Proceedings of the 2007 IEEE Symposium on
Foundations of Computational Intelligence (FOCI 2007)

Classification of Objects by Means of Features

James F. Peters
Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba R3T 5V6, Canada
Email: {jfpeters}@ee.umanitoba.ca

Abstract— **The problem considered in this paper is how to classify objects by means of features. The solution to this problem stems from the seminal work by Zdzisław Pawlak starting in the early 1980s, which led to the discovery of rough sets and approximation spaces. The interpretation of features in this paper takes its inspiration from the Pawlak's approach to knowledge representation systems. Explicit in the original work of Pawlak is a distinction between attributes of objects and knowledge about objects. In this paper, knowledge about an object is represented by a measurement associated with a feature of an object. In general, a feature is an invariant characteristic of objects belonging to a class (*e.g.*, select *contour* (outline) as a feature, where all objects in a class have an identifiable contour). Associated with each feature is a set of probe functions, where each probe function maps objects to a value set. The distinction between features and corresponding probe function values is usually made in the study of pattern recognition. Examples of approximations, approximation spaces and a granular approach to recognition of patterns in pairs of images, are given. The contribution of this paper is a straightforward refinement of Pawlak's original approach to classifying objects.**

I. Introduction

The problem considered in this paper is how to classify objects by means of features, which is directly related to [21]. The solution to this problem stems from measurements that are associated with features of objects as well as sample objects extracted from the environment. Sample objects and vectors of corresponding probe function values are combined in information tables. Such tables lead to the construction of approximation spaces, which were introduced by Zdzisław Pawlak [20] starting in the early 1980s and which provide a basis for set approximations (see, *e.g.*, [14], [15], [16]). A basic architecture for feature-based perception (and pattern recognition) is shown in Fig. 1. The contribution of this paper is a straightforward refinement of Pawlak's original approach to classifying objects.

This paper is organized as follows. The distinction between attributes and features in classifying objects is briefly considered in Sect. II. Features and corresponding probe functions are presented in Sect. III. An overview of approximation spaces as a basis for perception is given in Sect. IV. A sample approach to transformation of an image as a prelude to pattern recognition relative to pairs of images is covered in Sect. V. Probe functions associated with an object feature in images and a sample approximation space are presented in Sect. VI.

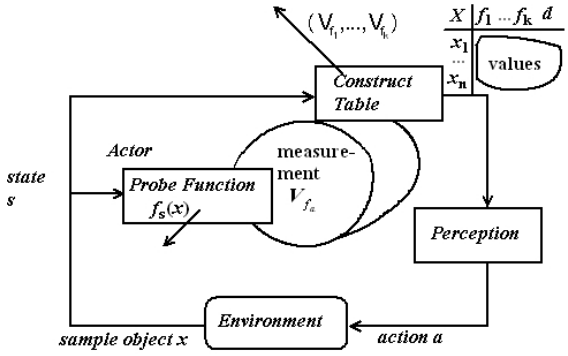

Fig. 1. Perceiving Sample Objects

II. Attributes, Features and Classifying Objects

An *attribute* is a quality regarded as characteristic or inherent in an object [11]. In philosophy, an attribute is a property of an object (e.g., spatial extension of a piece of wax) cite. The term *attribute* is commonly used in database theory [35], data mining [38], and philosophy [5]. In rough set theory [14], an attribute is treated as a partial function, which is a relation that associates each element of a set of objects (domain) with at most one element of a value set (codomain) [39]. The term *feature* was originally identified with the cast of a face [7]. More recently, the term *feature* is defined as the make, form, fashion or shape (of an object) [11]. This term comes from the Latin term *factura*, i.e., *facture*, which means the action or process of making an object or the result of an action or process (e.g., a work of art, image made with a digital camera). In effect, the term *feature* characterizes some aspect of the makeup of an object. From a philosophical perspective that can be traced back to Kant [8], features highlight an interest in the *appearances* of objects rather than calling attention to the properties or qualities that are somehow inherent in objects. The term *feature* is commonly used in pattern recognition theory [13], statistical learning theory [36], reinforcement learning [23], neural computing [1], science (e.g., ethology [9], [22]), image processing [6], [2], biotechnology, industrial inspection, the internet, radar, sonar, and speech recognition [4]. More recently, the term *feature* has been used in rough set theory [22], [2], [23].

There is a puzzle to solve here, namely, identifying appro-

priate contexts where one chooses attributes or features in rough set-based classification studies. From what has been observed so far, it is apparent historically, semantically and philosophically that there are perceived differences between attributes and features, even though there is an interest in the characteristics of objects in both cases. In data-intensive studies of objects where each attribute is the name of a single partial function, attributes are appropriate in classifying data. Features are favored over attributes in a growing number disciplines, where there is an interest in the composition (*e.g.*, construction, form, shape) of objects such as biological species, organism behaviour, sensor networks, sensor signals, sounds or images. This can be explained historically as well as mathematically, since it is possible to define more than one function for a single feature such as *colour* or *shape*. In pattern recognition, for example, it is common to associate more than one measurement (probe function value) with a single feature(see, *e.g.*, [6], [13]). In that case, it is appropriate to choose features rather than attributes in classifying objects. It is the one-to-many relation between single features and corresponding probe functions that underlies what follows in this article.

III. Features and Measurements

It was Zdzisław Pawlak who proposed classifying objects by means of their attributes considered in the context of an approximation space [17]. Explicit in the original work of Pawlak is a distinction between attributes of objects and knowledge about objects. In this paper, knowledge about an object is represented by measurements associated with each feature of an object. In general, a feature is an invariant property of objects belonging to a class [37]. The distinction between features and corresponding measurements associated with features is usually made in the study of pattern recognition (see, *e.g.*, [10], [13]). In this article, the practice begun by Pawlak [17] is represented in the following way. Let A denote a set of features for objects in a set X. For each $a \in A$, we associate a function f_a that maps X to some set V_{f_a} (range of f_a). The value of $f_a(x)$ is a measurement associated with feature a of an object $x \in X$. The function f_a is called a *probe* [13]. By $Inf_B(x)$, where $B \subseteq A$ and $x \in U$ we denote the *signature* of x, i.e., the set $\{(a, f_a(x)) : a \in B\}$. If the set $B = \{a_1, \ldots, a_m\}$, then Inf_B is identified with a vector $(f_{a_1}(x), \ldots, f_{a_m}(x))$ of probe function values for features in B.

IV. Approximation spaces

This section briefly presents some fundamental concepts in rough set theory and approximation spaces introduced by Zdzisław Pawlak during the early 1980s [17], elaborated in [12], [19], generalized in [30], [33] and extended in [21]. For computational reasons, a syntactic representation of knowledge in rough set theory is provided in the form of *data tables*.

A. Rough sets

Let U be a non-empty finite set (called a *universe*) and let $\mathcal{P}(U)$ denote the power set of U, *i.e.*, the family of all subsets of U. Elements of U may be, for example, objects, behaviours, or perhaps states. A *feature* $\mathcal{F}$ of elements in U is measured by an associated probe function $f = f_{\mathcal{F}}$ whose range is denoted by $\mathcal{V}_f$, called the *value set* of f; that is, $f : U \to \mathcal{V}_f$. There may be more than one probe function for each feature. For example, a feature of an object may be its weight, and different probe functions for weight are found by different weighing methods; or a feature might be colour, with probe functions measuring, *e.g.*, red, green, blue, hue, intensity, and saturation. The similarity or equivalence of objects can be investigated quantitatively by comparing a sufficient number of object features by means of probes [13]. For present purposes, to each feature there is only one probe function associated and its value set is taken to be a finite set (usually of real numbers). Thus one can identify the set of features with the set of associated probe functions, and hence we use f rather than $f_{\mathcal{F}}$ and call $\mathcal{V}_f = \mathcal{V}_{\mathcal{F}}$ a set of feature values. If F is a finite set of probe functions for features of elements in U, the pair (U, F) is called a *data table*, or *information system* (IS).

For each subset $B \subseteq F$ of probe functions, define the binary relation $\sim_B = \{(x, x') \in U \times U : \forall f \in B, f(x) = f(x')\}$. Since each $\sim_B$ is an equivalence relation, for $B \subset F$ and $x \in U$ let $[x]_B$ denote the equivalence class, or *block*, containing x, that is,

$$[x]_B = \{x' \in U : \forall f \in B, f(x') = f(x)\} \subseteq U.$$

If $(x, x') \in \sim_B$ (also written $x \sim_B x'$) then x and x' are said to be *indiscernible* with respect to all feature probe functions in B, or simply, *B-indiscernible*.

Information about a sample $X \subseteq U$ can be approximated from information contained in B by constructing a B-lower approximation

$$B_*X = \bigcup_{x:[x]_B \subseteq X} [x]_B,$$

and a B-upper approximation

$$B^*X = \bigcup_{x:[x]_B \cap X \neq \emptyset} [x]_B.$$

The B-lower approximation B_*X is a collection of blocks of sample elements that can be classified with full certainty as members of X using the knowledge represented by features in B. By contrast, the B-upper approximation B^*X is a collection of blocks of sample elements representing both certain and possibly uncertain knowledge about X. Whenever $B_*X \subsetneq B^*X$, the sample X has been classified imperfectly, and is considered a rough set. In this paper, only B-lower approximations are used.

B. Generalized approximation spaces

The basic model for an approximation space was introduced by Pawlak in 1981 [17], elaborated in [12], [19], generalized

in [30], [33], and applied in a number of ways (see, *e.g.*, [21], [22], [23], [31]). An approximation space serves as a formal counterpart of perception or observation [12], and provides a framework for approximate reasoning about vague concepts.

To be precise about what "approximation space" means, some definitions are required. A *neigbourhood function* on a set U is a function $N : U \rightarrow \mathcal{P}(U)$ that assigns to each $x \in U$ some subset of U containing x. A particular kind of neigbourhood function on U is determined by any partition $\xi : U = U_1 \cup \cdots \cup U_d$, where for each $x \in U$, the ξ-neigbourhood of x, denoted $N_\xi(x)$, is the U_i that contains x. In terms of equivalence relations in Section IV-A, for some fixed $B \subset F$ and any $x \in U$, $[x]_B = N_B(x)$ naturally defines a neigbourhood function N_B. In effect, the neigbourhood function N_B defines an *indiscernibility* relation, which defines for every object x a set of like-wise defined objects, that is objects whose value sets agree precisely (see, *e.g.*, [24]). An *overlap function* ν on U is any function $\nu : \mathcal{P}(U) \times \mathcal{P}(U) \rightarrow [0, 1]$ that reflects the degree of overlap between two subsets of U.

A *generalized approximation space* (GAS) is a triple (U, N, ν), where U is a non-empty set of objects, N is a neigbourhood function on U, and ν is an overlap function on U. In this work, only indiscernibility relations determine N.

A set $X \subseteq U$ is *definable* in a GAS if, and only if X is the union of some values of the neigbourhood function. Specifically, any information system (U, F) and any $B \subseteq F$ naturally defines parameterized approximation spaces $AS_B = (U, N_B, \nu)$, where $N_B = [x]_B$, a B-indiscernibility class in a partition of U.

A standard example (see, *e.g.*, [30]) of an overlap function is *standard rough inclusion*, defined by $\nu_{SRI}(X, Y) = \frac{|X \cap Y|}{|X|}$ for non-empty X. Then $\nu_{SRI}(X, Y)$ measures the portion of X that is included in Y. An analogous notion is used in this work. If $U = U_{beh}$ is a set of behaviours, let $Y \subseteq U$ represent a kind of "standard" for evaluating sets of similar behaviours. For any $X \subset U$, we are interested in how well X "covers" Y, and so we consider another form of overlap function, namely, *standard rough coverage* ν_{SRC}, defined by (1).

$$\nu_{SRC}(X, Y) = \begin{cases} \frac{|X \cap Y|}{|Y|} & \text{if } Y \neq \emptyset, \\ 1 & \text{if } Y = \emptyset. \end{cases} . \qquad (1)$$

In other words, $\nu_{SRC}(X, Y)$ returns the fraction of Y that is covered by X. In the case where $X = Y$, then $\nu_{\text{SRC}}(X, Y) = 1$. The minimum coverage value $\nu_{\text{SRC}}(X, Y) = 0$ is obtained when $X \cap Y = \emptyset$. One might note that for non-empty sets, $\nu_{SRC}(X, Y) = \nu_{SRI}(Y, X)$

An overview of the relation between approximation space-based perception and learning is shown at a very high level in Fig. 2. The basic idea is to learn desirable (fulfilling, rewarding) actions based on knowledge gleaned from the overlap between neighborhoods of similar objects and some standard (*e.g.*, lower approximation of a set such as a decision class (*i.e.*, concept) D. The patterns discovered from a perceptual view of decision tables constructed from sample objects from the environment, provide a basis for learning. A brief view of how pattern recognition can be carried out on pairs of images is briefly presented in what follows. A consideration of learning resulting from pattern recognition is outside the scope of this paper.

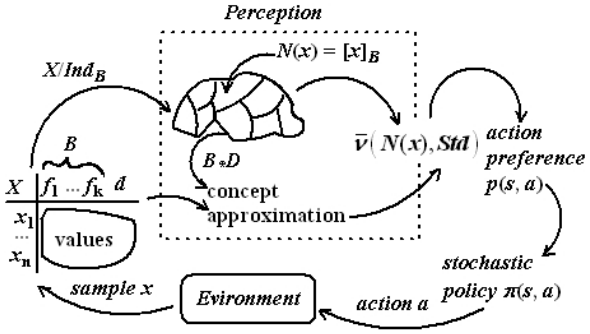

Fig. 2. Basic Structures in Perception-Based Learning

V. Preliminary Image Processing

An approach to quantizing and averaging the colors in a digital image based on [2] is briefly presented in this section. This is done to illustrate how a digital image can be transformed to facilitation pattern recognition in comparing pairs of images. Quantization has been defined as a process of converting analog signal to digital signal [3]. In what follows, the Lloyd quantization algorithm [3] has been used, see Alg. 1.

Algorithm 1: The Lloyd Algorithm [3] (alg. Q_n)

Input: image I, required number of colors n
Output: optimal codebook with n entries C_{opt}

Initialize codebook C_1 with n entries randomly;
set $m = 1$;
repeat
  Based on codebook C_m and using the nearest neighbour condition, partition image I into quantization cells R_m;
  Using centroid condition, find optimal codebook C_{m+1} for cells R_m;
  Set $m = m + 1$;
until *distortion caused by* C_m *is small enough*
Set $C_{opt} = C_m$;

The details concerning Alg. 1 are given in [2]. Briefly, the Lloyd algorithm consists of main two steps, which are repeated until the distortion caused by the codebook is small enough. The first step is the partitioning of the input image based on the current codebook. The partitioning is performed using nearest neighbour condition, e.g. each pixel is assigned to the cell closest to the color of given pixel. In the second step, a new codebook is created based on the partitioning from the first

step. Each codebook entry is replaced by a centroid of all colors of pixels from the corresponding cell.

Assume, that an original image denoted by I_o is given. First, quantization reducing number of colors to n_1 is performed. This step is denoted by formula 2 to obtain a quantized image denoted by $I_{q_{n_1}}$ (symbol Q represents the algorithm 1, where I_o is the input image I and n_1 is the required number of colors n).

$$I_o \xrightarrow{Q_{n_1}} I_{q_{n_1}} \tag{2}$$

As a result of quantization, the quantized image $I_{q_{n_1}}$ contains only n_1 colors. In the next step, the information from $I_{q_{n_1}}$ image is used to average the colors among all pixels, which are *connected*.

In quantized image, regions of pixels of the same color can be identified. These regions create segments. To each such segment is assigned a color, which is an average of all original colors from pixels belonging to this region. This step is denoted by the formula in (3), see also Alg. 2.

$$I_{q_{n_1}} \xrightarrow{Av_{I_o}} I_{Av_{n_1}} \tag{3}$$

The image $I_{Av_{n_1}}$ resulting from (3) has more than n_1 colors, where pixels are grouped into segments. This procedure, namely steps defined in (2) and (3), is repeated. The number of colors gradually decreases in consecutive iterations so that the creation of segments can be observed.

Algorithm 2: The Spatial Color Averaging (alg. $Av_{I_{s_n}}$)

Input: image I
Output: averaged image I_A

Mark all pixels from I as *not processed*;
foreach *pixel* p *not processed in* I **do**
  Find segment $S(p)$ in I containing pixel p;
  Assign to each corresponding pixel in I_A from $S(p)$ an average color of all pixels from $S(p)$;
  Set all pixels from $S(p)$ as *processed*;
end

The unwanted effect of the algorithm defined this way is that if a segment is created at some step, there are no chances to change it in consecutive steps. In other words, the first quantization plays a crucial role in the entire process. In addition, the resulting image still contains a lot of details (even though the number of colors has been reduced). An example of such an image processed using seven iterations described by the succession of mappings in (4), where numbers n_i for $i = 1, 2, ..., 6$[1] are 256, 64, 32, 16, 12, 8 and 4 is shown in Fig. 3.

$$I_{s_{n_{i-1}}} \xrightarrow{Q_{n_i}} I_{q_{n_i}} \xrightarrow{Av_{I_{s_{n_{i-1}}}}} I_{Av_{n_i}} \tag{4}$$

[1]For $i = 0$ it is assumed that $I_{s_{n_0}} = I_o$, and after each iteration $I_{s_{n_{i-1}}} = I_{Av_{n_{i-1}}}$.

Fig. 3. Image after 7 iterations of (4)

Fig. 4. Image after 7 iterations of (5)

To make the entire segmentation process more robust and force the creation of bigger segments, one extra step for each stage defined by (4) is added. That is, after the colors are recreated from the original image, a 3 by 3 median filter is used. This causes almost uniform areas to blur even more and allows edges of neighboring segments to overlap. As a result, all small details from the image are lost, and big uniform segments are formed instead. The final formula describing one step of this iterated algorithm is shown in (5).

$$I_{s_{n_{i-1}}} \xrightarrow{Q_{n_i}} I_{q_{n_i}} \xrightarrow{Av_{I_{s_{n_{i-1}}}}} I_{Av_{n_i}} \xrightarrow{M_{3x3}} I_{s_{n_i}} \tag{5}$$

The M_{3x3} symbol denotes the median filter which is applied to each pixel from an input image. The median filter is applied to 3 by 3 neighborhood of given pixel $p(x, y)$.

$$M_{3x3}(p) = median\{p(x-1, y-1), \ldots, p(x+1, y+1)\} \tag{6}$$

In order to find the median, all pixels are sorted by their color value and the one in the middle (e.g. at the 5-th place) is chosen.

The image that results from 7 iterations of (5) is shown in Fig. 4. There are still many small segments, but compared with the image in Fig. 3, where the median filter was not used, the number of small segments has been greatly reduced.

VI. Approach to Pattern Recognition

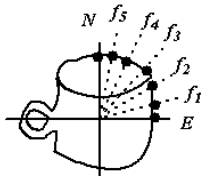

Fig. 5. Prototype Image

The problem considered here is to determine whether there is a correspondence between an object in a prototype image $\mathcal{I}$

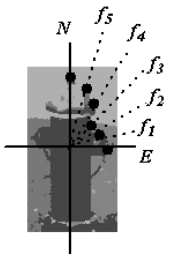

Fig. 6. Sample Image

TABLE I
SAMPLE IMAGE CONTOUR TABLE

x_i	N	E	NE_1	NE_2	d
x_0	0.1	0.1	0.1	0.1	1
x_1	0.5	0.8	0.1	0.1	0
x_2	0.6	0.1	0.1	0.1	0
x_3	0.7	0.3	0.1	0.1	1
x_4	0.1	0.1	0.1	0.1	1
x_5	0.5	0.8	0.1	0.1	0
x_6	0.010	0.1	0.1	0.1	1
x_7	0.025	0.1	0.1	0.1	0
x_8	0.01	0.1	0.1	0.1	1
x_9	0.5	0.8	0.1	0.1	0
x_{10}	0.01	0.1	0.1	0.1	0

(*e.g.*, cup in Fig. 5) and an object in a sample image I_1 (*e.g.*, fire hydrant in Fig. 6). By way of illustration, consider *contour* as a helpful feature in considering the form of various objects. Let $I_1, \mathcal{I}, f$ denote sample image, prototype image, probe function associated with *contour*, respectively. Then, following the approach suggested in [13], recognition of objects that are approximately the same is defined by comparing probe function values in

$$\mathcal{I} \approx (I_1)T \Leftrightarrow \forall f. |f(\mathcal{I}) - f(I_1)| < \varepsilon, \varepsilon \in [0, 1]$$

where $\mathcal{I}$ is approximately the same as I_1 after some transformation T, iff $|f(\mathcal{I}) - f(I_1)| < \varepsilon$ for all f associated with *contour*. For example, using Alg. 1 and Alg. 2, the transformation results in an image where the number of colors in an image is reduced and all small details are lost (big uniform segments are formed).

A. Contour Probe Functions

An obvious choice basis for setting up contour probe functions is the taxicab distance. The city block distance or taxicab distance [29] is so named because it is the shortest distance a car would drive in a city laid out in square blocks) [29]. A taxicab distance is computed using

$$d(x, y) = |x - 0| + |y - 0| = |x| + |y|,$$

which was first discovered by Hermann Minkowski (1864-1909). MaxDistance is also measured using the taxicab metric. One might also try $d(x, y) = \max\{|x|, |y|\}$ as a measure of distance (in both cases, the MaxDistance is the distance from the centre to an outside edge). The taxicab metric is preferred over the standard Euclidean distance because it saves on computation time. For example, we can define the following contour probe functions using MaxDistance. Let

$$d_m(x, y) = \max\{|x|, |y|\},$$

where $d_m(x, y)$ is measured from the center of mass of an object (*e.g.*, cup in Fig. 5), and x, y are the coordinates of a point in the intersection of a ray projecting from the center of mass to the edge of an object. Let $d_m(x_1, y_2), d_m(x_1^{'}, y_2^{'})$ denote MaxDistances relative to the prototype image I_1 and sample image I_2, respectively, each with a superimposed fine mesh of squares with uniform size. In what follows, let f_1, f_2 in Figs. 5 and 6 be represented by $diff_{NE_1}, diff_{NE_2}$, respectively. Then we can begin defining a collection of contour probe function as follows.

$$diff_E = |d(x, 0) - d(x^{'}, 0)|, diff_N = |d(0, y) - d(0, y')|,$$

$$diff_{NE_1} = |d_m(x_1, y_1) - d_m(x_2^{'}, y_2^{'})||,$$

$$diff_{NE_2} = |d_m(x_3, y_3) - d_m(x_4^{'}, y_4^{'})|.$$

E (East contour):

$$E\,(I_1, I_2) = \begin{cases} diff_E,\ if\ diff_E < \varepsilon, \\ 0,\ otherwise. \end{cases}$$

N (North contour):

$$N\,(I_1, I_2) = \begin{cases} diff_N,\ if\ diff_N < \varepsilon, \\ 0,\ otherwise. \end{cases}$$

$\mathbf{NE_1}$ (15^o projection from center of mass):

$$NE_1\,(I_1, I_2) = \begin{cases} diff_{NE_1},\ if\ diff_{NE_1} < \varepsilon, \\ 0,\ otherwise. \end{cases}$$

$\mathbf{NE_2}$ (30^o projection from center of mass):

$$NE_2\,(I_1, I_2) = \begin{cases} diff_{NE_2},\ if\ diff_{NE_2} < \varepsilon, \\ 0,\ otherwise. \end{cases}$$

Assume that the distance values of the contour probe functions are normalized relative to the maximum distance from the center of mass to a point intersected by at least one of the projecting rays (*e.g.*, d(x, y) in Fig. 6).

Remark 1: **Euclidean distance**
An alternative to the taxicab distance is the Euclidean distance d between the center of mass (0, 0) and (x, y) on the edge of an object in an image, where $d = \sqrt{(x-0)^2 + (y-0)^2} = \sqrt{x^2 + y^2}$. Let $d_m(x, y) = \max\{d_1, d_2\}$, where d_1, d_2 denote Euclidean distances in images I_1, I_2 that are being compared. This alternative is simpler to use because it is not necessary to superimpose a mesh on I_1, I_2.

B. Sample approximation space

Table I is set up relative to $B = \{N, E\}$ and a set of sample images $U = \{x_0, x_1, \ldots, x_{10}\}$. Let d=1(accept), d=0 (rej3ect), $D = \{x \in U : d(x) = 1\} = \{x_0, x_3, x_4, x_6, x_8\}$ be the decision class, and consider the following equivalence classes:
$[x_0]_B = \{x_0, x_4\}$, $[x_1]_B = \{x_1, x_5, x_9\}$, $[x_2]_B = \{x_2\}$,
$[x_3]_B = \{x_3\}$, $[x_6]_B = \{x_6\}$, $[x_7]_B = \{x_7\}$,
$[x_8]_B = \{x_8, x_{10}\}$.
Then letting $B_*D = \{x_0, x_3, x_4, x_6, x_8\}$ play the role of Y in equation (1) and each equivalence class $[x]_B$ play the role of X in Eq. 1, one obtains:
$\nu([x_0]_B, B_*D) = \frac{2}{5}$, $\nu([x_1]_B, B_*D) = 0$,
$\nu([x_2]_B, B_*D) = 0$ $\nu([x_3]_B, B_*D) = \frac{1}{5}$,
$\nu([x_4]_B, B_*D) = 0$ $\nu([x_5]_B, B_*D) = 0$
$\nu([x_6]_B, B_*D) = \frac{1}{5}$, $\nu([x_7]_B, B_*D) = 0$
$\nu([x_8]_B, B_*D) = \frac{1}{5}$.
In effect, there are 4 neighborhoods containing images that are B-similar to the images in B_*D. More features (*e.g.*, orientation) and corresponding probe functions need to be considered before we can reach a definitive conclusion about the two images.

VII. Conclusion

Although the suggested adjustment in the approach to classifying objects proposed by Zdzisław Pawlak is minor, it is quite important, since it suggests a new research stream in approximation space-based research concerning perception and pattern recognition. From what has been presented concerning the historic, semantic, philosophic and mathematical interpretation of attributes and features, it would seem that the view of attribute-based perception is quite different from feature-based perception provided by approximation spaces, especially if one considers the fact that features are favored in quite a number of disciplines.

Acknowledgements

The author gratefully acknowledges comments and suggestions concerning various parts of this paper by the anonymous referees, Maciej Borkowski, David Gunderson, Christopher Henry, Dan Lockery, Sheela Ramanna, and Andrzej Skowron. This research has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) grant 185986 and Manitoba Hydro grant T277.

References

[1] C.M. Bishop, *Neural Networks and Pattern Recognition*. UK, Oxford University Press, 1995.
[2] M. Borkowski, J.F. Peters, "Matching 2D image segments with genetic algorithms and approximation spaces," *Transactions on Rough Sets*, vol. V, LNCS 4100, 2006, 63-101.
[3] Gersho,A., Gray,R.M.: *Vector Quantization And Signal Compression*. Dordrecht, The Netherlands, Kluwer Academic Publishers, 1992.
[4] I. Guyon, S. Gunn, M. Nikravesh, M. Zadeh (Eds.), *Feature Extraction. Foundations and Applications*. Springer, Heidelberg, Studies in Fuzziness and Soft Computing 207, 2006.
[5] T. Honderich (Ed.), *The Oxford Companion to Philosophy*, 2nd Ed. UK, Oxford University Press, 2005.
[6] B. Jähne, *Digital Image Processing*, 6th Ed. Berlin, Springer, 2005.
[7] S. Johnson, *A Dictionary of the English Language*, 11th Ed. London, Clarke and Sons, 1816.
[8] I. Kant, Critque of Pure Reason, trans. by N.K. Smith. Toronto, Macmillan, 1929.
[9] P.N. Lehner, *Handbook of Ethological Methods*, 2nd Ed. UK, Cambridge University Press, 1996.
[10] J.M. Mendel, K.S., Fu (Eds.), *Adaptive, Learning and Pattern Recognition Systems. Theory and Applications*. London, Academic Press, 1970.
[11] *The Oxford English Dictionary*. London, Oxford University Press, 1933.
[12] E. Orłowska, "Semantics of Vague Concepts," *Applications of Rough Sets*, Institute for Computer Science, Polish Academy of Sciences, Report 469, March 1982.
[13] Pavel, M.: *Fundamentals of Pattern Recognition*, 2nd Edition. NY, Marcel Dekker, Inc., 1993.
[14] Z. Pawlak, A. Skowron, "Rudiments of rough sets," *Information Sciences*, vol. 177, pp. 3-27, 2006.
[15] Z. Pawlak, A. Skowron, "Rough sets: Some extensions," *Information Sciences*, vol. 177, pp. 28-40, 2006.
[16] Z. Pawlak, A. Skowron, "Rough sets and Boolean reasoning," *Information Sciences*, vol. 177, pp. 41-73, 2006.
[17] Z. Pawlak, *Classification of Objects by Means of Attributes*, Institute for Computer Science, Polish Academy of Sciences, Report 429, March 1981.
[18] Z. Pawlak, *Rough Sets*, Institute for Computer Science, Polish Academy of Sciences, Report 431, March 1981.
[19] Z. Pawlak, Rough sets, *International J. Comp. Inform. Science*, vol. 11, 1982), 341-356.
[20] J.F. Peters, A. Skowron, "Zdzisław Pawlak: Life and work," *Transactions on Rough Sets*, vol. V, pp. 1-24, 2006.
[21] J.F. Peters, "Near sets. Special theory about nearness of objects," *Fundamenta Informaticae*, vol. 76, pp. 1-28, 2007.
[22] J.F. Peters, "Rough ethology: Towards a Biologically-Inspired Study of Collective behaviour in Intelligent Systems with Approximation Spaces," *Transactions on Rough Sets*, vol. III, LNCS 3400, pp. 153-174, 2005.
[23] J.F. Peters, C. Henry, "Reinforcement learning with approximation spaces," *Fundamenta Informaticae*, vol. 71, nos. 2-3, pp. 323-349, 2006.
[24] J.F. Peters, A. Skowron, P. Synak, S. Ramanna, "Rough sets and information granulation." In: Bilgic, T., Baets, D., Kaynak, O. (Eds.), Tenth Int. Fuzzy Systems Assoc. World Congress IFSA, Instanbul, Turkey, *Lecture Notes in Artificial Intelligence*, 2715, Physica-Verlag, Heidelberg, pp. 370–377, 2003.
[25] J.F. Peters, S. Ramanna, "Measuring acceptance of intelligent system models." In: M. Gh. Negoita et al. (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, *Lecture Notes in Artificial Intelligence*, 3213, Part I, pp.764-771, 2004.
[26] L. Polkowski, A. Skowron (Eds.), Rough Sets in Knowledge Discovery 2, *Studies in Fuzziness and Soft Computing*, vol. 19. Heidelberg, Springer-Verlag, 1998.
[27] L. Polkowski, *Rough Sets. Mathematical Foundations*. Heidelberg, Springer-Verlag, 2002.
[28] D. Precup, *Temporal Abstraction in Reinforcement Learning*, Ph. D. dissertation, University of Massachusetts Amherst, May 2000.
[29] D.J. Schattschneider, The Taxicab Group, *American Mathematical Monthly*, vol. 91, no. 7, pp. 423-428, 1984.
[30] A. Skowron, J. Stepaniuk, "Generalized approximation spaces." In: Lin, T.Y.,Wildberger, A.M. (Eds.), *Soft Computing, Simulation Councils*, San Diego, pp. 18-21, 1995.
[31] A. Skowron, R. Swiniarski, P. Synak, "Approximation spaces and information granulation," *Transactions on Rough Sets*, vol. III, pp. 175-189, 2005.
[32] J. Słupecki, "Towards a generalized mereology of Leśniewski," Studia Logia, vol. VIII, pp. 131-155, 1958.
[33] J. Stepaniuk, "Approximation spaces, reducts and representatives." In [26], pp. 109-126, 1998.
[34] N. Tinbergen, "On aims and methods of ethology," *Zeitschrift für Tierpsychologie*, vol. 20, pp. 410-433, 1963.
[35] J.D. Ullman, *Principles of Database and Knowledge-Base Systems*, vol. 1. MD, Computer Science Press, 1988.
[36] V.N. Vapnik, *Statistical Learning Theory*. Chichester, UK, Wiley-Interscience, 1998.
[37] S. Watanabe, Pattern Recognition: Human and Mechanical. Chichester, UK, John Wiley & Sons, 1985.
[38] I.H. Witten, E. Frank, *Data Mining*, 2nd Ed. Amsterdam, Elsevier, 2005.
[39] Partial function: `http://en.wikipedia.org/wiki/Partial_function`