Multimedia Retrieval
2016-1017
Distance

Egon L. van den Broek
The project: Topics

- Painting similarity (or distance)
- Music distance
- Mood distance

Via the project you’ve experienced that defining distance is both complex and crucial for MMR.
The project: Two perspectives

• Man
• Machine

or?

• Objective
• Subjective
The default ...

• Default: distance = Euclidean distance

• This is how it is treated at high school.
• This can be; but, does not have to be.
Aims

• Learn what a metric is
• Be able to determine whether or not a claimed metric is a true metric
• Understand human’s distance heuristics
• Get introduced to some common MMR metrics
• Learn what semantic distance is
Accompanying literature

- The book’s chapters 8, 25, and 28
- The book’s Appendix B
One for all ...

“... whatever the media is, the same categorization techniques can be used. ... Hence, we do not differentiate categorization methods by their input data. This statement is not necessarily true for descriptions.” (p. 140)
Terminology

• Matching: given e.g. two vectors $A$ and $B$, determine their similarity, the extent to which they match
  ⇒ dissimilarity measures

• Retrieval: given a query object and a database of models, find the most similar ones
  ⇒ indexing: build a data structure to speed up the search
Matching Feature Vectors

Result of feature extraction: numerical values $x_1, \ldots, x_n$ assembled in a feature vector $x=(x_1,\ldots,x_n)$
e.g., 64 values for hue histogram, 8 for edge directions histogram, 16 for wavelet coefficients, 16 for Fourier coefficients of contour $\Rightarrow n=104$
In general: $n$-dimensional feature space
MMR as closed-loop system
Signal processing (pattern recognition)
Matching

Given two MM documents, or two derived feature vectors, how to compute the (dis)similarity?
Rule-based matching

if $f < \varepsilon$ then
    follow left branch
else
    follow right branch
endif

Note. Check what the book says about the random forest algorithm!
Distance-based matching/categorization

What algorithm to choose depends on
• what similarity measure, depends on
• what required properties, depends on
• what particular matching problem, depends on
• what application
What application?

- retrieval
- recognition and classification
- alignment, registration
- approximation
What problem?

• computation problem: $d(A,B)$
• decision problem: $d(A,B) \leq \varepsilon$?
• optimization problem: find $g$: $\min d(g(A),B)$
What properties?

• Metric properties
• Continuity
• Invariance
Define a metric

• Can we define a metric, a distance function?

• Can we give a formal definition?
Metric Properties

A metric on a set $S$ is a function: $d: S \times S \rightarrow [0,\infty)$, where $[0,\infty)$ is the set of non-negative real numbers and the following conditions (i.e., axioms) are satisfied:

1. **self-identity** $d(x,x) = d(y,y)$ or $d(x,y) = 0$ iff $x = y$
   
   • a.k.a.: coincidence axiom and the identity of indiscernibles

2. **positivity** $d(x,y) \geq 0$
   
   • a.k.a.: non-negativity or separation axiom (*included later*)

3. **symmetry** $d(x,y) = d(y,x)$

4. **triangle inequality** $d(x,z) \leq d(x,y) + d(y,z)$
   
   • a.k.a. subadditivity
... metrics (properties of)

• There are also semi-metrics, pseudo-metrics, quasimetrics, metametrics, semimetrics, premetrics, and pseudoquasimetrics.

• ultrametric: when 4 is replaced by: \( d(x,z) \leq \max(d(x,y),d(y,z)) \)

• translation invariant metric: \( d(x,y) = d(x+a,y+a) \)
Symmetry
Symmetry

• Movies
• Biometrics
• graphics
Symmetry

\[ d(A,B) = d(B,A) \]

not always so for human perception

variant A:                                prototype B:

\[ d(A,B) < d(B,A) \]
Triangle inequality

\[ d(A, B) + d(B, C) \geq d(A, C) \]

not always so for human perception, in particular for partial matching.
Partial similarity (1)

Fig. 2 Illustration of partial similarity problems of different classes of objects (left to right, top to bottom): non-rigid three-dimensional shapes, two-dimensional articulated shapes, images and text sequences

PARTIAL SIMILARITY
PARETO OPTIMUM
Partial similarity (2)
Continuity

Robustness
arbitrary small changes:
• deformation
• blurring
• cracks
• noise
lead to arbitrary small changes in similarity
Similarity from complex algorithms

Shape deformation similarity is non-metric: similarity can be assessed by minimizing the energy of deformation $E$ spent while maximizing matching $M$ between edges.
Invariance

\[ d(g(A), g(B)) = d(A, B) \]
or \[ d(g(A), B) = d(A, B) \]
for all \( g \) in transformation group \( G \)

Argyropelecus olfersi                         Sternoptyx dialphana

(D'Arcy Thompson, 1911)
Why non-metric distances?

• Mimicking human similarity
  – Human similarity judgments are not metric
  – Subset matching, ignoring the most dissimilar parts

• Robustness
  – Distance measures robust to outliers or very noisy data typically violate triangulation.
$L_p$ metrics

- Also called Minkowski distance
- $x=(x_1,...,x_n), y=(y_1,...,y_n)$

\[
L_p = \left( \sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}
\]

- Metric for $p \geq 1$
$L_1$ and $L_2$ metrics

- $L_1$: taxicab, city block, Manhattan, rectilinear distance

- $L_2$: Euclidean distance
$L_1$ and $L_2$ metrics: pros and cons

• The $L_1$ norm and the $L_2$ norm are mostly used because of their low computational cost.

• many false negatives because neighboring bins are not considered
$L_\infty$ metric

$L_\infty$: max, chessboard, Chebyshev, or Tchebychev distance, $\forall_i \max\{|x_i - y_i|\}$
$L_p$ metrics: relations

$p = 2^{-2} = 0.25$

$p = 2^{-1.5} = 0.354$

$p = 2^{-1} = 0.5$

$p = 2^{-0.5} = 0.707$

$p = 2^0 = 1$

$p = 2^1 = 2$

$p = 2^{1.5} = 2.828$

$p = 2^2 = 4$

$p = 2^\infty = \infty$
Histogram Matching

- Histogram seen as feature vector
  e.g. $d(H_1, H_2)$ is Euclidean distance
Hamming distance

Binary vectors/histograms

\[ d(H_1, H_2) = \sum_{i=1}^{n} |H_1[i] - H_2[i]| = \sum_{i=1}^{n} (H_1 \ XOR \ H_2) \]
Weighted mean variance

• Includes minimal information about the data distribution.
• Several implementation possible.
• Illustration by one of my own (Van den Broek et al., 2005). 😊
Intersection metric

A distance measure calculates the distance between two histograms. A distance of zero represents a perfect match. We use the histogram intersection distance \( D \) of Swain and Ballard\(^{25} \) between a query image \( q \) and a target image \( t \):

\[
D_{q,t} = \sum_{m=0}^{M-1} |h_q(m) - h_t(m)|,
\]

(1)

where \( M \) is the total number of bins, \( h_q \) is the normalized query histogram, and \( h_t \) is the normalized target histogram.
Extended Intersection metric (1)

These values are stored in a color bucket $b$, assigned to every color category (or quantized color space segment):

\[
\begin{align*}
    x_1(b) &= \#(b), \quad \text{i.e., the number of pixels in bucket } b; \text{ the original histogram value } h \\
    x_2(b) &= \alpha H(b), \quad \text{i.e., the mean hue } H \text{ of bucket } b \\
    x_3(b) &= \alpha S(b), \quad \text{i.e., the mean saturation } S \text{ of bucket } b \\
    x_4(b) &= \sigma H(b), \quad \text{i.e., the standard deviation of the hue values } H \text{ in bucket } b \\
    x_5(b) &= \sigma S(b), \quad \text{i.e., the standard deviation of the saturation values } S \text{ in bucket } b \\
    x_6(b) &= \sigma I(b), \quad \text{i.e., the standard deviation of the intensity values } I \text{ in bucket } b,
\end{align*}
\]

where $x_i(b)$ denotes value $i$ of color bucket $b$ of either query image $q$: $q_i(b)$ or of target image $t$: $t_i(b)$.
Edit Distance

- Feature vector interpreted as string of characters
- Edit distance operations
  - Insertion, where an extra character is inserted into the string
  - Deletion, where a character has been removed from the string
  - Transposition, in which two characters are reversed in their sequence
  - Substitution, which is an insertion followed by a deletion
Edit Distance

• Strings with a small edit distance are likely to be similar
• Edit distance is number of edit distance operations from one string to another
• Example: chaincodes
  12345678
  123845677 have distance 3 (2?)
Edit Distance

• Defined as: $d(x, y) = |x| + |y| - 2\text{LCS}(x,y)$, with LCS being the Longest Common Subsequence.

• Use dynamic programming to determine the edit distance.

• It is a metric when
  – every edit operation has positive cost and
  – for every operation, there is an inverse operation with equal cost.
Cosine Distance (1)

- A judgment of orientation and not magnitude
- Derives from the definition of dot product between two vectors, $x=(x_1,\ldots,x_n)$, $y=(y_1,\ldots,y_n)$
- Only angle relevant, not vector lengths
Calculate $a \cdot b$
Calculate $a \cdot b$ (in $nD$)
Cosine Distance (2)

\[ \mathbf{a} \cdot \mathbf{b} = \| \mathbf{a} \| \| \mathbf{b} \| \cos \theta \]

\[
\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\| \mathbf{A} \| \| \mathbf{B} \|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}
\]

\[ D_C(A, B) = 1 - S_C(A, B) \]

\[ d(x, y) = 1 - \cos(\angle(x, y)) = 1 - \frac{x \cdot y}{\| x \| \| y \|} \]

• Used in positive space, outcome bounded in [0,1].
• Commonly used in high-dimensional positive spaces.
Cosine Distance (3)

• Is it a metric?
  \[ d(x, y) = 1 - \cos(\angle(x, y)) = 1 - \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}|| \cdot ||\mathbf{y}||} \]

• Answer: no! Why?

• It does not satisfy:
  – self-identity (coincidence axiom)
  – triangle inequality (or subadditivity)

• Is this bad? If yes or no, why?