KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

„Image Processing and Computer Graphics"

Prezentacja multimedialna współfinansowana przez Unię Europejską w ramach Europejskiego Funduszu Społecznego w projekcie pt.

„Innowacyjna dydaktyka bez ograniczeń - zintegrowany rozwój Politechniki Łódzkiej - zarządzanie Uczelnią, nowoczesna oferta edukacyjna i wzmacniania zdolności do zatrudniania osób niepełnosprawnych"

Politechnika Łódzka

Politechnika Łódzka, ul. Żeromskiego 116, 90-924 Łódź, tel. (042) 631 28 83
www.kapitalludzki.p.lodz.pl

Contribution

Lecturer:

Piotr M. Szczypiński, Dr inż.
Institute of Electronics, Technical University of Lodz
Wolczanska 211/215, 90-924 Lodz, Poland
tel. +48426312638
http://www.eletel.p.lodz.pl/pms
pms@p.lodz.pl

Fragments of the lecture were also provided by

Andrzej Materka, Michał Strzelecki, Paweł Strumiłło, Piotr Romaniuk, Piotr Makowski

and were used with permission.

Features and classification

- Example
- Problem(s) statement
- Textural features
- Multidimensional analysis
- Geometrical parameters
- Feature selection
- Classification techniques

Example

Goal: find blue rectangles

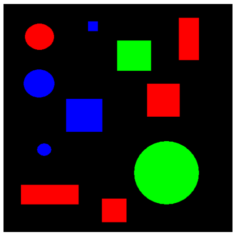
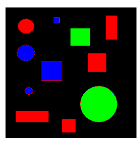
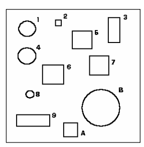


Image processing (from image to features)

1. Perform image enhancement.
2. Perform segmentation to find individual objects.
3. Compute numeric descriptors (features):
   a) Area
   b) Perimeter
   c) Roundness (area / perimeter^2)
   d) Average of blue component





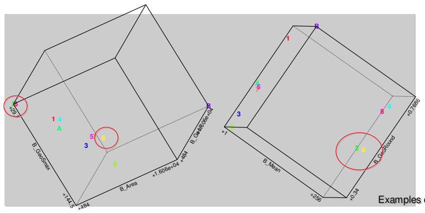
Report — File, Feature selection, Tools — mod2.par

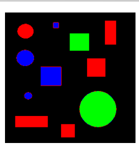
Feature name	1	2	3	4	5	6	7	8	9	A	B
B_GeoF	2975	484	4275	3391	5168	5994	5402	682	5676	2915	16058
B_GeoSmax	64.498	29.698	103.79	68.469	100.57	108.3	102.53	31.145	135.03	74.967	144.54
B_GeoRound	0.7151	0.5488	0.3968	0.7234	0.5107	0.54	0.5	0.7031	0.34	0.5248	0.7686
B_Area	2975	484	4275	3391	5168	5994	5402	682	5676	2915	16058
B_Mean	1	256	1	256	1	256	1	256	1	1	1

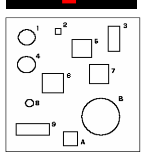
Examples created with MaZda 4.60

Pattern analysis (from features to decision)

3. Analyze feature vectors distribution in feature space.
4. Find such features for which:
   a) searched feature vectors group
   b) and are separate from other vectors



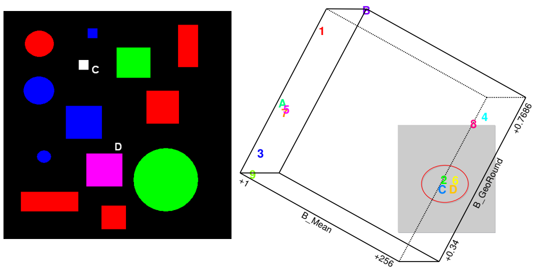




Examples created with MaZda 4.60

Pattern analysis (verification)

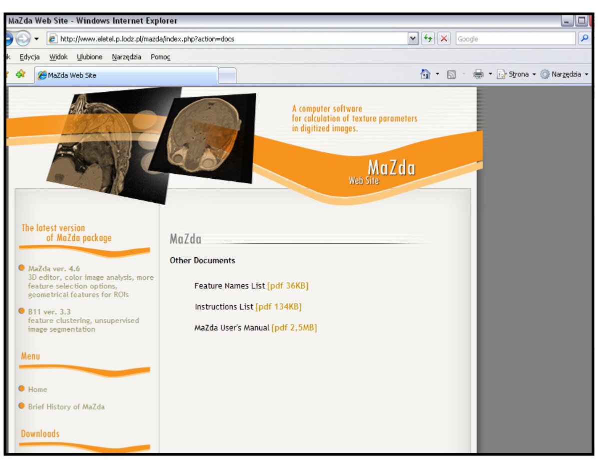
5. Verify with other images (data sets)



Examples created with MaZda 4.60

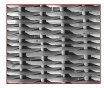
Problems

- Define a goal of image recognition, gather example images, establish methods for image processing
- Find numeric values, which can characterize and differentiate objects/regions/images
- Find a rule for classification of objects/regions/images based on feature space analysis
- Verify the method



Features

- Features
  - Shape based
    - Geometrical
    - Moment-based
    - Topological
  - Region based
    - Structural
    - Statistical
    - Transform
    - Model-based



Geometrical parameters - Area

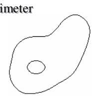
Area - number of the object pixels

Profile area - number of the profile pixels

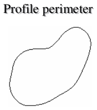
How to find profile?
Fill-in (flood fill algorithm) a background.
What lefts is a profile.

Geometrical parameters - Area

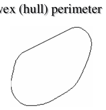
Perimeter



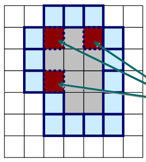
Profile perimeter



Convex (hull) perimeter



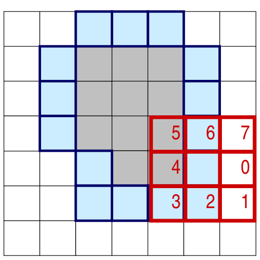
Geometrical parameters - Perimeter (pixels)



Contour pixels belong to the object and have at least one neighbor that does not belong to the object. (4 or 8 pixel neighborhood?)

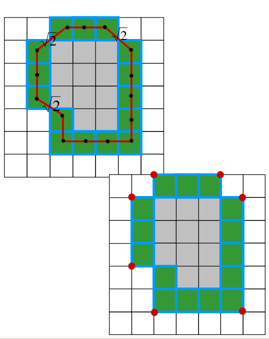
These would overestimate profile length for round objects

Geometrical parameters – Perimeter (boundary tracking)



1. Position a template over any contour pixel
2. Note coordinates of pixel on the tracking list
3. Check pixels in order given by the template
4. If checked pixel is a contour pixel shift a template over it and go to 2.
5. Finish when all the pixels are on the list

Geometrical parameters – Perimeter (length)



1. Contour length is a count of contour pixels
2. Contour length is the sum of line segments lengths connecting pixel centers
3. Contour length is estimated from:

Length = aN - bM

N - number of external sides of contour pixels
M - number of contour vertices

$$a = \frac{\pi(1+\sqrt{2})}{8} \qquad b = \frac{\pi}{8\sqrt{2}}$$

Roundness

Roundness (compactness) - is a measure of how region shape is different from a circular shape

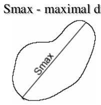
$$\text{roundness} = \frac{(\text{boundary length})^2}{4\pi(\text{area})}$$

For a circular boundary roundness is minimum and equals 1. For a square it is equal $4/\pi > 1$.

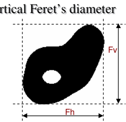
Geometrical parameters – Diameters

Spol - diameter of the area equivalent circle

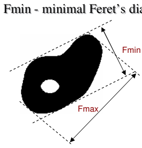
Smax - maximal diameter



Fh - horizontal Feret's diameter
Fv - vertical Feret's diameter



Fmax - maximal Feret's diameter
Fmin - minimal Feret's diameter



Geometrical parameters – Other Ratios

Malinowska ratio

$$Rm = \frac{\text{perimeter}}{2\sqrt{\pi\,\text{area}}} - 1$$

Blair-Bliss ratio
Danielsson ratio
Haralic ratio

TABLE I
FEATURES CALCULATED FOR A SINGLE, SEGMENTED CT SCAN. *FS1* REPRESENTS THE FIRST USED FEATURE SET, *FS2* REPRESENTS THE SECOND ONE.

Feature	*Definition*	*FS1*	*FS2*
Number of scan	n	yes	yes
Absolute circumference	L	no	yes
Absolute area	S	yes	yes
Relative circumference	$L_{rel} = \frac{L}{L_B}$	no	yes
Relative area	$S_{rel} = \frac{S}{S_B}$	yes	yes
Malinowska Coefficient	$R_M = \frac{L}{2\sqrt{\pi S}} - 1$	no	yes
Modified Malinowska Coeff.	$R_{mM} = \frac{2\sqrt{\pi S}}{L}$	no	yes
Blair-Bliss Coefficient	$R_B = \frac{S}{\sqrt{2\pi(\sum_i r_i)^2}}$	no	yes
Blair-Danielsson Coeff.	$R_D = \frac{S^3}{\left(\sum_i l_i\right)^2}$	no	yes
Haralick Coefficient	$R_H = \sqrt{\frac{(\sum_i d_i)^2}{n \sum_i d_i^2 - 1}}$	no	yes
Feret Coefficient	$R_F = \frac{L_h}{L_v}$	no	yes
Circularity Coefficient 1	$R_{C1} = 2\sqrt{\frac{S}{\pi}}$	no	yes
Circularity Coefficient 2	$R_{C2} = \frac{L}{\pi}$		
Center of mass in X	$R_{mx} = \frac{\sum_i x_i}{\sum_i 1}$		
Center of mass in Y	$R_{my} = \frac{\sum_i y_i}{\sum_i 1}$		

of neurons in the input layer is equal to
in a single feature vector. Number of
layer is equal to the number of outp
hidden layer is calculated using a sta
to the average size of input and ou
training epochs is equal to 500, learni
and momentum coefficient to 0.2.

Final decision process is the followi
is classified using the neural network.
and the dominant class in all classifica

2) Classification using automatic
proach: In general, an automatic ima
describes an input image (represente
vectors) with a subset of words from
Input of image annotation is a set of
dictionary. Output of automatic image
of dictionary. The goal of the method i
best possible way, according to prese

M. Paradowski, et. al.
On Automation of Brain CT Image Analysis
Computer Science and Information Technology, 2008. IMCSIT
http://www.proceedings2008.imcsit.org/pliks/200.pdf

Successful verification will lead to furtl

Shape factors

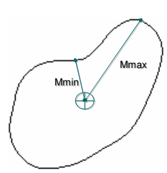
Table 2. List of selected shape factors [7, 8]

Factor	Formula	Remarks/comment
Feret Factor	$R_F = \frac{L_h}{L_V}$	It assumes a small value for elongated objects, is characterized by a large variability. It is easy to calculate, but susceptible to a change of scale
Malinowska Factor	$R_M = \frac{L}{2\sqrt{\pi S}} - 1$	It is characterized by a medium range of values, its values are larger for objects of elongated shape. It does not show a big influence on a scale changes and a figure turn
Circulation Factor (1) (in SigmaScan Pro it corresponds to a parameter of Feret Diameter)	$R_{C1} = \sqrt{\frac{4S}{\pi}}$	It calculates the diameter of a fictitious circular object that has the same area as the object being measured. It depends strongly on the object size; it does not differentiate shapes of the objects. **(accessible in SigmaScan Pro)**
Circulation Factor	$R_{C2} = \frac{L}{\pi}$	It determines a diameter of the circle with a perimeter equal to the perimeter of the object analyzed; it depends strongly on the object size; it does not differentiate shapes of the objects
Compactness Factor	$Compactness = \frac{L^2}{S}$	Its values are larger for objects of elongated shape (for a circle =4π). **(accessible in SigmaScan Pro)**
Shape Factors	$Shape\ Factor = \frac{4\pi S}{L^2}$	For a circle =1, for a line =0. **(accessible in SigmaScan Pro)**
Lp_1	$Lp_1 = \frac{r\,min}{R\,max}$	It reflects well circulation of the object; for a circle it is one
Lp_2	$Lp_2 = \frac{l_{max}}{L}$	For figures with a very "jerked" edge the factor will be of a low value; for regular figures it reflects well an elongation of the object
Lp_3	$Lp_3 = \frac{l_{max}}{l_{min}}$	For regular figures it reflects well an elongation of the object
Blair-Bliss Factor	$R_B = \frac{S}{\sqrt{2\pi \sum_i r_i^2}}$	It does not show changes by the change of scale and the turn of the object examined
Danielsson Factor	$R_D = \frac{S^3}{\left(\sum_i l_i\right)^2}$	It is characterized by a large range of variability; it is neither too resistant to a scale change nor the object turn; time for the factor calculation is tens times longer than in case of other factors
Haralick Factor	$R_H = \sqrt{\frac{\left(\sum_i d_i\right)^2}{n \sum_i d_i^2 - 1}}$	It is characterized by a very low variability range; it is not being deformed either through a shape change or an object turn

P. Matusiewicz, A. Czarski, H. Adrian,
Estimation of materials microstructure parameters using computer program SigmaScan Pro
Metallurgy and Foundry Engineering, 2007 - baztech.icm.edu.pl

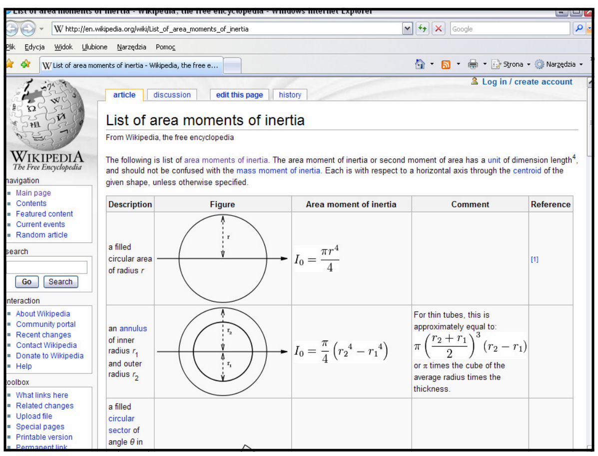
Moment-based parameters – Martin's radii

Mmin - Martin's minimal radius - minimum distance between gravity center and contour pixels
Mmax - Martin's maximal radius - maximum distance between gravity center and contour pixels
Maver - Martin's average radius - average distance between gravity center and contour pixels



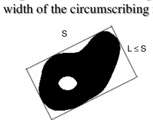
Moment-based parameters – Other

M2y - horizontal second order moment of inertia
M2y - vertical second order moment of inertia
M2xy - second order moment of inertia
Er - average distance from gravity center
Er2 - average square distance from gravity center
El - average distance from contour
El2 - average square distance from contour



Geometrical parameters – Other

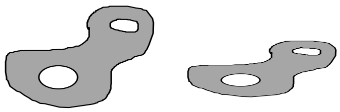
S - length of the circumscribing rectangle of minimal area
L - width of the circumscribing rectangle



SxL - area of circumscribing rectangle
D1 - diameter of profile inscribed circle of maximum area
D2 - diameter of circumscribing circle
Fd2 - area of circumscribing circle
LmaxE - length of the circumscribing ellipsis of minimal area
LminE - width of the circumscribing ellipsis of minimal area
FE - area of circumscribing ellipsis

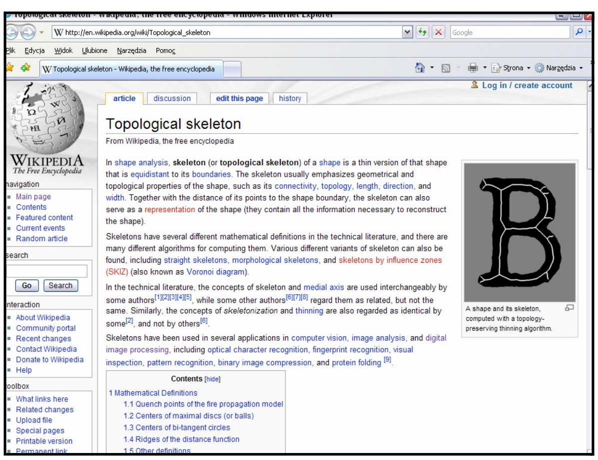
Topological features

Topologic shape features are invariant for *„rubber-sheet”* transformations of an elastic surface



Topological features (skeleton based)

No - number of skeletal loops (number of holes)
Nv - number of cavities
Ni - number of skeleton branches
Nx - skeletal junction number



http://en.wikipedia.org/wiki/Topological_skeleton

Topological skeleton

From Wikipedia, the free encyclopedia

In shape analysis, **skeleton** (or **topological skeleton**) of a shape is a thin version of that shape that is equidistant to its boundaries. The skeleton usually emphasizes geometrical and topological properties of the shape, such as its connectivity, topology, length, direction, and width. Together with the distance of its points to the shape boundary, the skeleton can also serve as a representation of the shape (they contain all the information necessary to reconstruct the shape).

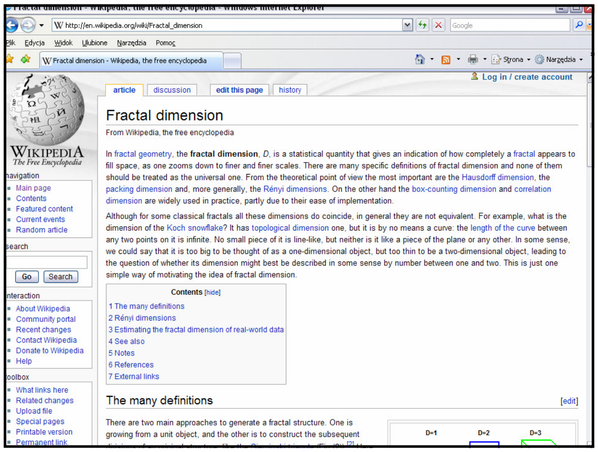
Skeletons have several different mathematical definitions in the technical literature, and there are many different algorithms for computing them. Various different variants of skeleton can also be found, including straight skeletons, morphological skeletons, and skeletons by influence zones (SKIZ) (also known as Voronoi diagram).

In the technical literature, the concepts of skeleton and medial axis are used interchangeably by some authors[1][2][3][4][5], while some other authors[6][7][8] regard them as related, but not the same. Similarly, the concepts of *skeletonization* and thinning are also regarded as identical by some[2], and not by others[6].

Skeletons have been used in several applications in computer vision, image analysis, and digital image processing, including optical character recognition, fingerprint recognition, visual inspection, pattern recognition, binary image compression, and protein folding [9].

A shape and its skeleton, computed with a topology-preserving thinning algorithm.

Contents [hide]


Fractal dimension - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/Fractal_dimension

Plik Edycja Widok Ulubione Narzędzia Pomoc

Fractal dimension - Wikipedia, the free encyclopedia

Strona Narzędzia

Log in / create account

article | discussion | edit this page | history

WIKIPEDIA
The Free Encyclopedia

navigation
- Main page
- Contents
- Featured content
- Current events
- Random article

search

Go | Search

interaction
- About Wikipedia
- Community portal
- Recent changes
- Contact Wikipedia
- Donate to Wikipedia
- Help

toolbox
- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link

Fractal dimension

From Wikipedia, the free encyclopedia

In fractal geometry, the **fractal dimension**, *D*, is a statistical quantity that gives an indication of how completely a fractal appears to fill space, as one zooms down to finer and finer scales. There are many specific definitions of fractal dimension and none of them should be treated as the universal one. From the theoretical point of view the most important are the Hausdorff dimension, the packing dimension and, more generally, the Rényi dimensions. On the other hand the box-counting dimension and correlation dimension are widely used in practice, partly due to their ease of implementation.

Although for some classical fractals all these dimensions do coincide, in general they are not equivalent. For example, what is the dimension of the Koch snowflake? It has topological dimension one, but it is by no means a curve: the length of the curve between any two points on it is infinite. No small piece of it is line-like, but neither is it like a piece of the plane or any other. In some sense, we could say that it is too big to be thought of as a one-dimensional object, but too thin to be a two-dimensional object, leading to the question of whether its dimension might best be described in some sense by number between one and two. This is just one simple way of motivating the idea of fractal dimension.

Contents [hide]

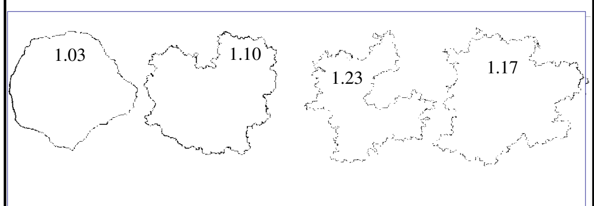
The many definitions [edit]

There are two main approaches to generate a fractal structure. One is growing from a unit object, and the other is to construct the subsequent

D=1 D=2 D=3

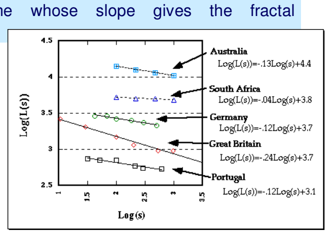
Fractal dimensions

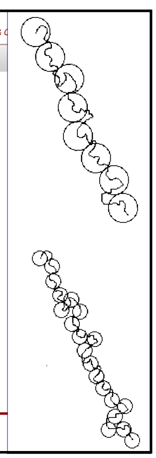
- The fractal dimension is the rate at which the length of an edge increases as the measurement scale is reduced. It may be understood as a measure of "roughness" of the edge or boundary.



Fractal Dimension

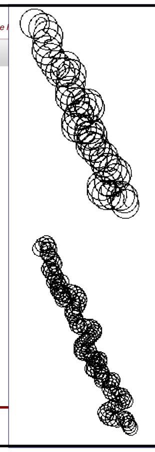
Richardson dimension -
is obtained by counting the number of strides needed to walk along the boundary, as a function of stride length; the result plotted on log-log axes, is a straight line whose slope gives the fractal dimension.





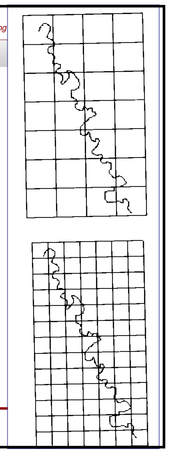
Fractal Dimension

Minkowski dimension -
in this method the circle is swept along a boundary (contour); plotting the area swept out by the circle (called sometimes the sausage) versus its radius on a log-log axes produces a line whose slope gives this fractal dimension.



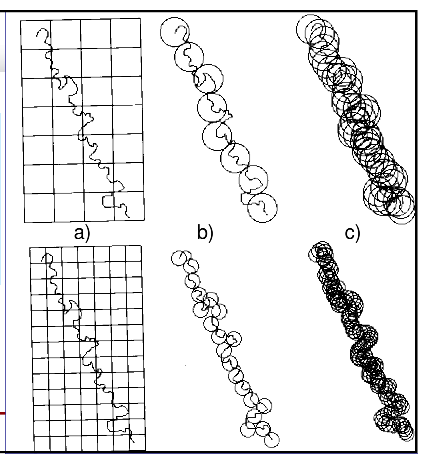
Fractal Dimension

Kolmogorov dimension -
a mesh of lines is drawn on the image and the number of grid squares through which the boundary passes is counted;
by plotting this number on log-log axes vs. the size of the grid and finding the slope of the line gives this fractal dimension.



Fractal Dimension

a) Kolmogorov,
b) Richardson,
c) Minkowski.



Region-based description

By characterising features of **region's interior** (regions texture).

A **texture** perceived by humans is a visualization of complex patterns composed of spatially organized, repeated subpatterns, which have a characteristic, somewhat uniform appearance. The local patterns within an image are perceived to demonstrate specific brightness, color size, roughness, directivity, randomness, smoothness, granulation, etc.

Although textures are easily perceived by humans, there is **no strict definition** what is a texture in image processing terms. Humans usually assess texture only qualitatively, while often **quantitative texture analysis is required**. To perform such quantification, **mathematically defined texture properties** have to be computed by texture analysis computer programs.

Approaches to texture analysis are usually categorised into
- statistical,
- structural,
- model-based and
- transform

http://dx.doi.org/10.1016/j.media.2008.12.002
http://www.eletel.p.lodz.pl/programy/cost/pdf_1.pdf

MaZda on-line User's Manual

Ukryj Wstecz Drukuj Opcje

1. The MaZda packa…
2. Getting started wi…
3. Theoretical backg…
3.1 Texture anal…
3.2 Feature redu…
3.3 Texture para…
4. B11 program for c…
5. 3D Data Analysis
6. References
7. Copyright

MaZda User's Manual

3.3. Texture parameters summary

MaZda allows computation of a variety of statistical parameters that are derived from image histogram, absolute gradient, run-length matrix co-occurrence matrix and autoregressive model (Table 3.3.1). One of the latest additions to **MaZda** are texture parameters derived from wavelet analysis.

Table 3.3.1. Texture parameters computed by **MaZda**

Histogram	Absolute gradient	Run-length matrix	Co-occurrence matrix	AR model
• mean • variance • skewness • kurtosis, • 1-% percentile • 10-% percentile • 50-% percentile • 90-% percentile • 99-% percentile	• mean • variance • skewness • kurtosis • percentage of pixels with nonzero gradient.	• run length nonuniformity • grey level nonuniformity • long run emphasis • short run emphasis • fraction of image in runs	• angular second moment • contrast • correlation • sum of squares • inverse difference moment • sum average • sum variance • sum entropy • entropy • difference variance • difference entropy	• θ_1 • θ_2 • θ_3 • θ_4 • σ

The run-length matrix-based parameters are computed 4 times for each ROI (for vertical, horizontal, 45-degree and 135-degree directions). The co-occurrence matrix-based parameters are [computed up to 20 times, for $(d,0)$, $(0,d)$, (d,d), $(d,-d)$ where the distance d can take values of 1, 2, 3, 4, and 5].

Definition and discussion of the above parameters can be found in (Haralick 1973, Haralick 1979, Hu 1994, Lerski 1993). A quick texture parameter reference is provided below for the user convenience.

MaZda on-line User's Manual

Ukryj Wstecz Drukuj Opcje

statistical

1. The MaZda packa…
2. Getting started wi…
3. Theoretical backg…
3.1 Texture anal…
3.2 Feature redu…
3.3 Texture para…
4. B11 program for c…
5. 3D Data Analysis
6. References
7. Copyright

3.3.1 Histogram-based features

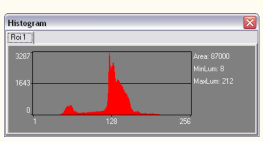
In the formulas that follow, $p(i)$ is a normalized histogram vector (i.e. histogram whose entries are divided by the total number of pixels in ROI), $i=1,2,\ldots,N_g$, and N_g denotes the number of intensity levels.

Mean: $\mu = \sum_{i=1}^{N_g} i p(i)$

Variance: $\sigma^2 = \sum_{i=1}^{N_g} (i-\mu)^2 p(i)$

Skewness: $\mu_3 = \sigma^{-3} \sum_{i=1}^{N_g} (i-\mu)^3 p(i)$

Kurtosis: $\mu_4 = \sigma^{-4} \sum_{i=1}^{N_g} (i-\mu)^4 p(i) - 3$

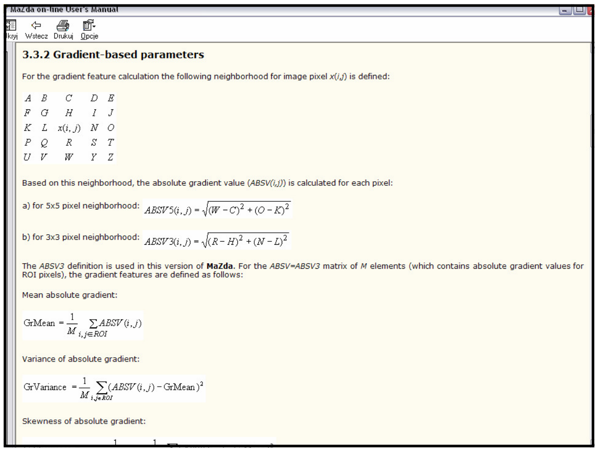


3.3.2 Gradient-based parameters

For the gradient feature calculation the following neighborhood for image pixel $x(i,j)$ is defined:

A	B	C	D	E
F	G	H	I	J
K	L	$x(i,j)$	N	O
P	Q	R	S	T
U	V	W	Y	Z

Based on this neighborhood, the absolute gradient value $(ABSV(i,j))$ is calculated for each pixel:



MaZda on-line User's Manual

Ukryj Wstecz Drukuj Opcje

3.3.2 Gradient-based parameters

For the gradient feature calculation the following neighborhood for image pixel $x(i,j)$ is defined:

A	B	C	D	E
F	G	H	I	J
K	L	x(i, j)	N	O
P	Q	R	S	T
U	V	W	Y	Z

Based on this neighborhood, the absolute gradient value ($ABSV(i,j)$) is calculated for each pixel:

a) for 5x5 pixel neighborhood: $ABSV5(i,j) = \sqrt{(W-C)^2 + (O-K)^2}$

b) for 3x3 pixel neighborhood: $ABSV3(i,j) = \sqrt{(R-H)^2 + (N-L)^2}$

The *ABSV3* definition is used in this version of **MaZda**. For the *ABSV=ABSV3* matrix of *M* elements (which contains absolute gradient values for ROI pixels), the gradient features are defined as follows:

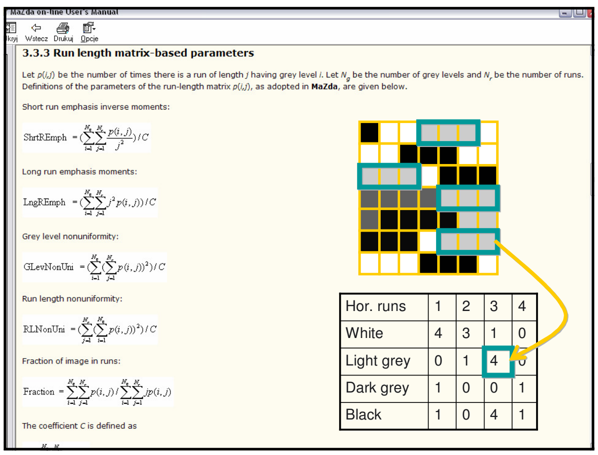
Mean absolute gradient:

$$\mathrm{GrMean} = \frac{1}{M} \sum_{i,j \in ROI} ABSV(i,j)$$

Variance of absolute gradient:

$$\mathrm{GrVariance} = \frac{1}{M} \sum_{i,j \in ROI} (ABSV(i,j) - \mathrm{GrMean})^2$$

Skewness of absolute gradient:



MaZda on-line User's Manual

Ukryj Wstecz Drukuj Opcje

3.3.3 Run length matrix-based parameters

Let $p(i,j)$ be the number of times there is a run of length j having grey level i. Let N_g be the number of grey levels and N_r be the number of runs. Definitions of the parameters of the run-length matrix $p(i,j)$, as adopted in **MaZda**, are given below.

Short run emphasis inverse moments:

$$\mathrm{ShrtREmph} = \left(\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} \frac{p(i,j)}{j^2}\right) / C$$

Long run emphasis moments:

$$\mathrm{LngREmph} = \left(\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} j^2 p(i,j)\right) / C$$

Grey level nonuniformity:

$$\mathrm{GLevNonUni} = \left(\sum_{i=1}^{N_g} \left(\sum_{j=1}^{N_r} p(i,j)\right)^2\right) / C$$

Run length nonuniformity:

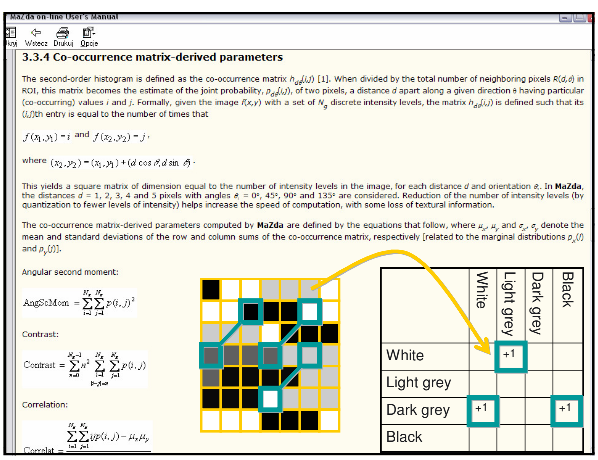
$$\mathrm{RLNonUni} = \left(\sum_{j=1}^{N_r} \left(\sum_{i=1}^{N_g} p(i,j)\right)^2\right) / C$$

Fraction of image in runs:

$$\mathrm{Fraction} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_r} p(i,j) \Big/ \sum_{i=1}^{N_g} \sum_{j=1}^{N_r} j p(i,j)$$

The coefficient *C* is defined as

Hor. runs	1	2	3	4
White	4	3	1	0
Light grey	0	1	4	0
Dark grey	1	0	0	1
Black	1	0	4	1



MaZda on-line User's Manual

Ukryj Wstecz Drukuj Opcje

3.3.4 Co-occurrence matrix-derived parameters

The second-order histogram is defined as the co-occurrence matrix $h_{d\theta}(i,j)$ [1]. When divided by the total number of neighboring pixels $R(d,\theta)$ in ROI, this matrix becomes the estimate of the joint probability, $p_{d\theta}(i,j)$, of two pixels, a distance d apart along a given direction θ having particular (co-occurring) values i and j. Formally, given the image $f(x,y)$ with a set of N_g discrete intensity levels, the matrix $h_{d\theta}(i,j)$ is defined such that its (i,j)th entry is equal to the number of times that

$f(x_1, y_1) = i$ and $f(x_2, y_2) = j$,

where $(x_2, y_2) = (x_1, y_1) + (d \cos\theta, d \sin\theta)$.

This yields a square matrix of dimension equal to the number of intensity levels in the image, for each distance d and orientation θ. In **MaZda**, the distances d = 1, 2, 3, 4 and 5 pixels with angles θ = 0°, 45°, 90° and 135° are considered. Reduction of the number of intensity levels (by quantization to fewer levels of intensity) helps increase the speed of computation, with some loss of textural information.

The co-occurrence matrix-derived parameters computed by **MaZda** are defined by the equations that follow, where μ_x, μ_y and σ_x, σ_y denote the mean and standard deviations of the row and column sums of the co-occurrence matrix, respectively [related to the marginal distributions $p_x(i)$ and $p_y(j)$].

Angular second moment:

$$\mathrm{AngScMom} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j)^2$$

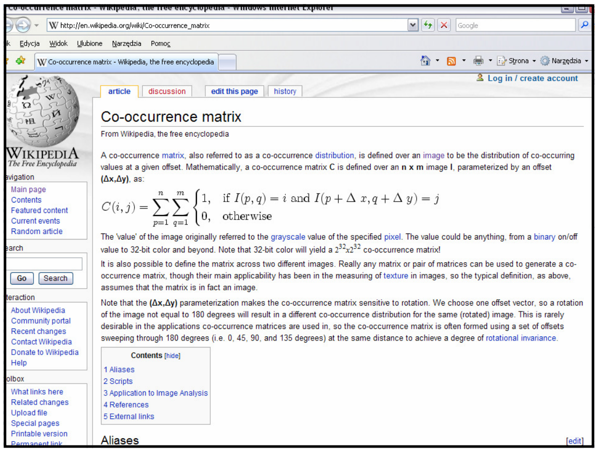
Contrast:

$$\mathrm{Contrast} = \sum_{n=0}^{N_g-1} n^2 \sum_{i=1}^{N_g} \sum_{\substack{j=1 \\ |i-j|=n}}^{N_g} p(i,j)$$

Correlation:

$$\mathrm{Correlat} = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} ijp(i,j) - \mu_x \mu_y}{\ldots}$$

	White	Light grey	Dark grey	Black
White		+1		
Light grey				
Dark grey	+1			+1
Black				



Co-occurrence matrix - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/Co-occurrence_matrix

Co-occurrence matrix

From Wikipedia, the free encyclopedia

A co-occurrence matrix, also referred to as a co-occurrence distribution, is defined over an image to be the distribution of co-occurring values at a given offset. Mathematically, a co-occurrence matrix **C** is defined over an **n x m** image **I**, parameterized by an offset **(Δx,Δy)**, as:

$$C(i,j)=\sum_{p=1}^{n}\sum_{q=1}^{m}\begin{cases}1, & \text{if } I(p,q)=i \text{ and } I(p+\Delta x,q+\Delta y)=j\\ 0, & \text{otherwise}\end{cases}$$

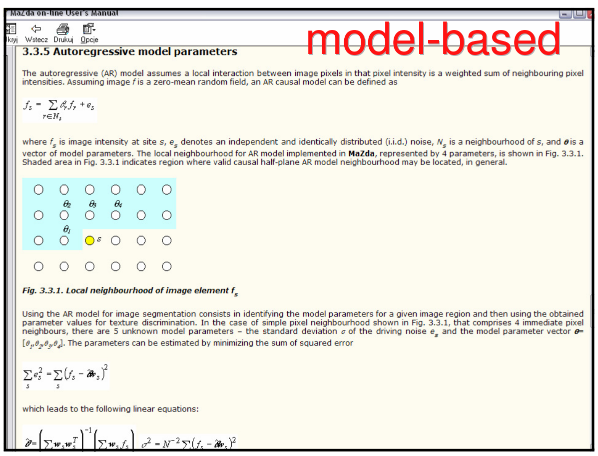
The 'value' of the image originally referred to the grayscale value of the specified pixel. The value could be anything, from a binary on/off value to 32-bit color and beyond. Note that 32-bit color will yield a $2^{32}x2^{32}$ co-occurrence matrix!

It is also possible to define the matrix across two different images. Really any matrix or pair of matrices can be used to generate a co-occurrence matrix, though their main applicability has been in the measuring of texture in images, so the typical definition, as above, assumes that the matrix is in fact an image.

Note that the **(Δx,Δy)** parameterization makes the co-occurrence matrix sensitive to rotation. We choose one offset vector, so a rotation of the image not equal to 180 degrees will result in a different co-occurrence distribution for the same (rotated) image. This is rarely desirable in the applications co-occurrence matrices are used in, so the co-occurrence matrix is often formed using a set of offsets sweeping through 180 degrees (i.e. 0, 45, 90, and 135 degrees) at the same distance to achieve a degree of rotational invariance.

Contents [hide]

Aliases



MaZda on-line User's Manual

model-based

3.3.5 Autoregressive model parameters

The autoregressive (AR) model assumes a local interaction between image pixels in that pixel intensity is a weighted sum of neighbouring pixel intensities. Assuming image *f* is a zero-mean random field, an AR causal model can be defined as

$$f_s=\sum_{r\in N_s}\theta_r f_r+e_s$$

where f_s is image intensity at site *s*, e_s denotes an independent and identically distributed (i.i.d.) noise, N_s is a neighbourhood of *s*, and $\boldsymbol{\theta}$ is a vector of model parameters. The local neighbourhood for AR model implemented in **MaZda**, represented by 4 parameters, is shown in Fig. 3.3.1. Shaded area in Fig. 3.3.1 indicates region where valid causal half-plane AR model neighbourhood may be located, in general.

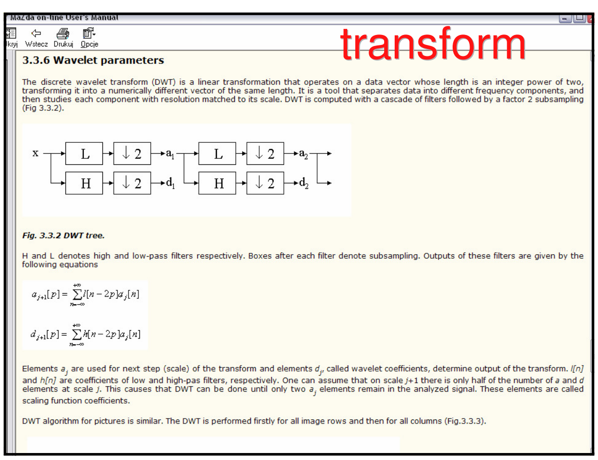
Fig. 3.3.1. Local neighbourhood of image element f_s

Using the AR model for image segmentation consists in identifying the model parameters for a given image region and then using the obtained parameter values for texture discrimination. In the case of simple pixel neighbourhood shown in Fig. 3.3.1, that comprises 4 immediate pixel neighbours, there are 5 unknown model parameters – the standard deviation σ of the driving noise e_s and the model parameter vector $\boldsymbol{\theta}=[\theta_1,\theta_2,\theta_3,\theta_4]$. The parameters can be estimated by minimizing the sum of squared error

$$\sum_s e_s^2=\sum_s\left(f_s-\hat{\theta}w_s\right)^2$$

which leads to the following linear equations:

$$\hat{\theta}=\left(\sum w_s w_s^T\right)^{-1}\left(\sum w_s f_s\right)\quad \sigma^2=N^{-2}\sum\left(f_s-\hat{\theta}w_s\right)^2$$



MaZda on-line User's Manual

transform

3.3.6 Wavelet parameters

The discrete wavelet transform (DWT) is a linear transformation that operates on a data vector whose length is an integer power of two, transforming it into a numerically different vector of the same length. It is a tool that separates data into different frequency components, and then studies each component with resolution matched to its scale. DWT is computed with a cascade of filters followed by a factor 2 subsampling (Fig 3.3.2).

Fig. 3.3.2 DWT tree.

H and L denotes high and low-pass filters respectively. Boxes after each filter denote subsampling. Outputs of these filters are given by the following equations

$$a_{j+1}[p]=\sum_{n=-\infty}^{+\infty}l[n-2p]a_j[n]$$

$$d_{j+1}[p]=\sum_{n=-\infty}^{+\infty}h[n-2p]a_j[n]$$

Elements a_j are used for next step (scale) of the transform and elements d_j, called wavelet coefficients, determine output of the transform. *l[n]* and *h[n]* are coefficients of low and high-pas filters, respectively. One can assume that on scale *j*+1 there is only half of the number of *a* and *d* elements at scale *j*. This causes that DWT can be done until only two a_j elements remain in the analyzed signal. These elements are called scaling function coefficients.

DWT algorithm for pictures is similar. The DWT is performed firstly for all image rows and then for all columns (Fig.3.3.3).

MaZda on-line User's Manual

Ukryj Wstecz Drukuj Opcje

DWT algorithm for pictures is similar. The DWT is performed firstly for all image rows and then for all columns (Fig.3.3.3).

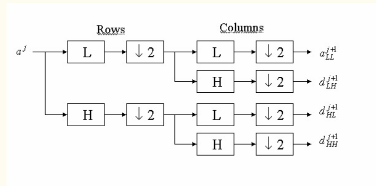
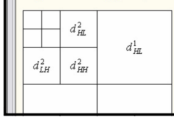


Fig. 3.3.3 Wavelet decomposition for two-dimensional pictures.

In this version of **MaZda**, only one set of DWT derived features is considered. It is a vector, which contains energies of wavelet coefficients calculated in subbands at successive scales.

To compute the wavelet features in the first step, Harr wavelet is calculated for whole image. As a result of this transform there are 4 subband images at each scale (Fig.3.3.4).



MaZda on-line User's Manual

Ukryj Wstecz Drukuj Opcje

In the next step, energy of ... is calculated at any considered scale in marked ROIs.

$$E_{subband,scale} = \frac{\sum_{x,y \in ROI} \left(d_{x,y}^{subband}\right)^2}{n}$$

where *n* is the number of pixels in ROI, both at given scale and subband.

Of course, ROIs are reduced in size in successive scales in order to correspond to subband image dimensions. In a given scale the energy is calculated only if ROI at this scale contains at least 4 points.

Output of this procedure is a vector of features containing energies of wavelet coefficients calculated in subbands at successive scales.

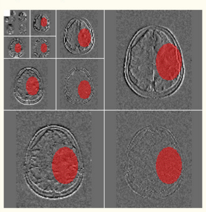


Fig. 3.3.5 Example of ROI reduced in successive scales

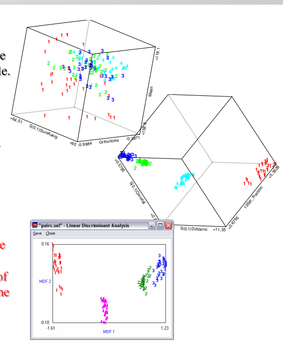
Pattern analysis/recognition

1. Several-hundred features mean analysis of a several-hundred-dimensional space. It would be time consuming, inefficient and in some cases not feasible.
2. Usually only a limited number of features carry relevant information needed for texture discrimination.

Selection of discriminative features and rejection of the rest is required.

3. The number of selected features is still unacceptably large.

Further reduction is feasible by transformation of the original features into a new space with lower dimensionality feature projection. This yields a set of new features, reduced in number in comparison to the original feature set.



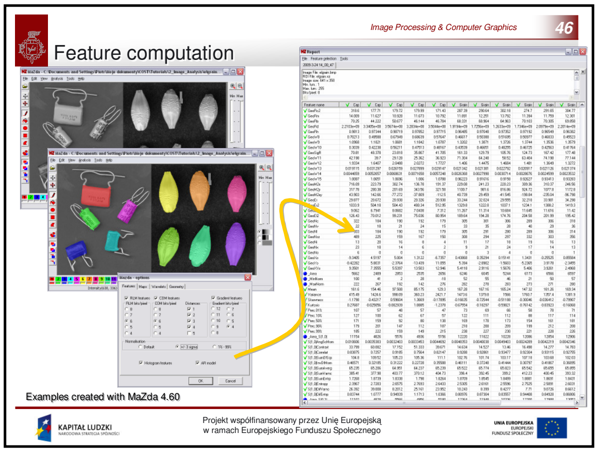
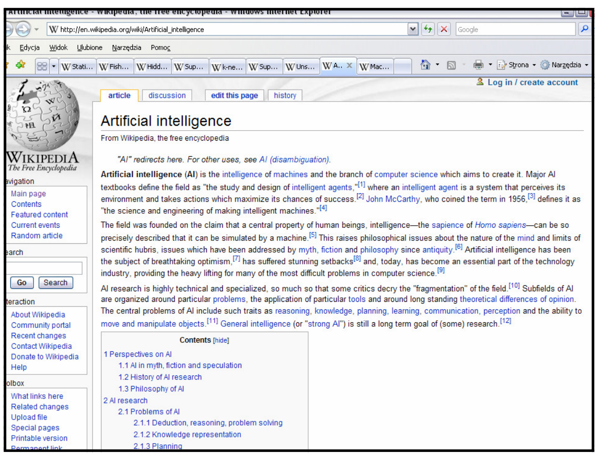
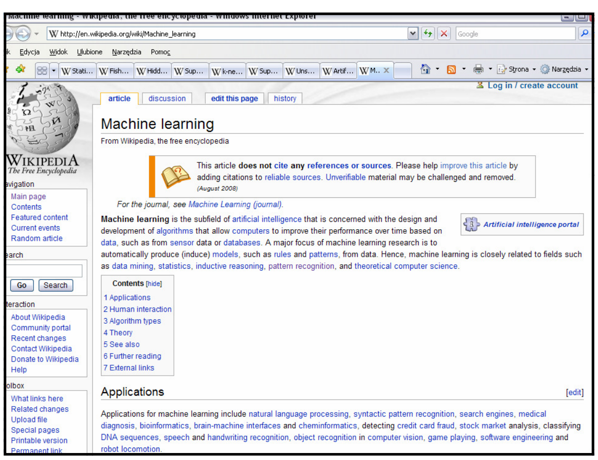


Image Processing & Computer Graphics
46
Feature computation
Examples created with MaZda 4.60
KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA


http://en.wikipedia.org/wiki/Artificial_intelligence
Log in / create account
article
discussion
edit this page
history
Artificial intelligence
From Wikipedia, the free encyclopedia
"AI" redirects here. For other uses, see AI (disambiguation).
Artificial intelligence (AI) is the intelligence of machines and the branch of computer science which aims to create it. Major AI textbooks define the field as "the study and design of intelligent agents,"[1] where an intelligent agent is a system that perceives its environment and takes actions which maximize its chances of success.[2] John McCarthy, who coined the term in 1956,[3] defines it as "the science and engineering of making intelligent machines."[4]
The field was founded on the claim that a central property of human beings, intelligence—the sapience of Homo sapiens—can be so precisely described that it can be simulated by a machine.[5] This raises philosophical issues about the nature of the mind and limits of scientific hubris, issues which have been addressed by myth, fiction and philosophy since antiquity.[6] Artificial intelligence has been the subject of breathtaking optimism,[7] has suffered stunning setbacks[8] and, today, has become an essential part of the technology industry, providing the heavy lifting for many of the most difficult problems in computer science.[9]
AI research is highly technical and specialized, so much so that some critics decry the "fragmentation" of the field.[10] Subfields of AI are organized around particular problems, the application of particular tools and around long standing theoretical differences of opinion. The central problems of AI include such traits as reasoning, knowledge, planning, learning, communication, perception and the ability to move and manipulate objects.[11] General intelligence (or "strong AI") is still a long term goal of (some) research.[12]
Contents [hide]
1 Perspectives on AI
1.1 AI in myth, fiction and speculation
1.2 History of AI research
1.3 Philosophy of AI
2 AI research
2.1 Problems of AI
2.1.1 Deduction, reasoning, problem solving
2.1.2 Knowledge representation
2.1.3 Planning
WIKIPEDIA
The Free Encyclopedia
Main page
Contents
Featured content
Current events
Random article
Go
Search
About Wikipedia
Community portal
Recent changes
Contact Wikipedia
Donate to Wikipedia
Help
What links here
Related changes
Upload file
Special pages
Printable version


http://en.wikipedia.org/wiki/Machine_learning
Log in / create account
article
discussion
edit this page
history
Machine learning
From Wikipedia, the free encyclopedia
This article does not cite any references or sources. Please help improve this article by adding citations to reliable sources. Unverifiable material may be challenged and removed. (August 2008)
For the journal, see Machine Learning (journal).
Machine learning is the subfield of artificial intelligence that is concerned with the design and development of algorithms that allow computers to improve their performance over time based on data, such as from sensor data or databases. A major focus of machine learning research is to automatically produce (induce) models, such as rules and patterns, from data. Hence, machine learning is closely related to fields such as data mining, statistics, inductive reasoning, pattern recognition, and theoretical computer science.
Artificial intelligence portal
Contents [hide]
1 Applications
2 Human interaction
3 Algorithm types
4 Theory
5 See also
6 Further reading
7 External links
Applications
[edit]
Applications for machine learning include natural language processing, syntactic pattern recognition, search engines, medical diagnosis, bioinformatics, brain-machine interfaces and cheminformatics, detecting credit card fraud, stock market analysis, classifying DNA sequences, speech and handwriting recognition, object recognition in computer vision, game playing, software engineering and robot locomotion.
WIKIPEDIA
The Free Encyclopedia
Main page
Contents
Featured content
Current events
Random article
Go
Search
About Wikipedia
Community portal
Recent changes
Contact Wikipedia
Donate to Wikipedia
Help
What links here
Related changes
Upload file
Special pages
Printable version

http://en.wikipedia.org/wiki/Pattern_recognition

Pattern recognition

From Wikipedia, the free encyclopedia

For other uses, see Pattern recognition (disambiguation).

Pattern recognition is a sub-topic of machine learning. It is "the act of taking in raw data and taking an action based on the category of the data".[citation needed] Most research in pattern recognition is about methods for supervised learning and unsupervised learning.

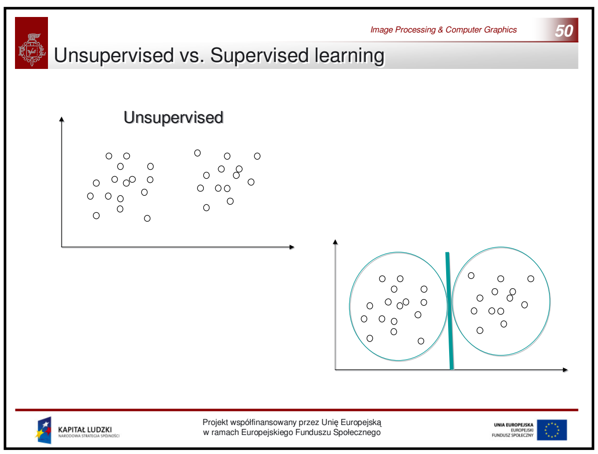
Pattern recognition aims to classify data (patterns) based either on *a priori* knowledge or on statistical information extracted from the patterns. The patterns to be classified are usually groups of measurements or observations, defining points in an appropriate multidimensional space. This is in contrast to **pattern matching**, where the pattern is rigidly specified.

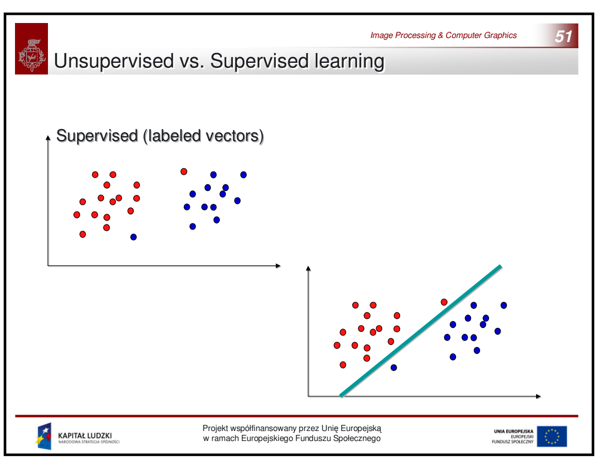
Overview [edit]

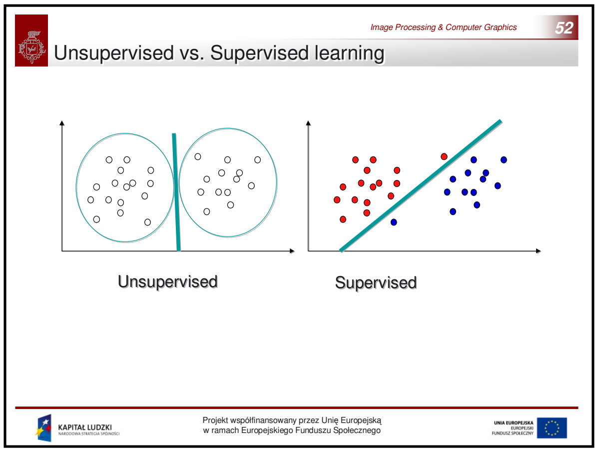
A complete pattern recognition system consists of a sensor that gathers the observations to be classified or described, a feature extraction mechanism that computes numeric or symbolic information from the observations, and a classification or description scheme that does the actual job of classifying or describing observations, relying on the extracted features.

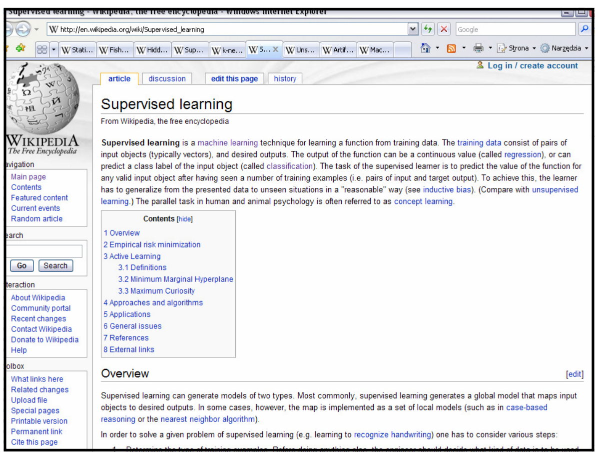
The classification or description scheme is usually based on the availability of a set of patterns that have already been classified or described. This set of patterns is termed the training set, and the resulting learning strategy is characterized as supervised learning. Learning can also be unsupervised, in the sense that the system is not given an *a priori* labeling of patterns, instead it itself establishes the classes based on the statistical regularities of the patterns.

The classification or description scheme usually uses one of the following approaches: statistical (or decision theoretic) or syntactic (or structural). Statistical pattern recognition is based on statistical characterizations of patterns, assuming that the patterns are generated by a probabilistic system. Syntactical (or structural) pattern recognition is based on the structural interrelationships of features. A wide









http://en.wikipedia.org/wiki/Supervised_learning

Supervised learning

From Wikipedia, the free encyclopedia

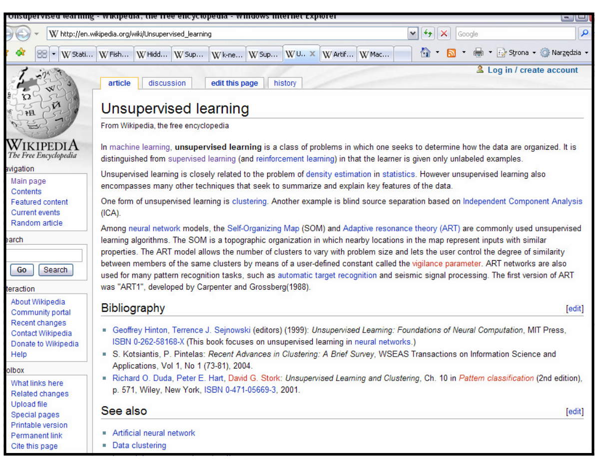
Supervised learning is a machine learning technique for learning a function from training data. The training data consist of pairs of input objects (typically vectors), and desired outputs. The output of the function can be a continuous value (called regression), or can predict a class label of the input object (called classification). The task of the supervised learner is to predict the value of the function for any valid input object after having seen a number of training examples (i.e. pairs of input and target output). To achieve this, the learner has to generalize from the presented data to unseen situations in a "reasonable" way (see inductive bias). (Compare with unsupervised learning.) The parallel task in human and animal psychology is often referred to as concept learning.

Contents [hide]

Overview [edit]

Supervised learning can generate models of two types. Most commonly, supervised learning generates a global model that maps input objects to desired outputs. In some cases, however, the map is implemented as a set of local models (such as in case-based reasoning or the nearest neighbor algorithm).

In order to solve a given problem of supervised learning (e.g. learning to recognize handwriting) one has to consider various steps:



http://en.wikipedia.org/wiki/Unsupervised_learning

Unsupervised learning

From Wikipedia, the free encyclopedia

In machine learning, **unsupervised learning** is a class of problems in which one seeks to determine how the data are organized. It is distinguished from supervised learning (and reinforcement learning) in that the learner is given only unlabeled examples.

Unsupervised learning is closely related to the problem of density estimation in statistics. However unsupervised learning also encompasses many other techniques that seek to summarize and explain key features of the data.

One form of unsupervised learning is clustering. Another example is blind source separation based on Independent Component Analysis (ICA).

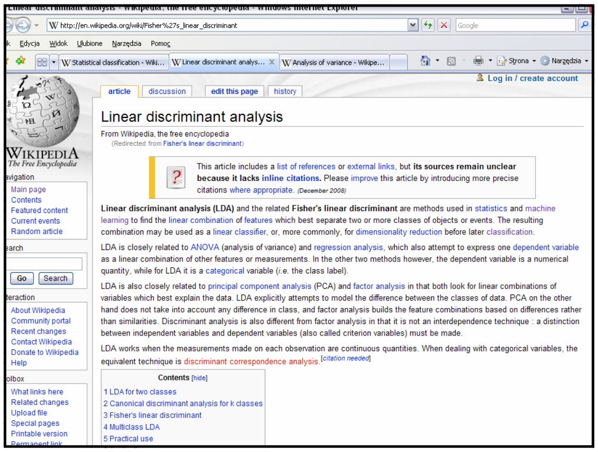
Among neural network models, the Self-Organizing Map (SOM) and Adaptive resonance theory (ART) are commonly used unsupervised learning algorithms. The SOM is a topographic organization in which nearby locations in the map represent inputs with similar properties. The ART model allows the number of clusters to vary with problem size and lets the user control the degree of similarity between members of the same clusters by means of a user-defined constant called the vigilance parameter. ART networks are also used for many pattern recognition tasks, such as automatic target recognition and seismic signal processing. The first version of ART was "ART1", developed by Carpenter and Grossberg(1988).

Bibliography [edit]

- Geoffrey Hinton, Terrence J. Sejnowski (editors) (1999): *Unsupervised Learning: Foundations of Neural Computation*, MIT Press, ISBN 0-262-58168-X (This book focuses on unsupervised learning in neural networks.)
- S. Kotsiantis, P. Pintelas: *Recent Advances in Clustering: A Brief Survey*, WSEAS Transactions on Information Science and Applications, Vol 1, No 1 (73-81), 2004.
- Richard O. Duda, Peter E. Hart, David G. Stork: *Unsupervised Learning and Clustering*, Ch. 10 in *Pattern classification* (2nd edition), p. 571, Wiley, New York, ISBN 0-471-05669-3, 2001.

See also [edit]

- Artificial neural network
- Data clustering



Linear discriminant analysis

From Wikipedia, the free encyclopedia
(Redirected from Fisher's linear discriminant)

This article includes a list of references or external links, but **its sources remain unclear because it lacks inline citations.** Please improve this article by introducing more precise citations where appropriate. *(December 2008)*

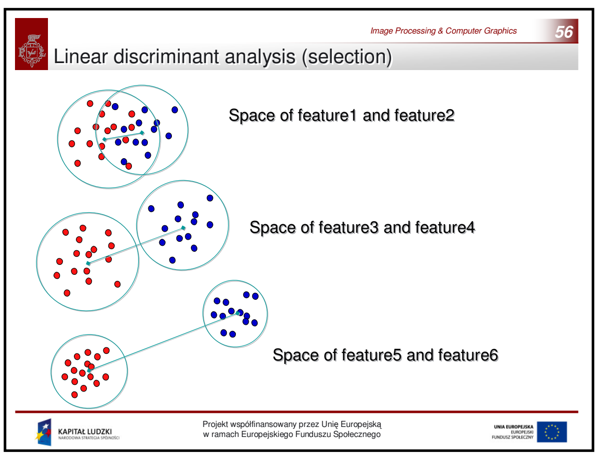
Linear discriminant analysis (LDA) and the related **Fisher's linear discriminant** are methods used in statistics and machine learning to find the linear combination of features which best separate two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification.

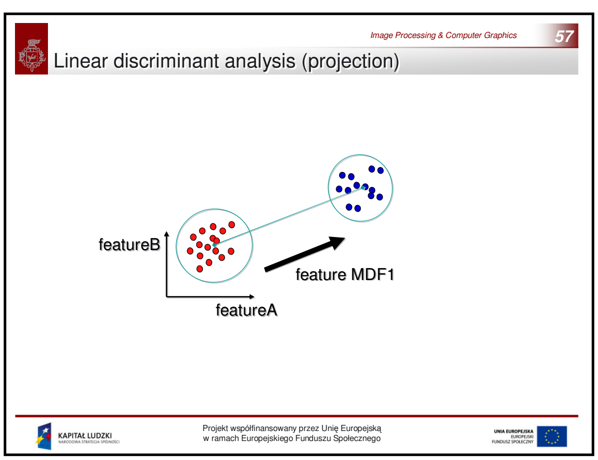
LDA is closely related to ANOVA (analysis of variance) and regression analysis, which also attempt to express one dependent variable as a linear combination of other features or measurements. In the other two methods however, the dependent variable is a numerical quantity, while for LDA it is a categorical variable (*i.e.* the class label).

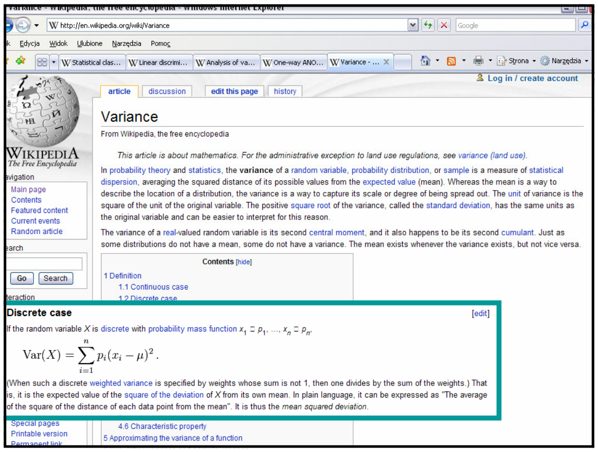
LDA is also closely related to principal component analysis (PCA) and factor analysis in that both look for linear combinations of variables which best explain the data. LDA explicitly attempts to model the difference between the classes of data. PCA on the other hand does not take into account any difference in class, and factor analysis builds the feature combinations based on differences rather than similarities. Discriminant analysis is also different from factor analysis in that it is not an interdependence technique : a distinction between independent variables and dependent variables (also called criterion variables) must be made.

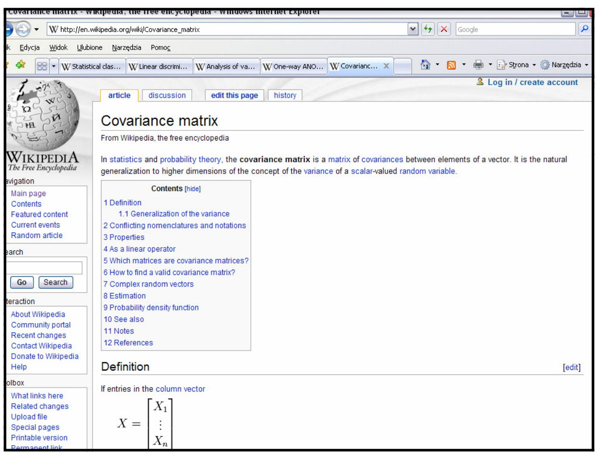
LDA works when the measurements made on each observation are continuous quantities. When dealing with categorical variables, the equivalent technique is discriminant correspondence analysis.[citation needed]

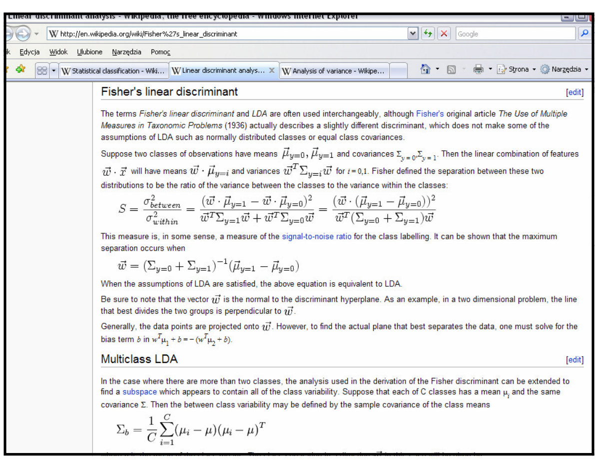
Contents [hide]

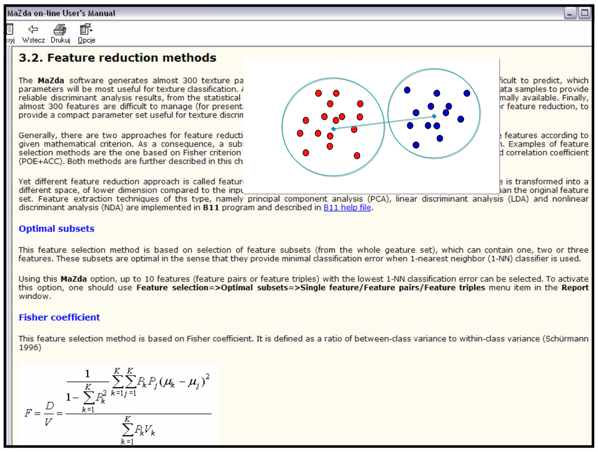





http://en.wikipedia.org/wiki/Variance
Variance
From Wikipedia, the free encyclopedia
This article is about mathematics. For the administrative exception to land use regulations, see variance (land use).
In probability theory and statistics, the variance of a random variable, probability distribution, or sample is a measure of statistical dispersion, averaging the squared distance of its possible values from the expected value (mean). Whereas the mean is a way to describe the location of a distribution, the variance is a way to capture its scale or degree of being spread out. The unit of variance is the square of the unit of the original variable. The positive square root of the variance, called the standard deviation, has the same units as the original variable and can be easier to interpret for this reason.
The variance of a real-valued random variable is its second central moment, and it also happens to be its second cumulant. Just as some distributions do not have a mean, some do not have a variance. The mean exists whenever the variance exists, but not vice versa.
Contents [hide]
1 Definition
1.1 Continuous case
Discrete case [edit]
If the random variable X is discrete with probability mass function x1 ↦ p1, ..., xn ↦ pn,
Var(X) = Σ_{i=1}^{n} p_i (x_i − μ)^2 .
(When such a discrete weighted variance is specified by weights whose sum is not 1, then one divides by the sum of the weights.) That is, it is the expected value of the square of the deviation of X from its own mean. In plain language, it can be expressed as "The average of the square of the distance of each data point from the mean". It is thus the mean squared deviation.
4.6 Characteristic property
5 Approximating the variance of a function


http://en.wikipedia.org/wiki/Covariance_matrix
Covariance matrix
From Wikipedia, the free encyclopedia
In statistics and probability theory, the covariance matrix is a matrix of covariances between elements of a vector. It is the natural generalization to higher dimensions of the concept of the variance of a scalar-valued random variable.
Contents [hide]
1 Definition
1.1 Generalization of the variance
2 Conflicting nomenclatures and notations
3 Properties
4 As a linear operator
5 Which matrices are covariance matrices?
6 How to find a valid covariance matrix?
7 Complex random vectors
8 Estimation
9 Probability density function
10 See also
11 Notes
12 References
Definition [edit]
If entries in the column vector
X = [X1 ⋮ Xn]


http://en.wikipedia.org/wiki/Fisher%27s_linear_discriminant
Fisher's linear discriminant [edit]
The terms Fisher's linear discriminant and LDA are often used interchangeably, although Fisher's original article The Use of Multiple Measures in Taxonomic Problems (1936) actually describes a slightly different discriminant, which does not make some of the assumptions of LDA such as normally distributed classes or equal class covariances.
Suppose two classes of observations have means μ_{y=0}, μ_{y=1} and covariances Σ_{y=0}, Σ_{y=1}. Then the linear combination of features w · x will have means w · μ_{y=i} and variances w^T Σ_{y=i} w for i = 0,1. Fisher defined the separation between these two distributions to be the ratio of the variance between the classes to the variance within the classes:
S = σ²_between / σ²_within = (w · μ_{y=1} − w · μ_{y=0})² / (w^T Σ_{y=1} w + w^T Σ_{y=0} w) = (w · (μ_{y=1} − μ_{y=0}))² / (w^T (Σ_{y=0} + Σ_{y=1}) w)
This measure is, in some sense, a measure of the signal-to-noise ratio for the class labelling. It can be shown that the maximum separation occurs when
w = (Σ_{y=0} + Σ_{y=1})^{-1} (μ_{y=1} − μ_{y=0})
When the assumptions of LDA are satisfied, the above equation is equivalent to LDA.
Be sure to note that the vector w is the normal to the discriminant hyperplane. As an example, in a two dimensional problem, the line that best divides the two groups is perpendicular to w.
Generally, the data points are projected onto w. However, to find the actual plane that best separates the data, one must solve for the bias term b in w^T μ1 + b = −(w^T μ2 + b).
Multiclass LDA [edit]
In the case where there are more than two classes, the analysis used in the derivation of the Fisher discriminant can be extended to find a subspace which appears to contain all of the class variability. Suppose that each of C classes has a mean μi and the same covariance Σ. Then the between class variability may be defined by the sample covariance of the class means
Σ_b = (1/C) Σ_{i=1}^{C} (μ_i − μ)(μ_i − μ)^T



MaZda on-line User's Manual

Wstecz Drukuj Opcje

3.2. Feature reduction methods

The **MaZda** software generates almost 300 texture pa... ...ficult to predict, which parameters will be most useful for texture classification. ... ...ata samples to provide reliable discriminant analysis results, from the statistical ... ...mally available. Finally, almost 300 features are difficult to manage (for present... ...r feature reduction, to provide a compact parameter set useful for texture discrim...

Generally, there are two approaches for feature reducti... ...e features according to given mathematical criterion. As a consequence, a sub... ...n. Examples of feature selection methods are the one based on Fisher criterion ... ...d correlation coefficient (POE+ACC). Both methods are further described in this ch...

Yet different feature reduction approach is called featur... ...e is transformed into a different space, of lower dimension compared to the inpu... ...an the original feature set. Feature extraction techniques of ths type, namely principal component analysis (PCA), linear discriminant analysis (LDA) and nonlinear discriminant analysis (NDA) are implemented in **B11** program and described in B11 help file.

Optimal subsets

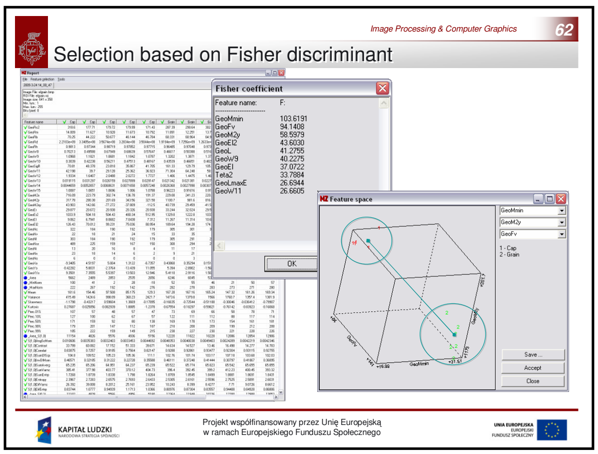
This feature selection method is based on selection of feature subsets (from the whole geature set), which can contain one, two or three features. These subsets are optimal in the sense that they provide minimal classification error when 1-nearest neighbor (1-NN) classifier is used.

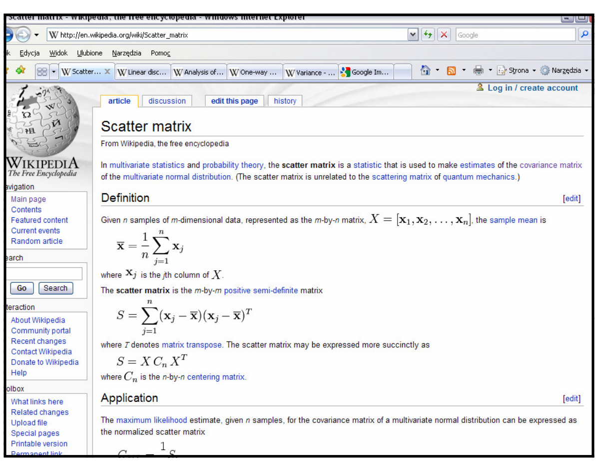
Using this **MaZda** option, up to 10 features (feature pairs or feature triples) with the lowest 1-NN classification error can be selected. To activate this option, one should use **Feature selection=>Optimal subsets=>Single feature/Feature pairs/Feature triples** menu item in the **Report** window.

Fisher coefficient

This feature selection method is based on Fisher coefficient. It is defined as a ratio of between-class variance to within-class variance (Schürmann 1996)

$$F = \frac{D}{V} = \frac{\frac{1}{1-\sum_{k=1}^{K} P_k^2} \sum_{k=1}^{K}\sum_{j=1}^{K} P_k P_j (\mu_k - \mu_j)^2}{\sum_{k=1}^{K} P_k V_k}$$





Scatter matrix - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/Scatter_matrix

article | discussion | edit this page | history

Log in / create account

Scatter matrix

From Wikipedia, the free encyclopedia

In multivariate statistics and probability theory, the **scatter matrix** is a statistic that is used to make estimates of the covariance matrix of the multivariate normal distribution. (The scatter matrix is unrelated to the scattering matrix of quantum mechanics.)

Definition [edit]

Given *n* samples of *m*-dimensional data, represented as the *m*-by-*n* matrix, $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, the sample mean is

$$\overline{\mathbf{x}} = \frac{1}{n}\sum_{j=1}^{n} \mathbf{x}_j$$

where $\mathbf{x}_j$ is the *j*th column of X.

The **scatter matrix** is the *m*-by-*m* positive semi-definite matrix

$$S = \sum_{j=1}^{n} (\mathbf{x}_j - \overline{\mathbf{x}})(\mathbf{x}_j - \overline{\mathbf{x}})^T$$

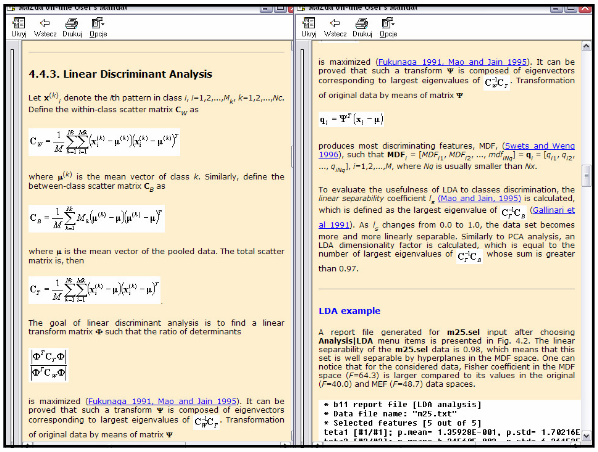
where *T* denotes matrix transpose. The scatter matrix may be expressed more succinctly as

$$S = X\, C_n\, X^T$$

where C_n is the *n*-by-*n* centering matrix.

Application [edit]

The maximum likelihood estimate, given *n* samples, for the covariance matrix of a multivariate normal distribution can be expressed as the normalized scatter matrix



MaZda on-line User's Manual

Ukryj Wstecz Drukuj Opcje

4.4.3. Linear Discriminant Analysis

Let $\mathbf{x}^{(k)}_i$ denote the *i*th pattern in class *i*, i=1,2,...,M_k, k=1,2,...,Nc. Define the within-class scatter matrix $\mathbf{C}_W$ as

$$C_W = \frac{1}{M}\sum_{k=1}^{Nc}\sum_{i=1}^{Mk}\left(\mathbf{x}_i^{(k)} - \boldsymbol{\mu}^{(k)}\right)\left(\mathbf{x}_i^{(k)} - \boldsymbol{\mu}^{(k)}\right)^T$$

where $\boldsymbol{\mu}^{(k)}$ is the mean vector of class *k*. Similarly, define the between-class scatter matrix $\mathbf{C}_B$ as

$$C_B = \frac{1}{M}\sum_{k=1}^{Nc} M_k\left(\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu}\right)\left(\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu}\right)^T$$

where $\boldsymbol{\mu}$ is the mean vector of the pooled data. The total scatter matrix is, then

$$C_T = \frac{1}{M}\sum_{k=1}^{Nc}\sum_{i=1}^{Mk}\left(\mathbf{x}_i^{(k)} - \boldsymbol{\mu}\right)\left(\mathbf{x}_i^{(k)} - \boldsymbol{\mu}\right)^T$$

The goal of linear discriminant analysis is to find a linear transform matrix $\boldsymbol{\Phi}$ such that the ratio of determinants

$$\frac{\left|\Phi^T C_T \Phi\right|}{\left|\Phi^T C_W \Phi\right|}$$

is maximized (Fukunaga 1991, Mao and Jain 1995). It can be proved that such a transform $\boldsymbol{\Psi}$ is composed of eigenvectors corresponding to largest eigenvalues of $C_W^{-1}C_T$. Transformation of original data by means of matrix $\boldsymbol{\Psi}$

$$\mathbf{q}_i = \boldsymbol{\Psi}^T\left(\mathbf{x}_i - \boldsymbol{\mu}\right)$$

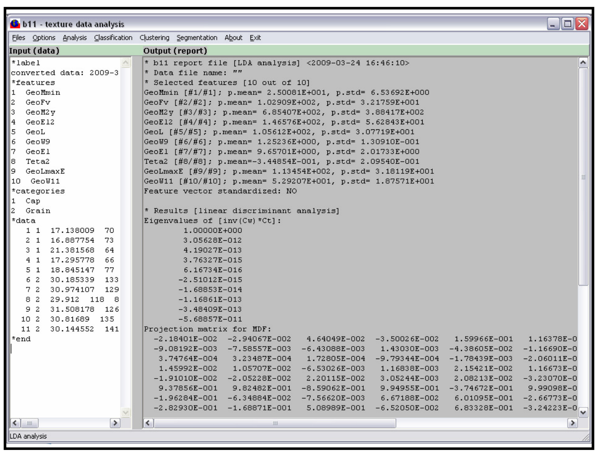
produces most discriminating features, MDF, (Swets and Weng 1996), such that $\mathbf{MDF}_i = [MDF_{i1}, MDF_{i2}, ..., mdf_{iNq}] = \mathbf{q}_i = [q_{i1}, q_{i2}, ..., q_{iNq}]$, i=1,2,...,M, where Nq is usually smaller than Nx.

To evaluate the usefulness of LDA to classes discrimination, the *linear separability* coefficient l_s (Mao and Jain, 1995) is calculated, which is defined as the largest eigenvalue of $C_T^{-1}C_B$ (Gallinari et al 1991). As l_s changes from 0.0 to 1.0, the data set becomes more and more linearly separable. Similarly to PCA analysis, an LDA dimensionality factor is calculated, which is equal to the number of largest eigenvalues of $C_T^{-1}C_B$ whose sum is greater than 0.97.

LDA example

A report file generated for **m25.sel** input after choosing **Analysis|LDA** menu items is presented in Fig. 4.2. The linear separability of the **m25.sel** data is 0.98, which means that this set is well separable by hyperplanes in the MDF space. One can notice that for the considered data, Fisher coefficient in the MDF space (F=64.3) is larger compared to its values in the original (F=40.0) and MEF (F=48.7) data spaces.

```
* b11 report file [LDA analysis]
* Data file name: "m25.txt"
* Selected features [5 out of 5]
teta1 [#1/#1]; p.mean= 1.35928E-001, p.std= 1.70216E
```



b11 - texture data analysis

Files Options Analysis Classification Clustering Segmentation About Exit

Input (data)

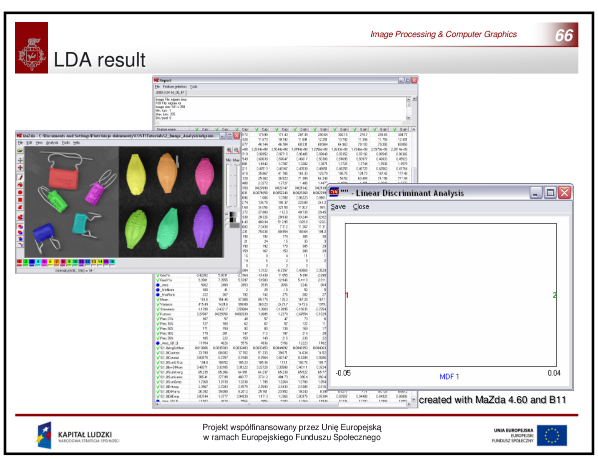
```
*label
converted data: 2009-3
*features
1  GeoMmin
2  GeoFv
3  GeoM2y
4  GeoE12
5  GeoL
6  GeoW9
7  GeoE1
8  Teta2
9  GeoLmaxE
10  GeoW11
*categories
1  Cap
2  Grain
*data
   1 1  17.138009  70
   2 1  16.887754  73
   3 1  21.381568  64
   4 1  17.295778  66
   5 1  18.845147  77
   6 2  30.185339  133
   7 2  30.974107  129
   8 2  29.912  118  8
   9 2  31.508178  126
  10 2  30.81689  135
  11 2  30.144552  141
*end
```

Output (report)

```
* b11 report file [LDA analysis] <2009-03-24 16:46:10>
* Data file name: ""
* Selected features [10 out of 10]
GeoMmin [#1/#1]; p.mean= 2.50081E+001, p.std= 6.53692E+000
GeoFv [#2/#2]; p.mean= 1.02909E+002, p.std= 3.21759E+001
GeoM2y [#3/#3]; p.mean= 6.85407E+002, p.std= 3.88417E+002
GeoE12 [#4/#4]; p.mean= 1.46576E+002, p.std= 5.62843E+001
GeoL [#5/#5]; p.mean= 1.05612E+002, p.std= 3.07719E+001
GeoW9 [#6/#6]; p.mean= 1.25236E+000, p.std= 1.30910E-001
GeoE1 [#7/#7]; p.mean= 9.65701E+000, p.std= 2.01733E+000
Teta2 [#8/#8]; p.mean=-3.44854E-001, p.std= 2.09540E-001
GeoLmaxE [#9/#9]; p.mean= 1.13454E+002, p.std= 3.18119E+001
GeoW11 [#10/#10]; p.mean= 5.29207E+001, p.std= 1.87571E+001
Feature vector standardized: NO

* Results [linear discriminant analysis]
Eigenvalues of [inv(Cw)*Ct]:
        1.00000E+000
        3.05628E-012
        4.19027E-013
        3.76327E-015
        6.16734E-016
       -2.51012E-015
       -1.68853E-014
       -1.16861E-013
       -3.48409E-013
       -5.68857E-011
Projection matrix for MDF:
  -2.18401E-002  -2.94067E-002   4.64049E-002  -3.50026E-002   1.59966E-001   1.16378E-0
  -9.08192E-003  -7.58557E-003  -6.43088E-003   1.43030E-003  -4.38605E-002  -1.16690E-0
   3.74764E-004   3.23487E-004   1.72805E-004  -9.79344E-004  -1.78439E-003  -2.06011E-0
   1.45992E-002   1.05707E-002  -6.53026E-003   1.16838E-003   2.15421E-002   1.16673E-0
  -1.91010E-002  -2.05228E-002   2.20115E-002   3.05244E-003   2.08213E-002  -3.23070E-0
   9.37856E-001   9.82482E-001  -8.59062E-001   9.94955E-001  -3.74672E-001   9.99098E-0
  -1.96284E-001  -6.34884E-002  -7.56620E-003   6.67188E-002   6.01095E-001  -2.66773E-0
  -2.82930E-001  -1.68871E-001   5.08989E-001  -6.52050E-002   6.83328E-001  -3.24223E-0
```

LDA analysis



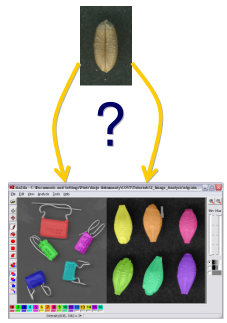
Classification (decision making)

Pattern classification problem can be approached by means of the so called ***decision functions***. The number of these functions is equal to the number of classes.

Let ***x***=*[x1, x2, …, xN]*. represent an *N*-dimensional pattern vector. For *M* pattern classes $\omega_1, \omega_2, \ldots \omega_M$, we are searching for *M* decision functions $d_1(\mathbf{x}), d_2(\mathbf{x}), \ldots, d_M(\mathbf{x})$ with a property that, if a pattern ***x*** belongs to class ω_i then

$$d_i(\mathbf{x}) > d_j(\mathbf{x}) \quad j = 1, 2, \ldots, M; i \neq j$$

In other words, decision function $d_i(\mathbf{x})$, "wins the competition" for assigning the input feature vector ***x*** to the pattern class ω_i.



Decision functions

The decision boundary between two arbitrary classes *i* and *j* ($i \neq j$) is defined by the function:

$$d_{ij}(\mathbf{x}) = d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0$$

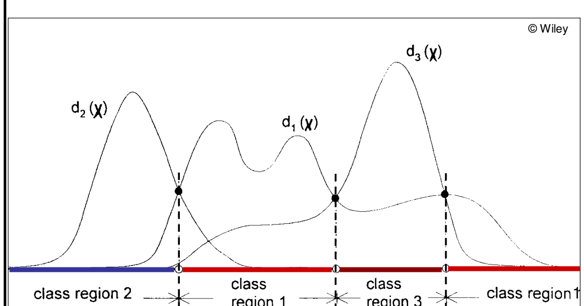
Then, for patterns of class ω_i :

$$d_{ij}(\mathbf{x}) > 0$$

and for patterns of class ω_j:

$$d_{ij}(\mathbf{x}) < 0.$$

Decision functions

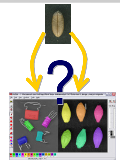


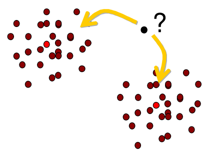
Minimum distance classifier

Assume that each pattern class is represented by a mean vector (also called a class *prototype*):

$$\boldsymbol{m}_i = \frac{1}{P_j} \sum_{x \in \omega_j} \boldsymbol{x}, \qquad j = 1,2,\ldots,M$$

where N_j is the number of pattern vectors from class ω_j. Possible way to determine the class membership of an unknown pattern vector $\boldsymbol{x}$ is to assign it to the class of its closest prototype vector.





Minimum distance classifier

If the Euclidean distance is used the distance measure is of the form:

$$D_j(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{m}_j\| \qquad j = 1,2,\ldots,M$$

and $\|\boldsymbol{x}\| = (\boldsymbol{x}^T \boldsymbol{x})^{1/2}$.

Feature vector $\boldsymbol{x}$ is assigned to class ω_j if $D_j(\boldsymbol{x})$ is the smallest distance.

Minimum distance classifier

The following distance function can be constructed

$$d_j(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{m}_j - \frac{1}{2} \boldsymbol{m}_j^T \boldsymbol{m}_j \qquad j = 1,2,\ldots,M$$

and assigning $\boldsymbol{x}$ to class ω_j if $d_j(\boldsymbol{x})$ gives the largest value.

The decision boundary between classes ω_i and ω_j for a minimum distance classifier is:

$$d_{ij}(\boldsymbol{x}) = d_i(\boldsymbol{x}) - d_j(\boldsymbol{x}) =$$
$$= \boldsymbol{x}^T(\boldsymbol{m}_i - \boldsymbol{m}_j) - \frac{1}{2}(\boldsymbol{m}_i - \boldsymbol{m}_j)^T(\boldsymbol{m}_i - \boldsymbol{m}_j) = 0$$

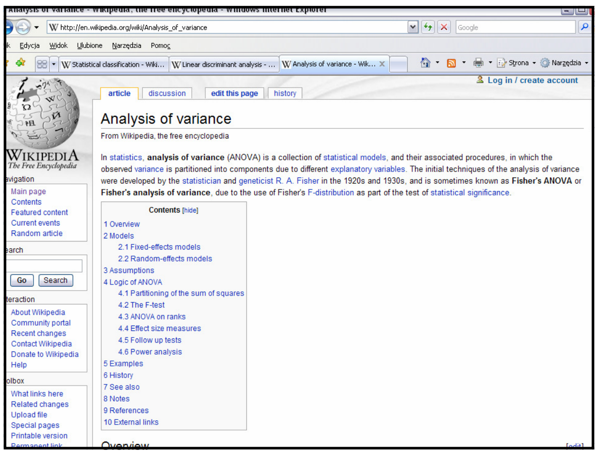
The surface defined by this equation is the perpendicular bisector to the line joining $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$. For $N=2$ the bisector is a line, for $N=3$ it is a plane, and for $N>3$ it is called a *hyperplane*.

Further reading

- Analysis of variance
- Principal Component Analysis
- Nonlinear Discriominant Analysis

- Linear classifier
- Margin classifier
- k-Nearest Neighbor classifier
- Support Vector Machines
- Perceptron Neural Network Classifier

KAPITAŁ LUDZKI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

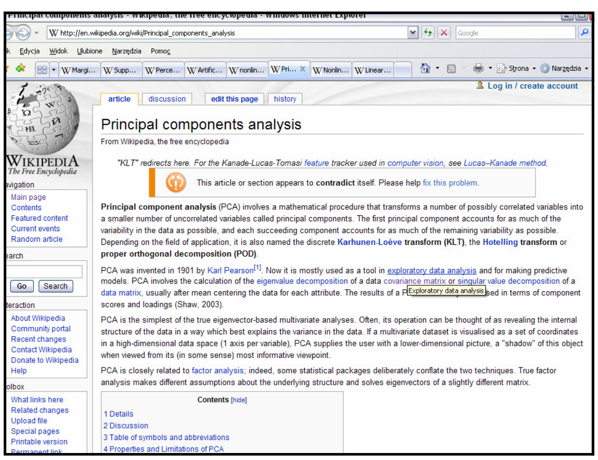


Analysis of variance

From Wikipedia, the free encyclopedia

In statistics, **analysis of variance** (ANOVA) is a collection of statistical models, and their associated procedures, in which the observed variance is partitioned into components due to different explanatory variables. The initial techniques of the analysis of variance were developed by the statistician and geneticist R. A. Fisher in the 1920s and 1930s, and is sometimes known as **Fisher's ANOVA** or **Fisher's analysis of variance**, due to the use of Fisher's F-distribution as part of the test of statistical significance.

Contents [hide]



Principal components analysis

From Wikipedia, the free encyclopedia

"KLT" redirects here. For the Kanade-Lucas-Tomasi feature tracker used in computer vision, see Lucas–Kanade method.

This article or section appears to **contradict** itself. Please help fix this problem.

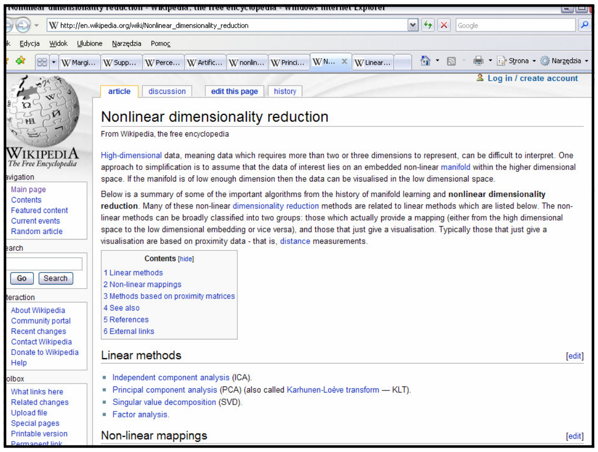
Principal component analysis (PCA) involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. Depending on the field of application, it is also named the discrete **Karhunen-Loève transform (KLT)**, the **Hotelling transform** or **proper orthogonal decomposition (POD)**.

PCA was invented in 1901 by Karl Pearson[1]. Now it is mostly used as a tool in exploratory data analysis and for making predictive models. PCA involves the calculation of the eigenvalue decomposition of a data covariance matrix or singular value decomposition of a data matrix, usually after mean centering the data for each attribute. The results of a P[Exploratory data analysis]sed in terms of component scores and loadings (Shaw, 2003).

PCA is the simplest of the true eigenvector-based multivariate analyses. Often, its operation can be thought of as revealing the internal structure of the data in a way which best explains the variance in the data. If a multivariate dataset is visualised as a set of coordinates in a high-dimensional data space (1 axis per variable), PCA supplies the user with a lower-dimensional picture, a "shadow" of this object when viewed from its (in some sense) most informative viewpoint.

PCA is closely related to factor analysis; indeed, some statistical packages deliberately conflate the two techniques. True factor analysis makes different assumptions about the underlying structure and solves eigenvectors of a slightly different matrix.

Contents [hide]



http://en.wikipedia.org/wiki/Nonlinear_dimensionality_reduction

Nonlinear dimensionality reduction

From Wikipedia, the free encyclopedia

High-dimensional data, meaning data which requires more than two or three dimensions to represent, can be difficult to interpret. One approach to simplification is to assume that the data of interest lies on an embedded non-linear manifold within the higher dimensional space. If the manifold is of low enough dimension then the data can be visualised in the low dimensional space.

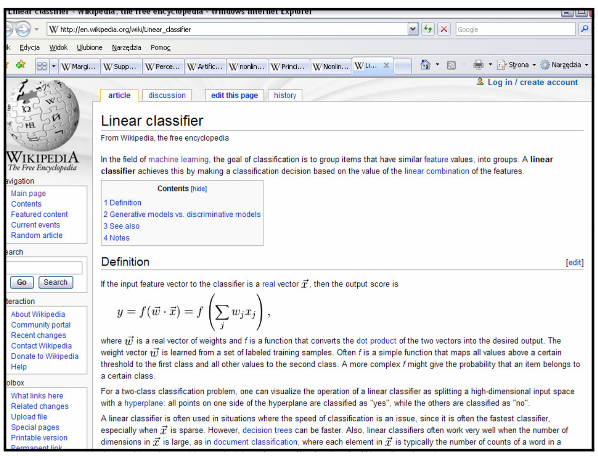
Below is a summary of some of the important algorithms from the history of manifold learning and **nonlinear dimensionality reduction**. Many of these non-linear dimensionality reduction methods are related to linear methods which are listed below. The non-linear methods can be broadly classified into two groups: those which actually provide a mapping (either from the high dimensional space to the low dimensional embedding or vice versa), and those that just give a visualisation. Typically those that just give a visualisation are based on proximity data - that is, distance measurements.

Contents [hide]

Linear methods [edit]

- Independent component analysis (ICA).
- Principal component analysis (PCA) (also called Karhunen-Loève transform — KLT).
- Singular value decomposition (SVD).
- Factor analysis.

Non-linear mappings [edit]



http://en.wikipedia.org/wiki/Linear_classifier

Linear classifier

From Wikipedia, the free encyclopedia

In the field of machine learning, the goal of classification is to group items that have similar feature values, into groups. A **linear classifier** achieves this by making a classification decision based on the value of the linear combination of the features.

Contents [hide]

Definition [edit]

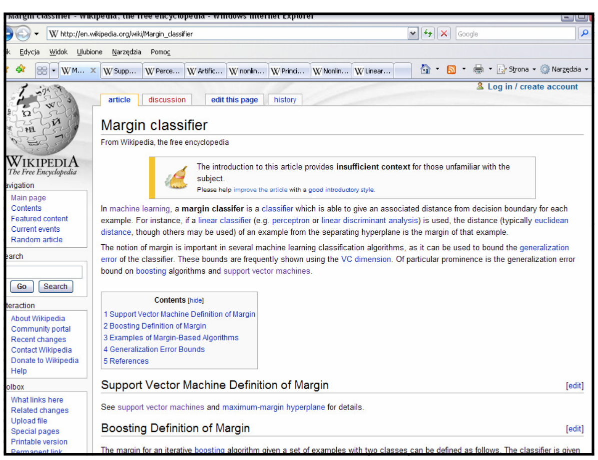
If the input feature vector to the classifier is a real vector $\vec{x}$, then the output score is

$$y = f(\vec{w} \cdot \vec{x}) = f\left(\sum_j w_j x_j\right),$$

where $\vec{w}$ is a real vector of weights and f is a function that converts the dot product of the two vectors into the desired output. The weight vector $\vec{w}$ is learned from a set of labeled training samples. Often f is a simple function that maps all values above a certain threshold to the first class and all other values to the second class. A more complex f might give the probability that an item belongs to a certain class.

For a two-class classification problem, one can visualize the operation of a linear classifier as splitting a high-dimensional input space with a hyperplane: all points on one side of the hyperplane are classified as "yes", while the others are classified as "no".

A linear classifier is often used in situations where the speed of classification is an issue, since it is often the fastest classifier, especially when $\vec{x}$ is sparse. However, decision trees can be faster. Also, linear classifiers often work very well when the number of dimensions in $\vec{x}$ is large, as in document classification, where each element in $\vec{x}$ is typically the number of counts of a word in a



http://en.wikipedia.org/wiki/Margin_classifier

Margin classifier

From Wikipedia, the free encyclopedia

The introduction to this article provides **insufficient context** for those unfamiliar with the subject.
Please help improve the article with a good introductory style.

In machine learning, a **margin classifer** is a classifier which is able to give an associated distance from decision boundary for each example. For instance, if a linear classifier (e.g. perceptron or linear discriminant analysis) is used, the distance (typically euclidean distance, though others may be used) of an example from the separating hyperplane is the margin of that example.

The notion of margin is important in several machine learning classification algorithms, as it can be used to bound the generalization error of the classifier. These bounds are frequently shown using the VC dimension. Of particular prominence is the generalization error bound on boosting algorithms and support vector machines.

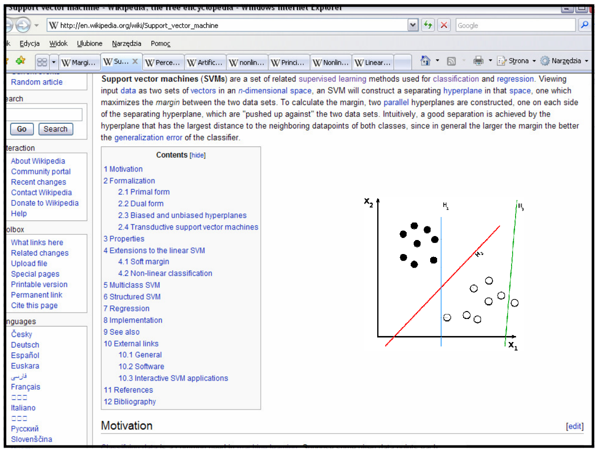
Contents [hide]

Support Vector Machine Definition of Margin [edit]

See support vector machines and maximum-margin hyperplane for details.

Boosting Definition of Margin [edit]

The margin for an iterative boosting algorithm given a set of examples with two classes can be defined as follows. The classifier is given

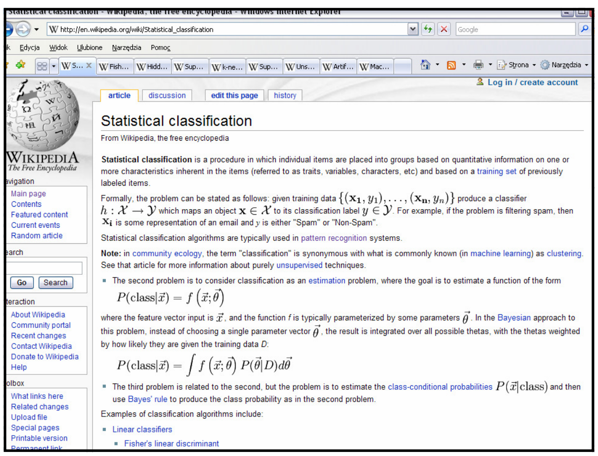

http://en.wikipedia.org/wiki/Support_vector_machine

Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. Viewing input data as two sets of vectors in an *n*-dimensional space, an SVM will construct a separating hyperplane in that space, one which maximizes the *margin* between the two data sets. To calculate the margin, two parallel hyperplanes are constructed, one on each side of the separating hyperplane, which are "pushed up against" the two data sets. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the neighboring datapoints of both classes, since in general the larger the margin the better the generalization error of the classifier.

Contents [hide]

1 Motivation
2 Formalization
2.1 Primal form
2.2 Dual form
2.3 Biased and unbiased hyperplanes
2.4 Transductive support vector machines
3 Properties
4 Extensions to the linear SVM
4.1 Soft margin
4.2 Non-linear classification
5 Multiclass SVM
6 Structured SVM
7 Regression
8 Implementation
9 See also
10 External links
10.1 General
10.2 Software
10.3 Interactive SVM applications
11 References
12 Bibliography

Motivation [edit]


http://en.wikipedia.org/wiki/Statistical_classification

Statistical classification

From Wikipedia, the free encyclopedia

Statistical classification is a procedure in which individual items are placed into groups based on quantitative information on one or more characteristics inherent in the items (referred to as traits, variables, characters, etc) and based on a training set of previously labeled items.

Formally, the problem can be stated as follows: given training data $\{(\mathbf{x_1}, y_1), \dots, (\mathbf{x_n}, y_n)\}$ produce a classifier $h : \mathcal{X} \to \mathcal{Y}$ which maps an object $\mathbf{x} \in \mathcal{X}$ to its classification label $y \in \mathcal{Y}$. For example, if the problem is filtering spam, then $\mathbf{x_i}$ is some representation of an email and y is either "Spam" or "Non-Spam".

Statistical classification algorithms are typically used in pattern recognition systems.

Note: in community ecology, the term "classification" is synonymous with what is commonly known (in machine learning) as clustering. See that article for more information about purely unsupervised techniques.

- The second problem is to consider classification as an estimation problem, where the goal is to estimate a function of the form

$$P(\text{class}|\vec{x}) = f\left(\vec{x};\vec{\theta}\right)$$

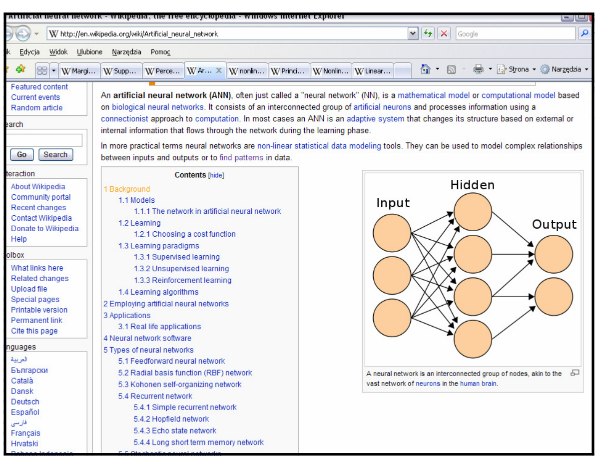
where the feature vector input is $\vec{x}$, and the function *f* is typically parameterized by some parameters $\vec{\theta}$. In the Bayesian approach to this problem, instead of choosing a single parameter vector $\vec{\theta}$, the result is integrated over all possible thetas, with the thetas weighted by how likely they are given the training data *D*:

$$P(\text{class}|\vec{x}) = \int f\left(\vec{x};\vec{\theta}\right) P(\vec{\theta}|D) d\vec{\theta}$$

- The third problem is related to the second, but the problem is to estimate the class-conditional probabilities $P(\vec{x}|\text{class})$ and then use Bayes' rule to produce the class probability as in the second problem.

Examples of classification algorithms include:

- Linear classifiers
  - Fisher's linear discriminant


http://en.wikipedia.org/wiki/Artificial_neural_network

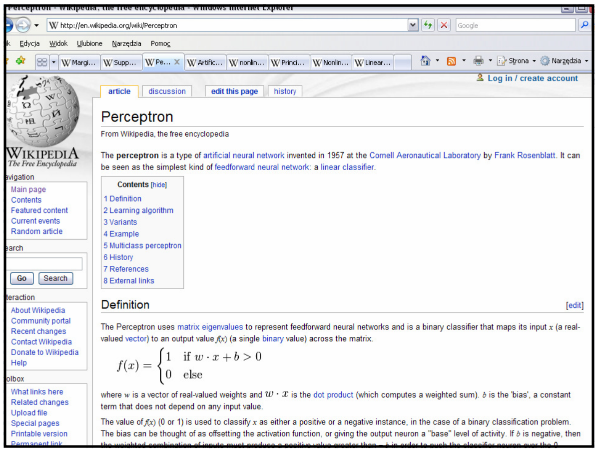
An **artificial neural network (ANN)**, often just called a "neural network" (NN), is a mathematical model or computational model based on biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase.

In more practical terms neural networks are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data.

Contents [hide]

1 Background
1.1 Models
1.1.1 The network in artificial neural network
1.2 Learning
1.2.1 Choosing a cost function
1.3 Learning paradigms
1.3.1 Supervised learning
1.3.2 Unsupervised learning
1.3.3 Reinforcement learning
1.4 Learning algorithms
2 Employing artificial neural networks
3 Applications
3.1 Real life applications
4 Neural network software
5 Types of neural networks
5.1 Feedforward neural network
5.2 Radial basis function (RBF) network
5.3 Kohonen self-organizing network
5.4 Recurrent network
5.4.1 Simple recurrent network
5.4.2 Hopfield network
5.4.3 Echo state network
5.4.4 Long short term memory network

A neural network is an interconnected group of nodes, akin to the vast network of neurons in the human brain.



Perceptron - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/Perceptron

Perceptron

From Wikipedia, the free encyclopedia

The **perceptron** is a type of artificial neural network invented in 1957 at the Cornell Aeronautical Laboratory by Frank Rosenblatt. It can be seen as the simplest kind of feedforward neural network: a linear classifier.

Contents [hide]

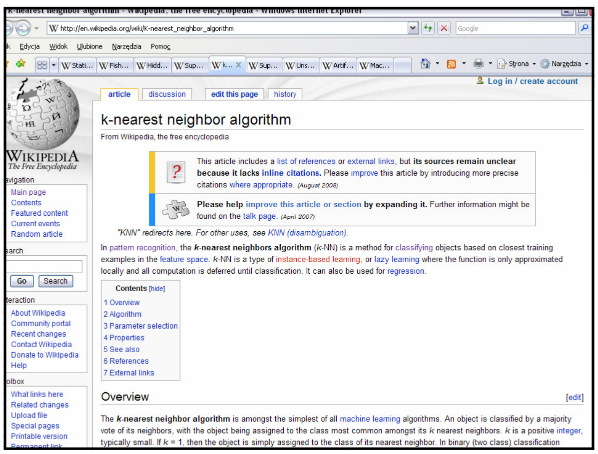
Definition [edit]

The Perceptron uses matrix eigenvalues to represent feedforward neural networks and is a binary classifier that maps its input x (a real-valued vector) to an output value $f(x)$ (a single binary value) across the matrix.

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{else} \end{cases}$$

where w is a vector of real-valued weights and $w \cdot x$ is the dot product (which computes a weighted sum). b is the 'bias', a constant term that does not depend on any input value.

The value of $f(x)$ (0 or 1) is used to classify x as either a positive or a negative instance, in the case of a binary classification problem. The bias can be thought of as offsetting the activation function, or giving the output neuron a "base" level of activity. If b is negative, then



k-nearest neighbor algorithm - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm

k-nearest neighbor algorithm

From Wikipedia, the free encyclopedia

This article includes a list of references or external links, but **its sources remain unclear because it lacks inline citations.** Please improve this article by introducing more precise citations where appropriate. *(August 2008)*

Please help improve this article or section by expanding it. Further information might be found on the talk page. *(April 2007)*

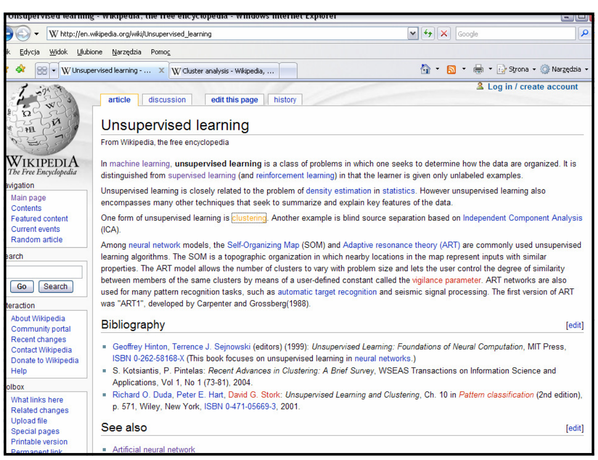
"KNN" redirects here. For other uses, see KNN (disambiguation).

In pattern recognition, the ***k*-nearest neighbors algorithm** (*k*-NN) is a method for classifying objects based on closest training examples in the feature space. *k*-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. It can also be used for regression.

Contents [hide]

Overview [edit]

The ***k*-nearest neighbor algorithm** is amongst the simplest of all machine learning algorithms. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors. k is a positive integer, typically small. If $k = 1$, then the object is simply assigned to the class of its nearest neighbor. In binary (two class) classification



Unsupervised learning - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/Unsupervised_learning

Unsupervised learning

From Wikipedia, the free encyclopedia

In machine learning, **unsupervised learning** is a class of problems in which one seeks to determine how the data are organized. It is distinguished from supervised learning (and reinforcement learning) in that the learner is given only unlabeled examples.

Unsupervised learning is closely related to the problem of density estimation in statistics. However unsupervised learning also encompasses many other techniques that seek to summarize and explain key features of the data.

One form of unsupervised learning is clustering. Another example is blind source separation based on Independent Component Analysis (ICA).

Among neural network models, the Self-Organizing Map (SOM) and Adaptive resonance theory (ART) are commonly used unsupervised learning algorithms. The SOM is a topographic organization in which nearby locations in the map represent inputs with similar properties. The ART model allows the number of clusters to vary with problem size and lets the user control the degree of similarity between members of the same clusters by means of a user-defined constant called the vigilance parameter. ART networks are also used for many pattern recognition tasks, such as automatic target recognition and seismic signal processing. The first version of ART was "ART1", developed by Carpenter and Grossberg(1988).

Bibliography [edit]

- Geoffrey Hinton, Terrence J. Sejnowski (editors) (1999): *Unsupervised Learning: Foundations of Neural Computation*, MIT Press, ISBN 0-262-58168-X (This book focuses on unsupervised learning in neural networks.)
- S. Kotsiantis, P. Pintelas: *Recent Advances in Clustering: A Brief Survey*, WSEAS Transactions on Information Science and Applications, Vol 1, No 1 (73-81), 2004.
- Richard O. Duda, Peter E. Hart, David G. Stork: *Unsupervised Learning and Clustering*, Ch. 10 in *Pattern classification* (2nd edition), p. 571, Wiley, New York, ISBN 0-471-05669-3, 2001.

See also [edit]

- Artificial neural network

B11 On-line User's Manual

Wstecz Drukuj Opcje

4.6.1. Agglomerative hierarchical clustering

Hierarchical clustering (HC) produces nested structure of data set partitions. These partitions consist of numerous small clusters at lower levels of hierarchy, combined in larger groups at higher levels. At the top-most level all samples are gathered in one common *conjoint* cluster. Construction of such nested structure can be processed in two alternative directions. The top-down approach is called *divisive hierarchical clustering* (DHC). It begins with placing all samples in a conjoint cluster, which is then partitioned into smaller groups in the subsequent stages of the procedure. In the bottom-up strategy, samples on initiation constitute separate (trivial) clusters which are next merged into larger groups (non-trivial clusters). This second approach, called *agglomerative hierarchical clustering* (AHC), is implemented in **b11**.

A convenient way to visualize the resulting structure of clusters is a hierarchical tree, called *dendrogram* (see Fig. 4.6.1). Data samples are tree leaves and are placed at the bottom. Horizontal edges are links between samples or groups of them, while vertical edges join links from lower levels with the links on upper levels. Height of a link, i.e. level at which a link is placed, is determined by the dissimilarity exhibited by clusters merged in that link. As it will be shown belowe, hierarchical tree helps determine the most appropriate level of data set partitioning.

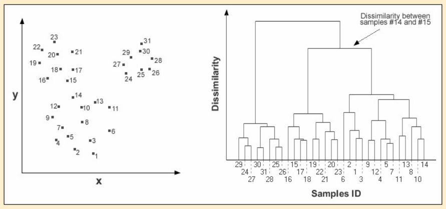
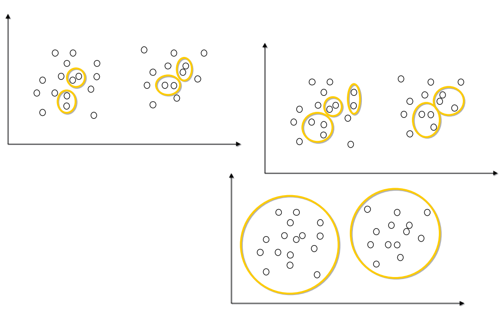


Fig. 4.6.1. On the left - data samples with assigned ID numbers. On the right - the corresponding dendrogram.

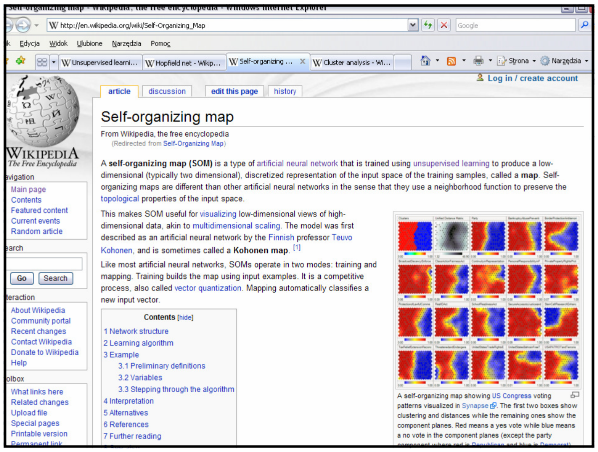
The basic object on which HC algorithm operates is an array of between-cluster dissimilarities, called proximity matrix **P**. If **X**=(x_1, ..., x_M) is the set

Unsupervised learning - Clustering



Further reading

- Neural networks
- Self-organizing maps
- Hierachical clustering
- Adaptive resonance



http://en.wikipedia.org/wiki/Self-Organizing_Map

Self-organizing map

From Wikipedia, the free encyclopedia
(Redirected from Self-Organizing Map)

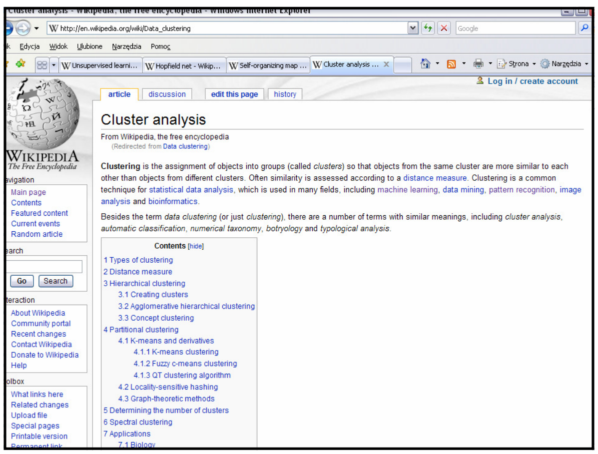
A **self-organizing map (SOM)** is a type of artificial neural network that is trained using unsupervised learning to produce a low-dimensional (typically two dimensional), discretized representation of the input space of the training samples, called a **map**. Self-organizing maps are different than other artificial neural networks in the sense that they use a neighborhood function to preserve the topological properties of the input space.

This makes SOM useful for visualizing low-dimensional views of high-dimensional data, akin to multidimensional scaling. The model was first described as an artificial neural network by the Finnish professor Teuvo Kohonen, and is sometimes called a **Kohonen map**. [1]

Like most artificial neural networks, SOMs operate in two modes: training and mapping. Training builds the map using input examples. It is a competitive process, also called vector quantization. Mapping automatically classifies a new input vector.

Contents [hide]

A self-organizing map showing US Congress voting patterns visualized in Synapse. The first two boxes show clustering and distances while the remaining ones show the component planes. Red means a yes vote while blue means a no vote in the component planes (except the party



http://en.wikipedia.org/wiki/Data_clustering

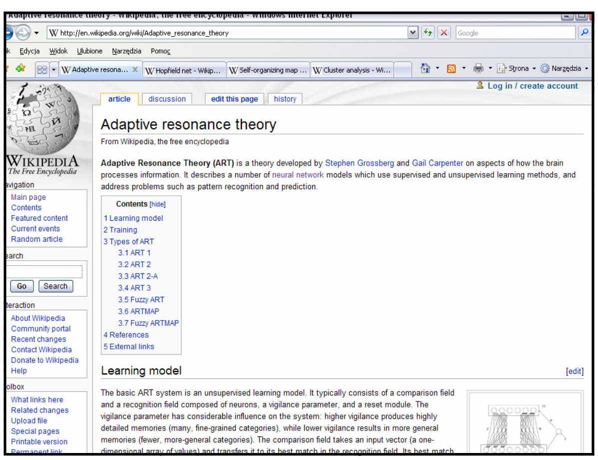
Cluster analysis

From Wikipedia, the free encyclopedia
(Redirected from Data clustering)

Clustering is the assignment of objects into groups (called *clusters*) so that objects from the same cluster are more similar to each other than objects from different clusters. Often similarity is assessed according to a distance measure. Clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics.

Besides the term *data clustering* (or just *clustering*), there are a number of terms with similar meanings, including *cluster analysis*, *automatic classification*, *numerical taxonomy*, *botryology* and *typological analysis*.

Contents [hide]



http://en.wikipedia.org/wiki/Adaptive_resonance_theory

Adaptive resonance theory

From Wikipedia, the free encyclopedia

Adaptive Resonance Theory (ART) is a theory developed by Stephen Grossberg and Gail Carpenter on aspects of how the brain processes information. It describes a number of neural network models which use supervised and unsupervised learning methods, and address problems such as pattern recognition and prediction.

Contents [hide]

Learning model [edit]

The basic ART system is an unsupervised learning model. It typically consists of a comparison field and a recognition field composed of neurons, a vigilance parameter, and a reset module. The vigilance parameter has considerable influence on the system: higher vigilance produces highly detailed memories (many, fine-grained categories), while lower vigilance results in more general memories (fewer, more-general categories). The comparison field takes an input vector (a one-dimensional array of values) and transfers it to its best match in the recognition field. Its best match

Summary, discussion and quiz

Summary, discussion and quiz

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

„ Image Processing and Computer Graphics"

**Prezentacja multimedialna współfinansowana przez
Unię Europejską w ramach
Europejskiego Funduszu Społecznego w projekcie pt.**
„Innowacyjna dydaktyka bez ograniczeń - zintegrowany rozwój Politechniki Łódzkiej - zarządzanie Uczelnią, nowoczesna oferta edukacyjna i wzmacniania zdolności do zatrudniania osób niepełnosprawnych"

Politechnika Łódzka
Politechnika Łódzka, ul. Żeromskiego 116, 90-924 Łódź, tel. (042) 631 28 83
www.kapitalludzki.p.lodz.pl