Introduction

CS6240 Multimedia Analysis

Leow Wee Kheng

Department of Computer Science
School of Computing
National University of Singapore

Introduction

What can be analyzed about multimedia objects such as images, video, sound, text documents?

Consider these two images:

We can ask a computer the following questions:

1. What are the colors in the images?
2. What are the objects in the images?
3. What are the similarities between the images?
4. What are the differences between the images?
5. How to obtain the boundaries of the objects in the images?
6. What are the 3D orientations of the objects in the images?
7. How to change the shape of one object to match the shape of the other?

- Q1 is about low-level features, i.e., colors.
  It is the easiest to answer.
- Q2 is about object recognition; usually difficult to answer.
- Q1, Q2 are examples of classification problems.
- Q3 and Q4 are examples of matching or comparison problems.
- Q5 is a segmentation problem; usually difficult to solve.
- Q6 is a pose estimation problem; requires knowledge about the objects.
- Q7 is a shape deformation problem; can be very difficult.

Basic Problem Solving Stages

1. Analyze the problem.
   Analyze the characteristics of the inputs and the desired outputs.
2. Define the problem using mathematical notations.
   Write down the characteristics of the inputs and desired outputs, and the relationship between them.
3. Develop an algorithm to solve the problem.
   Given the inputs, the algorithm produces outputs that satisfy the problem definition.
   Usually involve computing some kind of difference or error.
4. Measure and evaluate the performance of the algorithm.

Three Main Approaches

1. Vector-space approach:
   - Represent each multimedia object as a vector of feature values in a linear vector space, i.e., Euclidean space.
   - Each feature corresponds to a dimension of the vector space.
   - Compute difference using Euclidean distance in the vector space.
2. Exemplar-based approach:
   - Represent each object as a set of raw or minimally processed feature values, or a discrete structure.
   - The space of objects may not be a vector space.
   - Difference measure may not be Euclidean.
3. Model-based approach:
   - Represent each object as a set of feature values,
usually with domain knowledge about the object.
   - The space of objects may not be a vector space.
   - Difference measure may not be Euclidean.

Vector-Space Approach

A Euclidean n-space [3], denoted $\mathbb{R}^n$, consists of all ordered n-tuples of real numbers:

$$\mathbb{R}^n = \{\mathbf{x} = (x_1, \ldots, x_n) \,|\, x_1, \ldots, x_n \in \mathbb{R}\}\,.$$

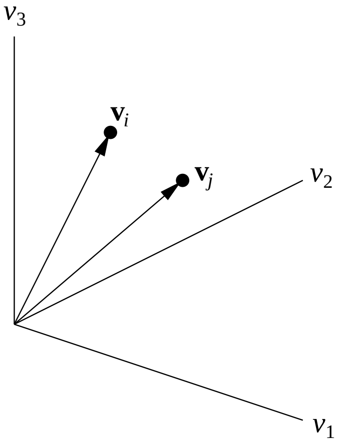
- $\mathbf{x}$ is a point in $\mathbb{R}^n$.
- addition:

$$(x_1, \ldots, x_n) + (y_1, \ldots, y_n) = (x_1 + y_1, \ldots, x_n + y_n)$$

- scalar multiplication:

$$a(x_1, x_2, \ldots, x_n) = (ax_1, ax_2, \ldots, ax_n) \;\; \text{for } a \in \mathbb{R}.$$

Example: 3-dimensional Euclidean space.



A real vector space $\mathcal{V}$ is a set of elements called vectors, with given operations of vector addition $+ : \mathcal{V} \times \mathcal{V} \to \mathcal{V}$ and scalar multiplication $\cdot : \mathbb{R} \times \mathcal{V} \to \mathcal{V}$ such that:

1. commutativity: $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
2. associativity: $(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$
3. zero vector: $\mathbf{v} + \mathbf{0} = \mathbf{v}$
4. negatives: $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$
5. distributivity: $a \cdot (\mathbf{u} + \mathbf{v}) = a \cdot \mathbf{u} + a \cdot \mathbf{v}$
6. multiplicative identity: $1 \cdot \mathbf{v} = \mathbf{v}$

Notes:

- Usually we omit the $\cdot$ for scalar multiplication.
- Euclidean n-space is a n-dimensional vector space.

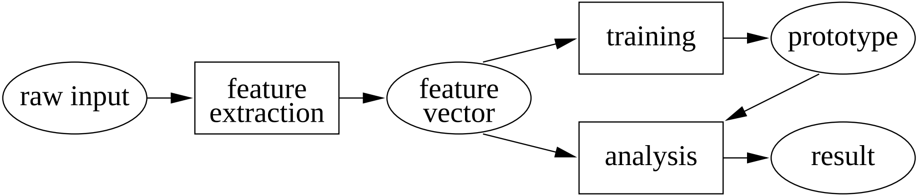
In vector space approach, an object is represented as a vector of feature values $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ in the real vector space.

- Each vector component corresponds to a dimension in the space.
- Each feature value corresponds to a coordinate in the space.
- The various dimensions are orthogonal to each other.
- In matrix notation, a vector is a column matrix:
  $\mathbf{v} = (v_1, v_2, \ldots, v_n)^T$.

Euclidean space is convenient for solving many problems:

- measuring difference or distance
- classification
- clustering
- regression

Typical Processing Stages



For more details on classification and clustering, take

- CS5242 Advanced Neural Networks
- EE5907 Pattern Recognition

Good references: [1, 2]

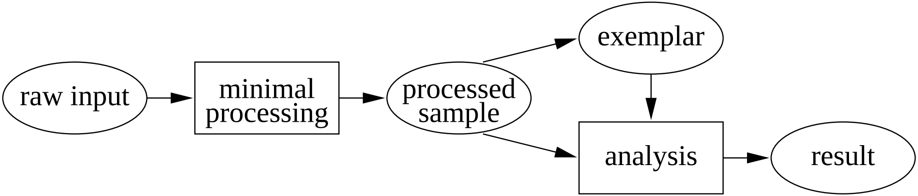
Exemplar-Based Approach

An exemplar is a representative example.

- Exemplars may not be represented as vectors.
  They can be represented as complex data structures.
- The space in which they reside may not be a vector space.
- The difference measure may not be Euclidean distance.

Example 1: Face images under different light conditions [4].

Typical Processing Stages



- The analysis method is usually direct measurement of similarity or difference between input and all exemplars.
- So, it is important to devise an efficient and accurate similarity measure.
- For more details, read relevant papers, e.g., [4].

Model-Based Approach

Why use models?

- Models can represent complex objects and knowledge about them.
- Models can help to reduce search space.

Example: Object recognition: recognize the car in the image.

(a)

(b)

(a) Image of cars in complex background.

(b) Extracted edges. Which edge belongs to which car?

- How to collect the edges that belong to the same car?
- This problem is impossible to solve without a model.

Main Types of Models I

1. Rigid Models:
   - Usually for 2D or 3D physical models, e.g., tables, chairs.
   - The models are rigid. They can be transformed only by linear transformations: scaling, translation, rotation.
2. Articulated Models:
   - Usually for 2D or 3D physical models with joints, e.g., human body.
   - The model parts are rigid and rotate about the joints.
3. Deformable Models:
   - Usually for 2D or 3D models, e.g., snake, balloon.
   - The models can be deformed into various shapes.
4. Statistical Models:
   - Can have any number of dimensions.
   - Usually adopt the linear vector space.
   - In some cases, the models are deformable.

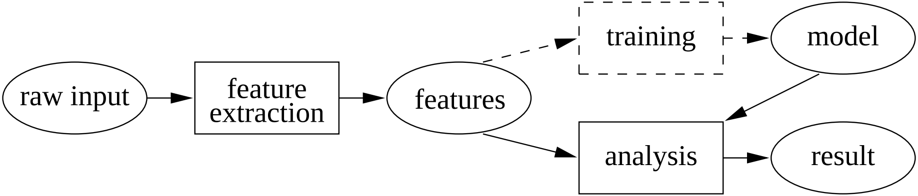
Main Types of Models II

5. Discrete Models:
    - Discrete representations, e.g., set, list, tree, graph, etc.
    - Can represent very simple and very complex objects.

Notes:

- There are overlaps between various model types.
- Can combine different model types to solve a problem.

Typical Processing Stages



- Models may be learned or hand-crafted.

Representation of Models

Parametric Model

- Represents a model by one or more parametric equations.
- Very concise representation.
- Allows for mathematical analysis.
- Suitable for representing objects with regular shapes.
- Examples:
  - Gaussian: mean and covariance matrix of input samples.
  - Gaussian mixture: weighted sum of Gaussians.
  - Polynomial functions: represent curves, surfaces, etc.

Non-parametric Model

- Represents a model by an aggregate of items, e.g., features, 2D/3D points.
- The representation can be simple (e.g., set, list) or complex (e.g., tree, graph).
- The representation requires more storage space.
- Suitable for representing objects with free-form shapes and abstract objects (e.g., text documents).

Hybrid Model

- Combines parametric and non-parametric models.

Reference

- R. O. Duda, P. E. Hart, and D. G. Stork.
  Pattern Classification.
  John Wiley, 2nd edition, 2001.
- S. Haykin.
  Neural Networks: A Comprehensive Foundation.
  Prentice-Hall, 1999.
- J. E. Marsden and M. J. Hoffman.
  Elementary Classical Analysis.
  W. H. Freeman, 2nd edition, 1993.
- T. Sim and T. Kanade.
  Combining models and exemplars for face recognition.
  In *Proc. CVPR Workshop on Models versus Exemplars in Computer Vision*, 2001.