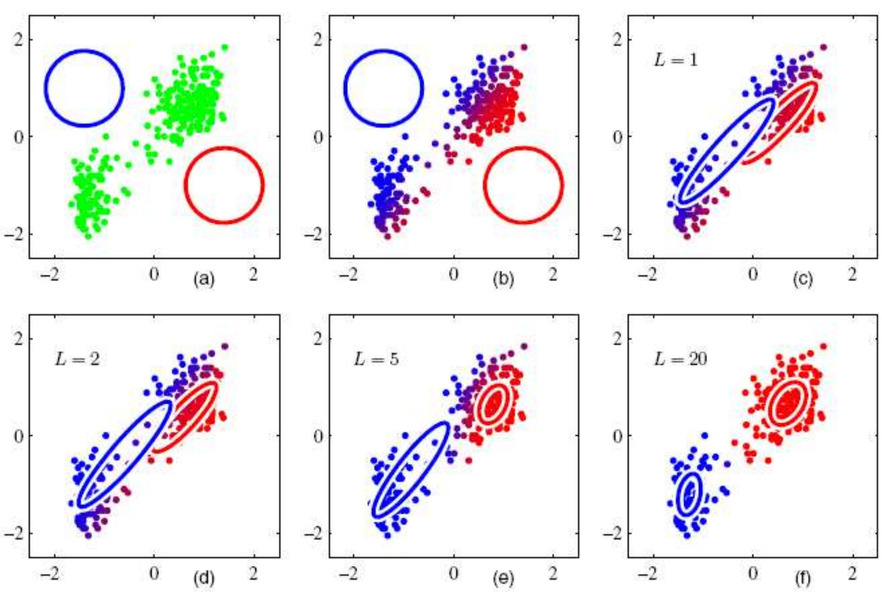
Expectation-Maximization (EM) for GMMs

- Convergence criterion: Difference between log likelihoods of successive iterations fall below a threshold (E.g. 10^{-3})



93

Illustration of Parameter Estimation



C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

94

Bayes Classifier: Multimodal Data

- Let C_1, C_2, …, C_i, …, C_M be the M classes
- Given: a test example $\mathbf{x}$
- Bayes decision rule:

Likelihood

Prior

Posterior Probability

$$P(C_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_i) P(C_i)}{p(\mathbf{x})}$$

Evidence

$$p(\mathbf{x} \mid C_i) = \sum_{q=1}^{Q} w_q \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\right)$$

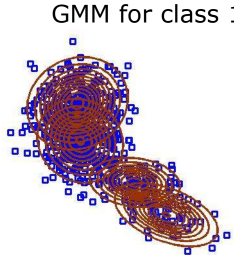
Class label for $\mathbf{x}$ = $\arg\max_i P(C_i \mid \mathbf{x})$

95

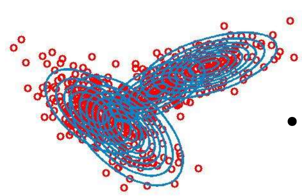
Bayes Classifier with Multimodal Gaussian Density (GMM) – Training Process

- Let C_1, C_2, …, C_i, …, C_M be the M classes
- Let $\mathcal{D}_1$, $\mathcal{D}_2$, …, $\mathcal{D}_i$, …, $\mathcal{D}_M$ be the training data for M classes
- Build GMM ($\boldsymbol{\lambda}$) for each of the classes

GMM for class 1, $\boldsymbol{\lambda}_1$

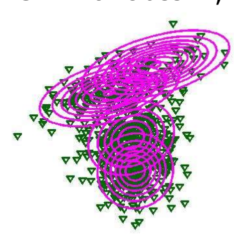


GMM for class 2, $\boldsymbol{\lambda}_2$



• • •

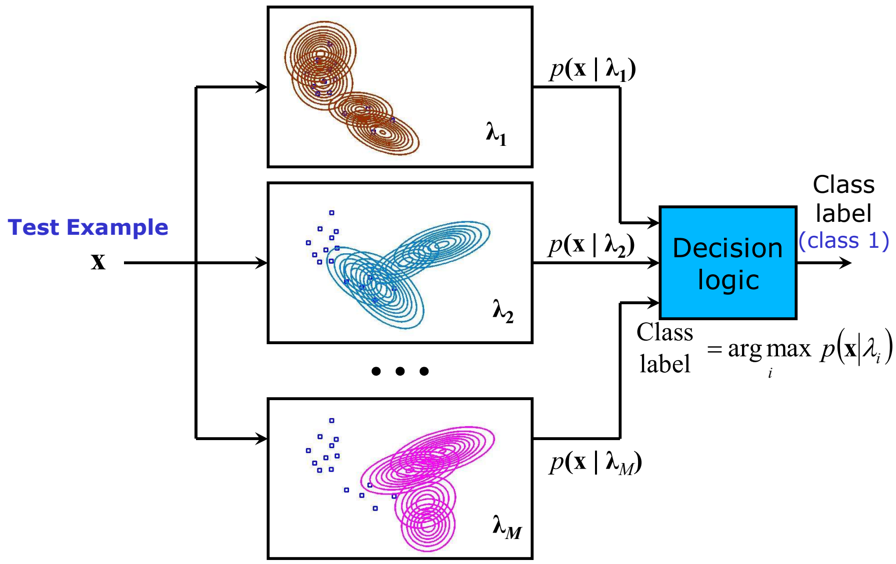
GMM for class M, $\boldsymbol{\lambda}_M$



$$\textbf{GMM for Class } i, \boldsymbol{\lambda}_i = \left[w_q, \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\right]_{q=1}^{Q}$$

96

Bayes Classifier with Multimodal Gaussian Density (GMM) – Classification



97

Determining Q, Number of Gaussian Components

- This is determined experimentally
- Starting with Q=1, test set is used to estimate the accuracy of the Bayes classifier
- This process is repeated each time by incrementing Q to allow for more Gaussian components
- The GMM with Q components that gives the maximum accuracy may be selected

98

Bayes Classifier with Gaussian Mixture Models – Summary

- Multimodal probability distribution for each class is represented by a Gaussian mixture model.
- GMM is a powerful way of modeling data
- Using GMM, a data of any arbitrary shaped distribution can be modeled
- In GMM, number of parameters to be estimated for each class is dependent on:
  - Dimensionality of the data space d
  - Number of Gaussian mixtures Q

    Qxd + Qx$(d(d+1))/2 + Q$
- For large values of d and Q, the number of examples required to estimate the parameters properly will be large.
- *When the estimated class-conditional densities are the same as the true densities, Bayes classifier gives minimum classification error*

99

Text Books

1. J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Third Edition, Morgan Kaufmann Publishers, 2011.

2. S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 2009.

3. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

100