T-61.6030: Multimedia Retrieval

Ch.2: Languages for Metadata
Ch.3: Pattern Recognition for Multimedia

Mats Sjöberg
What is metadata?

- literally, “data about data”
- descriptive information about data sources
- aids in organisation, identification, representation, localisation, interoperability, management and use of data
Why do we need metadata?

- multimedia objects are typically large in size, expensive to transport and process
- e.g. images, video, hard to summarise in textual, structured form
Why do we need metadata?

- description and identification
- querying, e.g. by author, genre, ...
- administration
- preservation, to facilitate archival and refreshing
- technical, e.g. data formats
Classification of metadata

• *content-independent*, e.g. date of creation and location

vs

• *content-dependent*, e.g. manual annotation, extracted features
Classification of metadata

• *domain-independent*, e.g. colour histogram of image

vs

• *domain-dependent*, e.g. land cover in GIS (geographic information system)
Semantics of metadata and metadata languages

• shared understanding of the meaning of metadata

• solution: metadata language with standardised semantics

• computer technical layer, e.g. Unicode text

• conventional layer, e.g. “author” is the original author of a book, not the publisher
Dublin Core (DC)

• workshop in 1995 in Dublin, Ohio (USA)
• continued development by Dublin Core Metadata Initiative (DCMI)
• specifies a framework for descriptive metadata
• widely used in e.g. libraries, universities, museums
DC: 15 core elements

- Contributor
- Coverage
- Creator
- Description
- Date
- Format
- Identifier
- Language
- Publisher
- Relation
- Right
- Source
- Subject
- Title
- Type
DC: web page example

Identifier = "http://dublincore.org/

Title = "Dublin Core Metadata Initiative -- Home Page"

Description = "The Dublin Core Metadata Initiative Web site"

Date = "2006-12-18"

Format = "text/html"

Language = "en"

Creator = "The Dublin Core Metadata Initiative (DCMI)"

Contributor = "The Dublin Core Usage Board"

Type = "InteractiveResource"
DC: some problems

• Not very precise, values fields are just text strings
• At best a recommended practice may exist
• Good for human consumption and full text search
• Bad for interoperability
DC: extensions

• extending using Dublin Core as the baseline: e.g. MPEG-7 and IEEE-LOM

• Dublin Core qualifiers are more explicit about attributes, e.g. “date.created”

• more precise value fields: e.g. “Person” data structure for “author”
Resource Description Framework (RDF)

• family of W3C specifications
• initially metadata for web resources in XML
• now more general: metadata about “resources” having “properties”
• RDF language, concrete RDFS schema
• e.g. Dublin Core as RDFS schema
RDF statement

- directed labelled graph
RDF example
RDF example

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:exterms="http://www.example.org/terms/">
  <rdf:Description
    rdf:about="http://www.example.org/index.html">
    <exterms:creation-date
      rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
      1999-08-16
    </exterms:creation-date>
    <dc:language>en</dc:language>
    <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>
  </rdf:Description>
</rdf:RDF>
RDF Schema

• an extension of RDF
• application specific classes and properties
• e.g. for Dublin Core:
  - “editor” is a property, domain:“Journal”, range:“Person”
  - “editor” is a sub-property of “contributor”
MPEG-7

• ISO/IEC Moving Pictures Experts Groups
MPEG-7 is a large and complex standard

• structure of media objects and relationships between components

• image, music, audio, video, 3D models, speech analysis and segmentation

• usually only specialised parts implemented
MPEG-7 elements

Main elements are:

• Description Definition Language (DDL)
• Descriptors (Ds)
• Description Schemes (DSs)
• Binary format (BiM)
• System Tools
MPEG-7 elements

- Descriptors:
  - Syntax & semantic of feature representation

- Description Schemes

- Structuring

- Definition

- Tags
  - <scene id=1>
  - <time> ....
  - <camera>..
  - <annotation
  - </scene>

- Instantiation

- Encoding & Delivery

- Description Definition Language

- Extension
Chapter 3: Pattern Recognition for Multimedia Content Analysis
Recognising patterns in multimedia content

Example 1: semi-automatic annotation

• high-quality metadata without manual work
• pattern classification and concept hierarchies
• e.g. LSCOM: Activities (e.g. walking), Scene (e.g. indoor, outdoor), People (e.g. soldier, pope)...
• need to design a classifier for each concept
Recognising patterns in multimedia content

Example 2: automatic interpretation of multimedia streams

- e.g. monitoring customers in shops, smart home applications, airport security
- typically *modelling* with finite state descriptions
- also *unsupervised learning*
Recognising patterns in multimedia content

We will look at four specific tasks:

• pattern classification
• modelling
• unsupervised learning, clustering
• dimensionality reduction
Pattern classification

• we have labelled example patterns
• task: generalise class structure, i.e. decide class of new input patterns
• feature representation: each pattern a point in a feature space
Pattern classification: some issues

- residual uncertainty
- limited availability of data, optimal prediction function: Bayes classifier:
  \[ f(x) = \arg\max_{k \in \mathcal{K}} p(k | X = x) \]
- need for prior assumptions
- noise and error
- under- and overfitting
- irrelevant features
Pattern classification

- Bayes classifier: chose class $k$ with highest probability given feature vector $x$
- natural approach: compare with $n$-nearest neighbours
- other approach: discriminant methods:
  - estimate feature value densities given a certain class, from this estimate $p(k|X = x)$
  - e.g. multivariate Gaussian density
Pattern classification: support vector machines

- training examples are mapped into a high-dimensional space
- seek support vector classifier, optimal separating hyper-plane
- maximise distance of samples to decision boundary, i.e. margin-optimisation
Fig. 3.2. Separating hyperplane.
Pattern classification: support vector machines

• hyper-plane defined as: \( \langle w, x \rangle + b = 0 \)

• choose \( w \) such that smallest margin: \( 1/\|w\| \)

margin hyper-planes: \( \langle w, x \rangle + b = \pm 1 \)

• constrained minimisation problem:

\[
\min_{w \in \mathcal{H}, b \in \mathbb{R}} \frac{1}{2} \|w\|^2 \\
\text{subject to: } y_i(\langle w, x_i \rangle + b) \geq 1, i = 1, \ldots, n
\]
Pattern classification: support vector machines

- constrained optimisation solved via Lagrangian function, and dual problem
- “slack” variables introduced for robustness
- classifier:
  \[ f(x) = \text{sign} \left( \sum_{i=1}^{n} \alpha_i y_i \langle x, x_i \rangle + b \right) \]
- replace inner product with kernel,

**kernel trick:** \[ k(x, x') = \langle \Phi(x), \Phi(x') \rangle \]
Modelling

• Hidden Markov Models very popular, especially in speech recognition

• states *hidden*, satisfy the Markov property

source: wikipedia
Modelling

• Example: we do not know the weather state $q_i$

• but can observe if people use umbrellas $o_i$

• Bayes’ rule: $P(q_i | o_i) = \frac{P(o_i | q_i) P(q_i)}{P(o_i)}$

• Generalised: $P(Q | O) \propto \prod_{i=1}^{n} P(o_i | q_i) \prod_{i=1}^{n} (q_i | q_{i-1})$
Unsupervised learning and clustering

• “unsupervised classification of patterns into groups” (Jain et al. 1999)

• uses similarity or proximity measures

• seek sparse and dense regions in a data set

• hierarchical methods or iterative optimisation
Hierarchical clustering

- keeps merging closest pair of clusters/patterns until a threshold
- cluster distance: *single link* or *complete link*
- also e.g. centroid, subset of points (CURE)
Iterative optimisation

- optimises a quality criterion
- usually a single partitioning
- simple example: \( k \)-means minimises summed squared errors
- not feasible for large multimedia databases
- solutions: divide and conquer, incremental approach, parallel processing
Dimension reduction

• reduce the number of features for pattern representation
• irrelevant features bad for classification
• reduces computational and memory costs
• ugly duckling theorem (Watanabe 1969)
• feature extraction and feature selection
Feature extraction

• create new features by combination and transformation of original features

• e.g. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA)

• PCA: retains dimensions that contribute most to variance of data

• LDA: seeks directions that best separate known classes in the data
Feature selection

• select a more *relevant* subset of features

• *filter methods*: rank variables according to some scoring function, e.g. *mutual information* between each variable and class

• *wrapper methods*: try different combinations, evaluate e.g. cross-validation

• *embedded methods*: selection embedded within learning process
Summary

• many more methods in machine learning and pattern recognition

• some main issues:
  – limited and noisy data
  – over- and underfitting
  – measuring classifier performance

• quality of pattern recognition
  ⇒ quality of automated multimedia analysis