A Prototype-Exemplars Hybrid Cognitive Model of "Phenomenon of Typicality" in Categorization: A Case Study in Biological Classification

Francesco Gagliardi (francesco.gagliardi@libero.it)
Department of Philosophical and Epistemological Studies
University of Rome *La Sapienza* — Via Carlo Fea — I-00161 Roma, Italy

Abstract

In this paper we introduce a new machine learning system developed to investigate the phenomenon of typicality in categorization processes of human mind. Experimental researches in cognitive psychology produced a lot of evidences that categorization is a process based on similarity between observed instances and representative instances, which constitutes the concept acquired by human mind. Conversely, the main theories developed till now, prototypes theory and exemplars theory, are unable to fully explain the experimental findings about the "phenomenon of typicality". The classifier algorithm, introduced here, is characterized by an adaptive concepts description based on intermediate abstracted items varying from prototypes to exemplars and so it is able to subsume prototypes and exemplars theories. We test it on a well known biological classification problem. The experimental results show this classifier can be considered as a typicality theory.

Keywords: Artificial Intelligence; Categorization; Prototypes; Exemplars; Typicality; Machine Learning; Classifier Systems; Synthetic Method.

Introduction

Categorization is the adaptive fundamental process by which we "cut" the physical and social reality. It permits us to understand and make predictions about objects and events in our world. (Houdé, 1998) (Kruschkea, 2001) (Medin, Aguilar, 1999). Therefore, categorization is a pervasive cognitive activity by which human mind divides the world in categories by building concepts that are mental representations of these categories.

The understanding of human mind processes of categorization is one of the most important and debated intellectual challenges of the cognitive science and of the artificial intelligence. In fact, categorization is a fundamental process for both human and artificial intelligence. In the following, we illustrate the principal theories of categorization that are developed in cognitive psychology, then we introduce the machine learning as the natural scenario for construction of computational models of categorization and in this framework we propose a new classifier system as a cognitive model of the "typicality phenomenon" in the categorization processes.

Theories of Categorization

The main theories concerning the concepts (Medin, 1989; Murphy, 2002; Thagard, 2005), illustrated in the following, are: the classical theory also known as Aristotelian, the prototypes theory and the exemplars theory.

Classical Theory

According to the classical theory a concept is defined by a set of features which are necessary and sufficient conditions, which are called *defining features*. For example, a geometrical shape is a triangle if and only if it has three sides. Outside the field of Mathematics it is really hard to find concepts which can be defined through necessary and sufficient conditions. According to the classical theory concepts are mentally represented by definitions or better they *are* logical predicates.

Prototypes Theory

The first theory (Rosch, 1975; Rosch e Mervis, 1975), which overcomes many of the problems related to categorization encountered by the classical theory, affirms that concepts are prototypes representing the typical characteristics of objects of a category rather than necessary and sufficient conditions. For example, something is a dog if presents many of the typical characteristics of dogs such as the tail, four legs and others. But it is still a dog even if it lacks some of the typical characteristics. Therefore, according to the *prototypes theory* people tend to identify a category of objects and to reason about its members, by referring to a precise object that is *typical* of the category.

Exemplars Theory

A different point of view consists of considering concepts as a collection of stored exemplars in memory. People increment the number of stored exemplars by observing different objects belonging to the same category, and so they can categorize new objects according to the stored ones. This theory, known as *exemplars theory*, was introduced for the first time by Medin and Schaffer in 1978 (Medin, Schaffer, 1978). It is totally different from the previous ones because rejects the idea, common to the classical and prototypes theories, that people have a kind of representation able of describing the whole category.

For instance, the concept of dog is nor a definition comprising all the dogs, neither a representation of the typical characteristics of dogs, but instead, it is the set of single observed dogs that are remembered by people. In a way, there is not anymore the concept itself (cf. Murphy, 2002; pg. 49), such as it is normally intended: a representation valid for each object of the category.

The "Phenomenon of Typicality"

We can consider both the prototypes and exemplars theories as incomplete and unsatisfactory.

In fact, Gregory Murphy (Murphy, 2002; pg. 4) pointed out the necessity to find new ways of thinking about the categorization problem, rather than to persist in the diatribe prototypes versus exemplars, and therefore, to find a more "inclusive" approach . We think this may consist of an elaboration of new theories which subsume the existent ones, and hence, contain them as special or limits cases. We follow this point of view in the development of machine learning systems introduced in the following sections.

Beyond the limits of single proposed theories, the existence of typicality is without criticisms and can be seen as a "phenomenon" in the categorization processes (cf. *"Typicality as phenomenon"* in (Murphy; 2002 pg. 28)).

This "phenomenon" cannot be fully explained by any existing theory because *"no theory has a ready explanation for all of the findings even within each specific topic"* (Murphy; 2002 pg. 5).

Categorization and Machine Learning

An aspect common to prototypes and exemplar theories is the hypothesis that categories are represented by instances of the classes. In the former case the instances are abstracted from the observations while in the latter instances are the same previously observed. In the field of machine learning (see following sec.) and in particular in the problem of automatic classification, one of the known learning methodologies is the *instances based learning*, for which the learned classes are represented by instances, more or less abstract, of the class itself. Therefore, the machine learning field is the suitable framework where we can link the categorization theories, based on prototypes and exemplars, with the known classifier algorithms. Moreover, in this field it is possible to define new classifier algorithms as computational models of categorization based on typicality that generalize both the theories of prototypes and exemplars.

Machine Learning

Machine learning (Witten, Frank, 2005) (Michie, Spiegelhalter, Taylor, 1994) (Duda, Hart, Stork, 2000) (Russell, Norvig, 2002) is the field of the Artificial Intelligence (AI) concerned with programmes learning from experience and improving their performances (Russel, Norvig 1995; pg.523). In the subfield of *supervised learning*, the system is provided with the experience of a human expert, the teacher, who provides a category *label* for every instance of the data belonging to the training set. The system uses such information to learn categories and to predict the category for new instances.

Supervised Learning Classification

The problem of classification, known also as *pattern recognition*, *discriminant analysis* and *supervised classification*, concerns the construction of classifier systems which can assign to every presented input instance, the proper class among a set of possible classes. This task is carried out in two steps, one of learning or training, and one of predicting, that is properly classification. In the former a set of labelled data, called the *training set*, is considered in order to learn the function which maps the observations in the classes. In the latter phase, data for which the belonging class is unknown are considered and the classifying function, learned during the training phase, is used to predict their classes.

Cognitive Plausibility of Concepts Description

Classification algorithms strongly depend on the kind of classes representation which they infer from data, called *concept description* and which they then use to classify new instances. In fact, we can distinguish classification algorithms by considering the type of classes representation on which they are based. (Witten e Frank, 2005; chap. 3; pg. 61-82) (Michie, Spiegelhalter, Taylor, 1994; chap. 12: pg. 228-245)

Instance-based Learning. In instance-based learning systems the knowledge, extracted from the training set, consists of the storage of directly observed or abstract instances belonging to the set of all possible observations. These instances which are saved in memory form the categories representation. Classification is performed comparing a new instance, for which the class is unknown, with the labelled instances in memory. Also other learning methods are based on instances, in meaning that they learn classes using the instances of the training set, but they do not use instances themselves to represent the classes. Instead, the method called *instances based learning* is a way of learning totally different from others because it uses the instances themselves to represent the learned classes.

The instances based representation, unlike other widely used representations in machine learning (e.g. rules, decision trees, etc.), is the only coherent with both the prototypes and exemplars theories and hence, in accordance with the "typicality view" on categorization, is the one to be used in order to develop classifier systems characterized by cognitive plausibility. Other representations (e.g. classification rules) can be only related to the classical theory of categorization, and therefore, they lack a truly cognitive plausibility.

Instance-based Classifier Systems

Instance based learning is founded on four elements: the notion of similarity between observations, the classes representation, the classification algorithm and the learning algorithm.

In the following we briefly describe them:

- *Similarity* is formalized through a definition of a metric in the space of all possible observations, by which it is possible to quantify the distance between objects and thus also between new instances and the ones stored as representative of the classes.
- *Classes representation* is constituted of a set of couples composed of an instance and the relative class. It is created by the learning algorithm and is used by the classification algorithm.
- *Learning algorithm* uses the training set to construct a set of representative instances. In case of iterative

construction the algorithm can also use the classification function in order to calculate performances and then to improve them.

- *Classification algorithm* assigns a class to each new observation based on a criterion of greater similarity to the representative instances. It uses a given distance and a set of representative instances.

Categorization Theories and Instance-based Classifier Systems

We recall briefly some classifiers, yet known in literature, that can be related to the categorization theories, then we describe a new classifier which we propose as a typicality theory.

Prototypes based Algorithms

The *Nearest Prototype Classifier* is one of the simplest classification systems. (Kuncheva, Bezdek, 1998) (Bezdek et al., 1998). The learning algorithm is based on the construction of a single representative instance for every class. Each of the representative instances is called the *prototype* of the relative class and it is calculated as the barycentre of the instances belonging to that class. The *NPC* assigns to any new observed instance, the class which prototype is the nearest. There exist some other prototypes based classifiers, as for example the *NMPC* (Nearest Multiple Prototypes Classifier) (Bezdek et al., 1998) that we do not report here for the sake of brevity and because their properties are not much interesting considering that in the following we illustrate some hybrid algorithms.

Exemplars based Algorithms

The Nearest Neighbour Classifier (*NNC*). The *Nearest Neighbour Classifier* is the first instance-based classifier defined (Cover, Hart, 1967), it is based on the plain comparison between the new instances and the training set.

The learning phase is *de facto* absent because the set of the representative instances is the entire training set, for this reason this classifier is called *memory based*. The *NNC* assigns to any new instance the class of the closest representative instance. This classifier is very sensible to the learning phase, in fact, in presence of noisy data in the training set, its subsequent performances and generalization ability decrease. It is possible to overcome this drawback by means of the classifier introduced in the next section.

The k-Nearest Neighbour Classifier (*k-NNC*) The *k-NNC* share with the *NNC* the absence of learning phase: it is based on the memorization of the training set; while it differs from the *NNC* during the classification phase.

It considers the *k* exemplars of the training set which are nearest to the new observed instance and it assigns the class with a criterion such the "majority rule" among the *k* considered exemplars.

The *Prototype-Exemplar Learning Classifier* (*PEL-C*)

The algorithm *PEL-C* (*Prototype-Exemplar Learning Classifier*) is a new classifier based on an iterative learning phase that stops when the classes representation satisfies a given condition. Its classification algorithm is the nearest neighbour rule (*NN rule*): to any new observed instance it assigns the class of the nearest instance among the representative ones. Concerning the learning phase, the starting step is, as the *NPC*, the calculation of one prototype for every class, then for any new learning iteration, the misclassified instance in the preceding iteration, which results the farthest from the nearest prototype of its own class (viz. the most atypical one), is added as *candidate instance.* This candidate instance may or may not undergo an abstraction process by a re-calculation of the prototypes positions of its class. This process consists of applying the nearest neighbour rule having considered in the categories representation the insertion of this new candidate instance.

If the abstraction takes place, the considered instance "generates" a new prototype and consequently the old ones adapt; otherwise it is stored as an exemplar of that class in the set of representative instances. In fact, the more atypical is the new candidate instance, or far respect to the other representative instances of its class, the bigger is the possibility that its insertion in the set of the representative instances does not imply an abstraction process.

We present the learning algorithm. We indicate **TS** as the training set, **RI** as the representative instance set and C_k as the items of the class conventionally labelled with *"k"*. The learning procedure is the following:

1. Initialize **RI** with the barycentre of the classes C_k
2. **WHILE** NOT (*Termination Condition*)

[Find a new candidate instance]

2.1 Calculate the distances between every instance of **TS** and every instance of **RI**

2.2 Among the misclassified instances of **TS**, find the instance which is the farthest from the nearest instance of **RI** belonging to its own class. Call it *X* and assume that it belongs to the class C_k

2.3 Add *X* to **RI**.

*[Update **RI**]*

2.4 Consider only instances of **RI** and **TS** belonging to C_k. Call them as $\mathbf{RI_k}$ and $\mathbf{TS_k}$, respectively

2.5 Update the positions of **RI** using the *k-means* clustering algorithm applied only to $\mathbf{TS_k}$ with starting conditions $\mathbf{RI_k}$:

2.5.1 Apply the *NN rule* to the items of $\mathbf{TS_k}$ respect to the $\mathbf{RI_k}$

2.5.2 Iteratively re-calculate the locations of instances of $\mathbf{RI_k}$ by updating the barycenters calculated respect to the subclasses determined with the *NN rule* .

3. **END**

Algorithm 1: Learning Algorithm of *PEL-C*

The behaviour of this classifier system can vary from the one of the *NPC*, completely based on prototypes, to the one of *NNC*, completely based on exemplars, in an adaptive way and according to the chosen termination condition, to the kind of particular categorization problem and to the corresponding considered *data set.*

In intermediate cases the number and the kind of the representative instances is dynamically determined as a combination of prototypes, exemplars and hybrid instances of an intermediate abstraction level. In fact, any new *candidate instance* is inserted in representative item set as an exemplar (see 2.3 of previous algorithm), but in the next phase (see 2.5 of previous algorithm) it can be subjected to an abstraction process, and it can even become a prototype of a new subset of the considered class.

In general, we can think about different possible termination conditions of this classifier learning algorithm, such as the following:

- *Number of iterations*. The number of learning iterations is set and, hence, the number of instances which will constitute the category descriptions. Obviously, this choice is not so useful to cognitive modelling because the number of representative instances of a class is not known *a priori.*
- *Descriptive accuracy*. The accuracy percentage in classification of the training set is fixed and it can be equal to or less than *1*. This can be useful in order to avoid possible overfitting problems. In fact, the maximization of accuracy on the training set, that is, the most detailed description of the classes based on the considered training set, often causes a decreasing of generalization ability of the system on new observations.
  In the case it is set to *1*, the system is forced to classify correctly all the training set, and the obtained classifier is the yet known *T.R.A.C.E.* (acronym of *Total Recognition by Adaptive Classification Experiments* (Nieddu, Patrizi, 2000)). This algorithm is proved to converge in a finite number of iterations to imposed descriptive accuracy, under not restrictive hypotheses, as for example, the absence of contradictory items in the training set.
- *Generalization or predictive accuracy.* The system can estimate its own performance on a new instances by using a technique of *cross validation* (Henery, 1994; pg. 108) (Witten e Frank, 2005; pg. 149) as varying the number of iterations. Therefore, the system is able to find the minimum number of iterations to obtain the maximum capability of generalizing (predictive accuracy on new instances), also if it worsens the descriptive accuracy (on instances of training set). In fact, beyond a certain value of the iteration number, the accuracy on the training set increases while the accuracy on new instances used for validation does not: this is the so called overfitting phase.

This latter termination condition is an adaptive version of the previous one, and it is the one we have chosen because it has no need of *a priori* knowledge as the previous two.

Instance-Based Algorithms and Categorization Theories

The introduced classifiers can be easily related to the theories of categorization shown in preceding section.

The *NPC* classifier builds category descriptions, the concepts, which are linked to prototypes theory, while the classifiers such as the *NNC* and the *k-NNC* are in accordance with the exemplars theory both for the type of memory based representation and classification criterion.

In the framework of instance based classification the *NPC* and the *NNC* can be considered as limit cases because the former performs classification by considering a unique abstract instance for any single class, the *prototype*, while the latter retains all *exemplars* of the training set for the next classification. In a sense the *NPC* and the *NNC* represent respectively, the "radical" versions of the prototypes and exemplars theories.

Hybrid Algorithms

As explained before about the *"typicality phenomenon"*, it is necessary to develop computational models which subsume both the theories, and hence, which reproduce them as particular cases. There exist different instance based algorithms which take their place in an intermediate position between *NPC* and the *NNC*. Some of them are a modified version of the *NPC* for which the number of representative instances is increased, such as the *NMPC* (Nearest Multiple Prototypes Classifier) (Bezdek et al., 1998), and some others are a modified version of the *NNC*, for which the number of stored exemplars are reduced, such as the family of *DROP-n* (Wilson, Martinez, 2000)[1]. Hence, these algorithms are variations of the prototypes based ones and exemplars based ones respectively.

The *PEL-C* algorithm is instead an *hybrid* algorithm because it uses a representation of the classes composed of both prototypes and exemplars (see next sec.). Moreover, this algorithm has the interesting property of presenting as limits cases exactly the *NPC* and *NNC,* being able to vary in all possible intermediate cases (see the following figure).

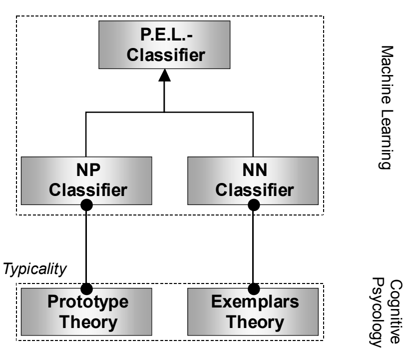

Figure 1: Hierarchy of Classification Algorithms and Cognitive Theories of Categorization

Therefore, this classifier satisfies the need for a more inclusive approach in the categorization study, so it can be considered as a *typicality theory* that subsumes both the prototypes and exemplars theories.

Experimental Results

We show here the comparison of the experimental results obtained applying the different classifier systems,

[1] See (Wilson, Martinez, 2000) also for a review of some other *exemplars based* algorithms.

introduced before, to the problem of categorizing three types of the Iris plant, based on the set of observations introduced by Fisher (Fisher, R. A. 1938) which can be found in the *Machine Learning Repository* of *University of California at Irvine* (Asuncion, Newman, 2007). The Iris classification problem was chosen because it is perhaps the most well known categorization problem (at least in the machine learning literature) and it is probably the most used data set for the comparison among classification algorithms.

This data set contains three classes, each of them represents a different type of the Iris plant (see the following table). Each class is composed of 50 instances. The attributes of every observation are the length and width of both petal and sepal. The classification problem consists of determining the class according to these attributes.

Table 1: Classes of *'Iris'* data set

Class Id.	Class	# Instances	Perc. (%)
1	Iris Setosa	50	33.3%
2	Iris Versicolor	50	33.3%
3	Iris Virginica	50	33.3%
	Total	150	

We carried out different test suites for the different classifier systems: NPC, NNC, k-NNC and PEL-C. Each test suite was prepared by using as cross validation the *leave-one-out* method (Lachenbruch, Mickey, 1968) (Henery, 1994; pg. 108) (Witten e Frank, 2005; pg. 151)), therefore, each test case, belonging to the test suite, is composed of one test instance and of the remaining instances as training set. For the *k-NNC* and *PEL-C* we executed runs, constituted of more test suites as varying *k,* or the stop condition, respectively. In the next table we show, as varying the used classifier, the accuracy (hit ratio), the cardinality of the concepts description (the number of representative instances), the number of pure prototypes and exemplars present in the representative instances and finally the mean percentage of representativeness, inside their own class, of the representative instances.

Table 2: Comparison of Classifier Systems on *'Iris'* data set

	Classifier System	accuracy	# representative items	# pure prototypes	# pure exemplars	mean percentage of representativeness
Prototypes Based	**NPC**	0.927	3	**3**	**0**	100.0%
Hybrid	**PELC**	**0.967**	7	**1**	**1**	42.9%
Exemplars Based	**k-NNC**	**0.967**	149	**0**	**149**	n/a
	NNC	0.953				2.0%

We observe that the *PEL-C* reaches the best classification performance which is as good as the one of *k-NNC*, that has the *k* optimised in order to have the performance maximum. Nevertheless, *PEL-C* obtains it with only *7* instances against the *149* of the *k-NNC*, this behaviour represents a strong "cognitive parsimony" for the same performance. Of the *7* instances used by the *PEL-C* to build the concepts description, *1* is a pure prototype, *1* is a pure exemplar and *5* are representative instances with an intermediate abstraction. The *NPC* uses *3* instances which are pure prototypes, while the *NNC* and *K-NNC* use 149 exemplars (the whole training set). In the following table we show in detail the concepts description found by the *PEL-C* for the Iris data set.

Table 3: The *concepts description* obtained by *PEL-C* on *'Iris'* data set

Representative Items (mm)				Class	Cardinality	Relative Percentage	
Sepal length	Sepal width	Petal length	Petal width				
50.06	34.28	14.62	2.46	Iris Setosa	50	**100%**	←Prototype
62.11	30	45.15	14.30	Iris Versicolor	27	52.94%	chorus of prototypes
54.89	25.16	38	11.47		19	37.25%	
62	24.25	47.25	14.75		5	9.80%	
70	31.59	58.74	21.74	Iris Virginica	27	55.10%	chorus of prototypes
61.59	27.68	52.05	18.59		21	42.86%	
49	25	45	17		**1**	2.04%	← Exemplar

We observe that the representative instances type varies from a pure prototype (the first row) to a pure exemplar (last row), while the concepts vary from a class totally based on a prototype (Iris Setosa) to a class based on a prototypes chorus plus *1* exemplar (Iris Virginica). Thus, *PEL-C* shows how an hybrid representation of categories is fundamental to achieve a more effective categorization process, than a pure representation based only on prototypes or on exemplars, which were proposed by the two main theories of categorization. In fact, this type of representation is obtained dynamically, in a spontaneous way by maximizing the predictive power of the classes representation, as we show in the following plot.

In the next figure 2 we show classification accuracy on training set (line with "+") and on test set composed of new observations (line with "○"), as varying the number of learning iterations and consequently the number of representative instances. The *PEL-C* behaviour corresponds to the maximum of accuracy on test set, achieved with *5* learning iterations and a concepts description, shown before, composed of *7* instances.

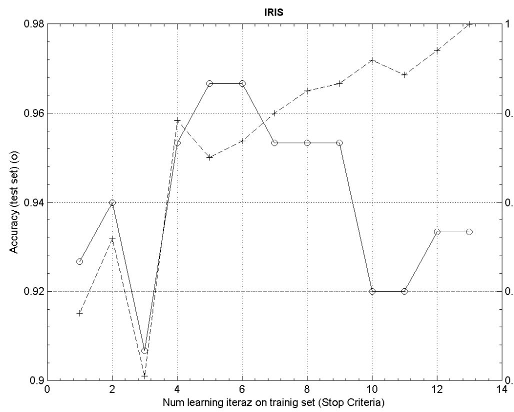

Figure 2: Learning dynamics of *PEL-C* on *'Iris'* data set

Conclusions

Results of the present work are twofold. One is a cross-fertilization between cognitive psychology and machine learning, in particular between the researches about categorization theories and the field of instance based machine learning. The other one, probably the most important, is the realization of a classifier system based on prototypes-exemplars hybrid representations of categories, which models the "phenomenon" of typicality in categorization, capturing *proto*-typical and *a*-typical aspects of it. Therefore, this classifier can be considered as a typicality theory which subsumes both the prototypes and exemplars theories, thus it is also useful to overcome the unfruitful diatribe of prototypes versus exemplars presents in cognitive psychology. However, it will be necessary to compare the previous classifier systems on other categorization problems to confirm further the present findings.

Acknowledgments

I would like to thank Prof G. Trautteur and Prof R. Cordeschi for the valuable discussions about matters illustrated in the present paper. Heartedly thanks are due to Dr. A. Brindisi for her support.

References

Asuncion, A. & Newman, D.J. (2007). *UCI Machine Learning Repository*. Irvine, CA: University of California, School of Information and Computer Science. (http://archive.ics.uci.edu/beta/index.html)

Bezdek, J.C., Reichherzer, T.R., Lim, G.S., Attikiouzel, Y. (1998) *Multiple-prototype classifier design*. IEEE Transactions on Systems, Man and Cybernetics, Part C, 28 (1): 67-79 (http://dx.doi.org/10.1109/5326.661091)

Cover, T. M., Hart, P. E. (1967) *Nearest Pattern Classification* IEEE Trans. on Information Theory, Vol. IT-13 (1): 21-27

Duda R., Hart P., Stork D. (2000) *Pattern Classification (2nd Ed)*. John Wiley & Sons, New York, NY. (ISBN: 978-0-471-05669-0)

Fisher, R. A. (1938). *The statistical utilisation of multiple measurements*. Ann. Eugen., 8: 376–386.

Henery, R. J. (1994) *Methods for Comparison* in (Michie, Spiegelhalter, Taylor, 1994; Chap. 7; pp. 107-124)

Houde, O. (1998) *Categorization*. In Houde, O.,Kayser, D., Koenig, O., Proust, J., Rastier, F. (Eds.) *Vocabulaire de sciences cognitives*. Presses Universitaires de France.

Kruschkea, J.K. (2001) *Categorization and Similarity Models*. In Smelser, N.J., Baltes, P.B. (Eds.) *International Encyclopedia Of The Social & Behavioral Sciences*. Pergamon. pp. 1532-1535 (http://dx.doi.org/10.1016/B0-08-043076-7/00622-7)

Kuncheva, L.I., Bezdek, J.C. (1998) *Nearest prototype classification: clustering, genetic algorithms,or random search?* IEEE Transactions on Systems, Man and Cybernetics, Part C, Volume: 28, Issue: 1, pages: 160-164 (http://dx.doi.org/10.1109/5326.661099)

Lachenbruch, P., Mickey, R. (1968). *Estimation of error rates in discriminant analysis*. Technometrics, 10: 1-11.

Medin, D.L. (1989). *Concepts and conceptual structure*. American Psychologist, 44, 1469-1481.

Medin, D.L., Aguilar, C. (1999) *Categorization*. In Wilson, R.A., Keil, F. (Eds.) *The MIT Encyclopedia of the Cognitive Sciences (MITECS)*. MIT Press, pp. 104-106. (ISBN 0-262-23200-6)

Medin, D. L., Schaffer, M. M. (1978). *Context theory of classification learning*. Psychological Review, 85, 207–238.

Michie, D., Spiegelhalter, D.J., Taylor, C.C. (Eds) (1994) *Machine Learning, Neural and Statistical Classification*. Prentice Hall, (Ellis Horwood Series in Artificial Intelligence) (ISBN 013106360X) (Url: http://www.maths.leeds.ac.uk/~charles/statlog/)

Murphy, G. L. (2002). *The big book of concepts*. Cambridge, MA: MIT Press.

Nieddu L, Patrizi G. (2000) *Formal methods in pattern recognition: A review*. European Journal of Operational Research 120: 459-495

Rosch, E. (1975). *Cognitive Representations of Semantic Categories*. Journal of Experimental Psychology, 104(3): 192-233

Rosch, E., Mervis, C.B. (1975). *Family resemblance: Studies in the internal structure of categories*. Cognitive Psychology, 7, pp. 573-605

Russell, S.J., Norvig, P. (1995). *Artificial Intelligence. A Modern Approach*. Prentice Hall, Englewood Cliffs, NJ (ISBN 0-13-103805-2)

Russell S.J., Norvig P. (2002) *Artificial Intelligence. A Modern Approach (2nd Ed.)* Prentice Hall, Englewood Cliffs, NJ

Thagard, P. (2005). *Mind: Introduction to cognitive science (2nd ed.)*. Cambridge, MA: MIT Press.

Wilson, D.R., Martinez, T.R. (2000). *Reduction techniques for exemplar-based learning algorithms*. Machine Learning, 38(3), pp. 257-268. (http://citeseer.ist.psu.edu/wilson00reduction.html)

Witten, I.H., Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. *(2nd Ed.)* Morgan Kaufmann, San Francisco, CA.