

The Role of Prototypicality in Exemplar-Based Learning

Yoram Biberman

Department of Mathematics and Computer Science
Ben-Gurion University of the Negev
P.O.B. 653, 84105 Beer-Sheva, Israel
e-mail: yoramb@cs.bgu.ac.il, phone: 972-2-437-244, fax: 972-7-472909

keywords: Concept learning, Exemplar-Based Learning, Prototypes.

Abstract. This paper examines the role of prototypicality in exemplar-based concept learning methods. It proposes two approaches to prototypicality: a shared-properties approach, and a similarity-based approach, and suggests measures that implement the different approaches. The proposed measures are tested in a set of experiments. The results of the experiments show that prototypicality serves as a good storing filter in storage reduction algorithms; combining it in algorithms that store all the training set does not improve significantly the accuracy of the algorithm. Finally, prototypicality is a useful notion only in a subset of the domains; a preliminary examination of those domains and their characteristics is proposed.

1 Background and Motivation

Human concept learning is studied intensively for about seventy years. In its early days the research concentrated about *logical concepts* defined by a set of conditions. Logical concepts have the property that any given object either belongs to the concept or not, and all members of a concept represent it equally, or belong to it to the same extent (e.g., ‘prime number’, as each number is either prime or not, and no number is ‘more prime’ than another; ‘grandmother’ can be defined as the mother of a parent, and any grandmother is as good exemplar of this concept as any other). Logical concepts are thus *unstructured*.

During the last three decades it is argued that *natural concepts*, occurring in everyday life, are totally different from logical concepts, cannot be defined by a set of necessary and sufficient conditions, and are structured. The first to propose this idea was Wittgenstein [18] who asked “how would you define the concept of *game*?” He claimed that this concept contains a large variety of objects; each game is similar to some other games in some of its properties, but there are no properties common to all of them. Wittgenstein concluded that natural concepts, like games, can not be defined by rules. He suggested that members of a natural concept share a *family resemblance*: each member of the concept is similar to *few* other members of its concept, or shares *few* properties

with *some* other members of its concept, the member need not be similar to *all* other members of its concept.

In line with this argument, Rosch and Mervis [15] claim that natural concepts are represented in our mind by *an image* of the *prototype* of the concept. The prototype is an object that belongs to the concept, and represents it best.

Rosch further argues that prototypicality is a graded property, i.e., members of a concept can be rated according to the extent that they represent the concept. Thus, Rosch suggests that natural concepts are *structured*: different members have different status in them, and prototypicality may serve as a measure of membership in a concept. She writes:

“The basic hypothesis was that members of a category come to be viewed as prototypical of the category as a whole in proportion to the extent that they bear a family resemblance to (have attributes which overlap those of other members of) the category. Conversely, items viewed prototypical of one category will be those with least family resemblance to or membership in other categories. In natural categories of concrete objects, the two aspects of family resemblance should coincide rather than conflict.” ([15], p. 575)

Thus Rosch suggests that the rated prototypicality of an item is affected by two factors: The number of properties it shares with other members of its concept, and the number of properties it shares with members of contrasting concepts. We term the first factor *focality*, and the second *peripherality*. Rosch argues that in a natural concept the two measures correlate.

The notion of prototypicality is now considered central to theories of human categorization. Many findings concerning phenomenon about human concept learning are explained in terms of prototypicality. For example, Rosch showed that children first learn to classify prototypical members of a category, their ability to classify peripheral members develops only later [14]. Posner and Keele [11] found that more prototypical members of a category are classified faster and more accurately, and are better remembered.

In modern machine learning prototypicality does not have such a central status. Rendell [13] adopts a classic definition of prototypicality, he proposes that *the* prototype of a concept is the centroid of a ball in an Euclidean space. In STAGGER [17] the prototypicality of an object is the extent that it satisfies the necessary and sufficient conditions for membership in the concept. In ID3 [12] the prototypicality of an example may be obtained by considering its depth as a leaf of a decision tree. In AQ15 [8] one can infer the prototypicality of an object from the rule that captures it (if a concept is defined by a set of rules, then a rule that covers more members of the concept is considered more prototypical). A similar approach is held by Bergadano et al. [4] in the POSEIDON system, where they suggest that a *Base Concept Representation* should capture the prototypical members of each concept; the exceptional cases would be identified by an *Inferential Concept Interpretation* process. STAGGER, AQ15, POSEIDON and ID3 relate to prototypicality but do not rely on it: the prototypicality of the objects can be inferred but is not used in the learning or classification process.

In PROTOS [10], EACH [16] and PEBLS [6] the prototypicality of an object O refers to the system's 'confidence' in O as a reliable representative of its class. O is considered a good representative of its concept to the extent that if the system uses O in order to classify unseen objects (using a nearest neighbour, or similar, classification method) then the system predictions would be relatively accurate. The weight of an exemplar E in EACH and PEBLS, which can be seen as its prototypicality, is the quotient between number of objects E correctly classifies (using a nearest neighbour classification rule), and the total number of objects E classifies. A similar approach is held by PROTOS.

As prototypicality bears so much importance in human categorization we might expect that it would also be useful in machine learning. Therefore in the sequel I propose and examine two computational models of prototypicality. The examination shows that while some usages of prototypicality in exemplar-based concept learning improves the accuracy of the learning algorithm other usages degrade the accuracy.

The next section presents the intuitions that led to the two prototypicality measures that are presented in sections 3 and 4. Sections 5.2 and 5.3 presents experiments that were conducted with the proposed measures. The measures were combined in algorithms that store all the training set (Sec. 5.2), and in these algorithms they did not cause an improvement in the accuracy of the algorithm; they were also embedded in storage-reduction algorithms (Sec. 5.3), and in these algorithms they served as a successful storage filters: When the more prototypical items from each concept are stored the accuracy of the algorithm is better than when other criterions for storing exemplars are applied.

2 Intuition

Following Rosch I suggest that an object is considered *prototypical* in a concept if it is *focal* and not *peripheral*; where focality relates to the extent that the object represents the concept under discussion, and peripherality describes the extent that it represents other concepts.

Two alternative approaches to prototypicality are examined. The first is termed *the shared-properties approach*; it is derived from Rosch's theory. Rosch suggests that the focality of an object is determined by the number of properties it shares with other members of its concept; thus if a concept C_1 contains seven members, which in some feature f have the following values $\{a, a, a, a, b, b, b\}$ then the focality of the value a in C_1 is greater than that of b . Consider now another concept C_2 with the following set of values in f : $\{a, a, a, a, x, y, z\}$; I suggest that while in C_2 a is the *only* 'representative' value, in C_1 a is the *best* representative, but there is another value that also serves as a good representative of the category; therefore, the focality of a in C_2 is greater than its focality in C_1 . Thus, the focality of a value v in a category C is determined by two factor: (a) the prevalence of v in C , (b) the prevalence of other values $u \neq v$ in C .

Another, more subtle, point in Rosch's theory that was adopted by the shared-properties based prototypicality measure is that prototypicality is not

a gestalt property of the object as a whole, but may be expressed as a function of its distinct features. Therefore, the focality of an object is defined as the the sum of the focality of its features.

The peripherality of an object is defined as its focality among members of contrasting concepts; the prototypicality of an object is the difference between its focality and its peripherality.

Neumann [9] proposed a concept acquisition model that argue that humans, during the learning phase, count the frequency of each feature in each concept (e.g., red occurs half of the times among members of C). During the test phase, probe items that are described by more frequent features are classified more easily. Thus Neumann's model also rels on feature counting processes.

An alternative approach to prototypicality, one that emerges from Wittgenstein's ideas, relies more directly on similarity evaluations. I denote this approach *the similarity-based approach*. We may suggest that the focality of an object should be defined as its average similarity to other members of its concept, the peripherality of an object is its similarity to members of contrasting concepts, and, like in the previous approach, the prototypicality of an object is the difference between its focality and its peripherality. Two variants of this approach are also tested.

The two approaches are not independent, as if an object shares many properties with other members of its concept, it is, generally, also similar to many other object as well; yet there are also differences between the two definitions, for example an object might be highly similar to few other objects, but not share many properties with many other members of its concept.

The next section presents the shared-properties approach to prototypicality, the section afterward presents the alternative, similarity-based approach to prototypicality.

3 The Shared-Properties Approach to Prototypicality

During the learning process the teachers supply examples of different concepts (assume that a positive example of a certain concept is a negative example for the other concepts). Let $\{E_1, \dots, E_m\}$ be a set of m examples that represent a concept C . Each example $E_i = (v_{i1}, \dots, v_{in})$ is a vector of n values, where each value represents a feature, and is taken from a nominal scale domain. For each E_i in C we define its focality, peripherality, prototypicality.

Denote by C_f the set of values in the feature f occurring in members of the concept C . Denote by $num(v, C_f)$ the number of examples that belong to C and share the value v in the feature f .

3.1 The Focality Measure

The focality of an example in a concept reflects the extent that this example is a good representative of the concept as it shares many features with other members of the concept. The focality of an example E in a concept C is a sum of

the focalities of its features, $Foc_{ex}(E, C) = \sum_{j=1}^n foc_{val}(v_j, C)$; (the same holds for the other two measures as well, therefore in the sequel we concentrate upon focality/peripherality/prototypicality of a value. In most cases C is understood from the context, and is, therefore, omitted from the notation).

The focality of a value v in a feature f in a concept C reflects the extent that v represents C_f ; it is influenced by two factors: (a) the extent that v is frequent in C : (b) the extent that other values represent C , or are frequent in C :

To combine the above two factors we may define the focality of v in C_f to be:

$$foc_{val}(v, C_f) = num^2(v, C_f) - \sum_{v_i \neq v} num^2(v_i, C_f) .$$

The first factor in the above definition reflects the prevalence of v in C , the second factor reflects the prevalence of other values. To normalize the above measure to the range [0..1] we actually define it to be:

$$foc_{val}(v, C_f) = \frac{num^2(v, C_f) - \sum_{v_i \neq v} num^2(v_i, C_f)}{2 \sum_{v_j} num^2(v_j, C_f)} + 0.5 .$$

For example, the focality of b in $\{b, b, b, b, a, a, a\}$ is $(4^2 - 3^2)/(2 \cdot (4^2 + 3^2)) + 0.5 = .64$, while the focality of b in $\{b, b, b, b, x, y, z\}$ is $(4^2 - (1^2 + 1^2 + 1^2))/(2 \cdot (4^2 + 1^2 + 1^2 + 1^2)) + 0.5 = .84$.

The focality measure is a continuous necessity condition: A normal necessity condition is Boolean— an object either satisfies the condition or not, and if it does not then it is not considered member of the concept. Above it was argued that Boolean conditions are not suited for natural concepts that lack sharp boundaries; here the Boolean condition is relaxed to a continuous measure: the measure gets the extreme values 0 and 1 when a normal necessity condition can be applied, i.e. when there is a single value that can serve as a necessity membership condition; the measure gets intermediate values in cases where some values are more prevalent than others, in these cases the measure is larger for the more prevalent values; for example: $foc_{val}(a, \{a, a, a, b\}) > foc_{val}(a, \{a, a, b, b\})$ as in the left concept a is ‘a more necessary value for membership in the concept’ in the sense that there are less non- a members of this concept, which means that in order to be a member of the concept, an example should have an a in this feature.

3.2 The Peripherality and Prototypicality Measures

The peripherality of a value v in C_f reflects the extent that v is frequent among examples of other concepts: the more frequent v is outside C , the more peripheral it is in C . Denote by \overline{C} the set of examples that do not belong to the concept C . The peripherality is dual to the focality and it is defined to be the focality in \overline{C} :

$$per_{val}(v, C_f) = foc_{val}(v, \overline{C}_f).$$

The prototypicality of a value v in C_f reflects the extent that v is frequent among members of C and is infrequent in \overline{C} , while taking into consideration the extent that other values have this property; in other words, the prototypicality of v is the extent that v and only v characterizes C and only C . The prototypicality is define to be:

$$proto_{\text{val}}(v, C_f) = foc_{\text{val}}(v, C_f) - per_{\text{val}}(v, C_f).$$

Both the focality and peripherality measures are in the range $[0..1]$, therefore the prototypicality measure is in the range $[-1..1]$. A larger value in each measure evidences that the example has more of this property; that is, if the focality of an example E_1 is larger than the focality of E_2 in a concept C , then it evidences that the properties (or features) of E_1 are more prevalent in C then the properties of E_2 ; if the prototypicality of E_1 is larger than that of E_2 it indicates that E_1 is a better representative of C as its properties are more prevalent in C , and less prevalent in concepts contrasting with C . If the prototypicality of E is less than zero it indicates that E represents \overline{C} better than it represents C .

This section presented a prototypicality measure that originates from the shared-properties approach. The next section presents two alternative prototypicality measures that originate from the similarity based approach.

4 The Similarity-Based Approach to Prototypicality

The similarity-based approach to prototypicality defines prototypicality by means of similarity. Two variants of this approach are tested. The first variant defines the focality of an exemplar E that belongs to a concept C as the average similarity between E and *all* other members of C , the peripherality of E is defined as the average similarity between E and *all* members of \overline{C} . Formally expressed, let E_1, \dots, E_m be the members of C , E'_1, \dots, E'_k be the exemplars that do not belong to C ; denote by $Sim(E_j, E_j)$ the similarity between the examples E_i, E_j . The focality of E in C is:

$$Foc_{\text{ex}}(E, C) = \sum_{i=1}^m Sim(E, E_i)/m \quad .$$

The peripherality E in C is defined as:

$$Per_{\text{ex}}(E, C) = \sum_{i=1}^k Sim(E, E'_i)/k \quad .$$

The second variant of the similarity-based approach defines the focality of E as the average similarity between E and the three members of C that are most similar to it; peripherality is, accordingly, defined as the average similarity between E and the three objects in \overline{C} that are most similar to E . In both variants the prototypicality of E is the difference between its focality and peripherality.

Two considerations led me to try the second variant as well: (a) If a concept is composed of two or more homogeneous and distinct clusters that are dissimilar

one from the other (e.g., in a xor like configuration) then the second definition might be more successful than the first one. (b) In the context of exemplar-based learning, prototypicality generally serves as a ‘utility measure’ over the exemplars, that is, it should evidence how useful and accurate an exemplar is as a classifier of probe items in the context of a k nearest neighbour, or similar, classification method. For this usage, similarity to items that are distant from E is not important, only the near neighbours of E : whether they belong to C or to contrasting concepts is relevant.

The rest of the paper presents the experiments that were conducted with the proposed prototypicality measures, and discusses their results.

5 Experimental Results and Discussion

Two main sets of experiments or comparisons were performed with the proposed prototypicality measures: The first set involved exemplar-based learning methods that store, and use during in the test phase, all the training set, the second set of experiments involves storage reduction algorithms, i.e., algorithms that store, and use during the test phase, only a subset of the training examples. It turns out that: (a) combining prototypicality measures in algorithms of the first kind does not improve their accuracy; (b) prototypicality measures serve as successful ‘storing filters’ [7] in storage reduction methods, that is, by storing the more prototypical members of each concept, the algorithm would achieve best results.

I shall first describe the negative results, which I find no less interesting. I would like to suggest that these negative results raise substantial questions concerning the role of prototypicality (at least) in computerized learning systems, and concerning the relation between human categorization and computerized models.

5.1 The Examples that were Used in the Experiments

This section overviews the domains that were used in the experiments, and their main properties.

The examples that were used were obtained from the repository of machine learning databases cited in the University of California, Irvine (UCI). As this paper concentrates upon examples that are described by nominal domain features only databases that satisfy this condition are used. On the other hand almost all nominal valued databases are examined; thus we may argue that the set of domains that is used is representative of the set of domains used in machine learning. Figure 1 presents a statistical overview of the examples.

Three of the databases that are used in the following experiments are good examples of domains that contain natural concepts: (a) The ‘Zoo’ domain contains different kinds of animals. (b) The ‘LED display’ example can be described as an ‘artificial natural concept’: It is artificial on the one hand, as it is produced by a computer program, but it has many characteristics of natural concepts on the other hand; it also resembles the kind of concepts psychologists use in their

laboratory experiments that aim to investigate human categorization in natural domains. (c) ‘Hayes-Roth and Hayes-Roth (1977)’ is an example of a database that was borrowed from such experiments. The characterization of other concepts is less obvious, at least for someone who is not expert in these fields.

The domains that were used in the experiments

Missing values
 Frequency of most frequent concept (%)
 Number of concepts
 Number of values in attribute
 Number of predicting attributes
 Number of examples

Domain #	Domain	Missing values	Frequency of most frequent concept (%)	Number of concepts	Number of values in attribute	Number of predicting attributes	Number of examples
1	1984 U.S. Congressional voting	435	16	2	2	61	y
2	LED display	500	7	2	10	10	n
3	LED display + 17 irrelevant attributes	500	24	2	10	10	n
4	Tic-Tac-Toe endgame	958	9	3	2	65	n
5	Standardized audiology	226	69	2	24	48	y
6	Lung cancer data	32	56	3	3	41	y
7	E. coli promoter gene sequences (DNA)	106	57	4	2	50	n
8	Primate splice-junction gene sequences (DNA)	1200	60	4	3	50	n
9	Zoo	101	16	2	7	41	n
10	Hayes-Roth & Hayes-Roth (1977)	160	4	4	3	41	n

Fig. 1. A statistic overview of the examples.

5.2 Prototypicality in algorithms that store all the training set

The proposed prototypicality measures were added to a basic Nearest Neighbour (NN) classifier that bases its predictions over a single exemplar—the one that is most similar to the probe, and to a classifier that bases its predictions over a set of three exemplars. The following subsections presents these experiments, and their results.

1-NN Classifiers The most common, and probably basic, exemplar-based learning method is the IB1 algorithm [2] that bases its classification of a probe over its similarity to a single exemplar—the one that is most similar to the probe, i.e. its Nearest Neighbour (NN). This algorithm works as follows:

- *During the learning phase:* Store all the exemplars the teacher supplies as they are without any processing.

- *During the test phase:* In order to classify a probe object P — Evaluate the similarity between P and all the stored exemplars, and predict that P belongs to the same concept as the exemplar that is most similar to it.

We could expect that if the algorithm in addition to evaluating the similarity between P and each stored exemplar E_i , would also consider the prototypicality of E_i in its concept, it would be more accurate. Therefore we may propose the following Proto-IB1 algorithm:

- *During the learning phase:* Store all the exemplars the teacher supplies. For each exemplar compute its prototypicality.
- *During the test phase:* In order to classify a probe object P — For each stored exemplar E_i , compute $Sim(P, E_i) \cdot Proto(E_i)$ (where $Sim(P, E_i)$ denotes the similarity between P and E_i , and $Proto(E_i)$ denotes the prototypicality of E_i in its concept), and predict that P belongs to the same concept as the exemplar for which this term is maximal.

A comparison between these two algorithms is depicted in Fig. 2. The indices in first line of the figure relates to the domains from Fig. 1. The second line presents the accuracy of a basic IB1 algorithm. (All the algorithms that are examined in this experiment utilize the Context-Similarity measure [5].) The third line presents the classification rate of a Proto-IB1 algorithm that uses the shared-properties based definition of prototypicality. The last two lines depict the accuracies of variants of Proto-IB1 that use the similarity-based definition of prototypicality: In the fourth line prototypicality of an exemplar E that belongs to a concept C is defined as the average similarity between E and its three nearest neighbours in C , versus its three nearest neighbours outside C . In the fifth line the prototypicality of E is defined as the average similarity to all other members of its concept, versus to all the objects that belong to contrasting concepts. Each entry in the table is a result of averaging over 50 runs.

Few conclusions can be drawn from a comparison between the accuracies of the four classifiers. Probably the most notable one is that in seven out of the ten domains that were examined, the basic IB1 algorithm performs best. In three domains the Proto-IB1 algorithms were more successful, two out of these domains are the artificial examples: ‘LED display’ and ‘Hayes Roth & Hayes Roth (1977)’. These domains were proposed by researchers that synthesized them on the basis of their assumptions concerning the structure of prevalent natural domains. (cf. [1,3] concerning performing experiments, and synthesizing artificial domains.) It should be said that the assumptions their inventors held are widely accepted among researchers in cognitive psychology. In both these domains concepts are composed of ‘ideal’ prototypes and other examples that are distortions of the prototypes. The finding that in them the prototypicality-based algorithms perform well evidence that the proposed definitions of prototypicality are valid— if the concepts are truly composed of ideal prototypes and other objects with graded degrees of prototypicality then using the proposed prototypicality measures improves the accuracy. The finding that even in these two domains the accuracy of the three different Proto-IB1 algorithms is not always similar evidences that

**Average classification accuracy
of 1-NN algorithms**

Domain #	1	2	3	4	5	6	7	8	9	10
IB1	93	60	64	57	81	44	77	78	96	95
Proto-IB1 (shared properties)	89	70	62	49	69	48	57	69	94	94
Proto-IB1 (similarity to 3 vs. 3 objects)	92	54	60	49	81	51	74	78	94	96
Proto-IB1 (similarity to all objects)	89	61	52	46	23	49	50	48	88	25

Fig. 2. The accuracy of four classifiers that base their classification on a single exemplar. ‘IB1’ is the basic nearest neighbour algorithm, ‘Proto-IB1 (shared properties)’ refer to the variant of Proto-IB1 that utilizes the shared-properties based definition of prototypicality, ‘Proto-IB1 (similarity to 3 vs. 3 objects)’ refer to a variant of Proto-IB1 that utilizes a similarity-based definition of prototypicality, but examines the similarity to only three other objects within the exemplar’s concept versus to three other objects from contrasting concepts, ‘Proto-IB1 (similarity to all objects)’ is the variant that examines the similarity to all other objects when evaluating the prototypicality of an object.

even in domains that are composed of structured concepts (that are composed of prototypes), the way to capture this structure may differ. In this experiment the most successful prototypicality measure is the one that bases its evaluation over similarity to ‘3 vs. 3’ objects. The fact that this definition is far more successful than the similarity-based definition that relies over similarity to all the exemplar hints that most of the concepts in our domains are not composed of a single distinct cluster. Finally, and what I find as the most interesting result, the fact that in seven out of the eight real life domains the prototypicality-based classifiers were less accurate than the basic IB1 algorithm raise a question whether the common assumption that natural concepts are structured and contain prototypes is valid, or, if we rephrase the question in a more positive manner: what is the role of prototypes and prototypicality in natural domains? Do they have different role in human learning than in computerized one?

The Family Resemblance Measure and 3-NN Classifiers In a slightly different experiment I compared algorithms that base their classification of a probe item not over its similarity to a single exemplar, but over its three nearest neighbours. The basic 3-NN classifier works as follows:

- *During the learning phase:* Store all the exemplars the teacher supplies as they are.
- *During the test phase:* In order to classify a probe object P — Evaluate the similarity between P and all the stored exemplars. Let S be the set of three exemplars that are most similar to P . If at least two members of S belong to the same concepts C , then predict that P also belongs to this concept, otherwise predict the P belongs to the same concept as its nearest neighbour.

The basic 3-NN algorithm is compared with a version that bases its predictions over three exemplars from each concept, and also weights the prototypicality of the stored exemplars. The measure that is used is termed *family resemblance* (*fare*). The family resemblance between a probe object P and a concept C that is represented by the exemplars $\{E_1, \dots, E_m\}$ is define as follows

$$fare(P, C) = \sum_{i=1}^3 Sim(P, E'_i) \cdot Proto(E'_i) / 3 ,$$

where E'_i belongs to C , and the summation goes over the three exemplars in C that maximize $Sim(P, E'_i) \cdot Proto(E'_i)$.

The *fare* based algorithm that is examined is termed a *family resemblance classifier*. It operates as follows:

- *During the learning phase:* Store all the exemplars the teacher supplies. For each exemplar compute its prototypicality.
- *During the test phase:*
 In order to classify a probe object P do:
 For each concept C_i compute $fare(P, C_i)$.
 Predict that P belongs to the concept for which $fare(P, C_i)$ is maximal.

The results of the comparison between the basic 3-NN algorithm and the family resemblance classifiers are depicted in Fig. 3.

Average classification accuracy of 3-NN algorithms

Domain #	1	2	3	4	5	6	7	8	9	10
3-NN classifier	94	64	71	56	81	49	76	78	93	96
fare-3 (shared properties)	90	69	65	49	50	49	55	69	93	96
fare-3 (similarity to 3 vs. 3 objects)	92	59	65	48	81	52	74	78	95	89

Fig. 3. The accuracy of three classifiers that base their classification on three exemplars. ‘3-NN classifier’ is the basic three-nearest-neighbour algorithm, ‘fare-3 (shared properties)’ refer to the family resemblance classifier that utilizes the shared-properties based definition of prototypicality, ‘fare-3 (similarity to 3 vs. 3 objects)’ refer to a variant of the family resemblance classifier that utilizes the similarity-based definition of prototypicality.

The first finding that emerges from Fig. 3, and that is consistent with the previous experiment, is that in four domains the basic 3-NN algorithm is more accurate than the prototypicality-based methods, while in three domains the prototypicality-based algorithms (as a group) are most successful.

As can be seen from Fig. 3, again, the prototypicality-based algorithms are successful on the two artificial domains: ‘LED display’ and the ‘Hayes-roth &

Hayes-Roth (1977)'. A third domain in which these algorithms are accurate is the 'Zoo' domain. Thus we may say that the prototypicality-based classifiers are successful in domains that are composed of structured concepts, that contain more and less prototypical members. But, from the results of the two experiments we may also conclude that many concepts do not have this desired(?) structure. A clear example is the 'Tic-Tac-Toe endgame' domain. Each example in this domain represents a possible board configuration at the end of a 'Tic-Tac-Toe' game. The examples are divided into two concepts: 'a win for player \times ' or not. This domain is not structured, and contains no prototypes: there is no 'prototypical winning configuration', each 'winning configuration' is as good as any other one, by changing a single attribute we may turn each 'winning configuration' into a non winning one, that is we may move the example from one concept to the other. Other domains are more difficult to analyze: For example in the '1984 U.S. congressional voting' domain we may wonder are there prototypical 'Democrat members'? Can we infer from the results of the experiment that the answer to this question is negative, or maybe we have to invoke another definition of prototypicality in order to reveal the structure that does exist in this domain? Databases #5 and #6 represent diseases, while domains #7 and #8 are taken from the molecular biology, we may prematurely speculate that in the first area prototypicality-based methods are relatively successful, while in the latter they are not. A more cautious conclusion would be that different domains have different properties, that could be used by a learning algorithm, if and when these properties would be better explored.

5.3 Prototypicality in Storage-Reduction algorithms

In the previous section prototypicality was used as a kind of weighting over the exemplars in algorithms that store, and use during the test phase, all the training set. There are situations in which we need to store only a subset of the training set; in such situations the algorithm needs to decide which exemplars would be stored, and which would be discarded. Different authors have proposed different storing criterions ([2,19]). Here I would like to suggest the following storing criterion: From each concept C , store the $f(C)$ most prototypical exemplars (where $f(C)$ is determined by some constraints).

The proposed storing criterion was tested, by comparing it with three other storage-reduction algorithms: IB2, IB3 [2], and TIBL [19]. The comparison was done in the following way: The IB2 and TIBL algorithms were executed on a given training set and test set; the number of exemplars they stored from each concept, and their accuracy was recorded; then the prototypicality-based algorithm was ran on the same dataset, the algorithm stored from each concept the same number of exemplars as IB2 (or as TIBL) did, and its accuracy was measured. All the algorithms used during the test phase the same classification method: the nearest-neighbour, and stored the same number of exemplars, they differ in their storing criterion.

The results of the comparisons are depicted in Fig.4, 5.

**Average classification accuracy
IB2, IB3, Proto-IB2**

Domain #	1	2	3	4	5	6	7	8	9	10
IB2	.66	.55	.42	.53	.65	.39	.68	.63	.92	.68
IB3	.90	.58	.41	.65	.69	.40	.69	.65	.90	.75
Proto-IB2	.84	.67	.68	.47	.77	.42	.76	.77	.90	.96

Fig. 4. The accuracy of three storage reduction algorithms: IB2, IB3 and Proto-IB2. The latter stores from each concept the same amount of exemplars as IB2 does, but picks the most prototypical objects.

**Average classification accuracy
TIBL, Proto-TIBL**

Domain #	1	2	3	4	5	6	7	8	9	10
TIBL	.88	.44	.26	.50	.70	.29	.50	.57	.87	.89
Proto-TIBL	.59	.59	.65	.47	.81	.46	.73	.77	.91	.96

Fig. 5.

If we compare the prototypicality-based algorithm with IB2, IB3, TIBL we notice that it is more accurate than IB2 in nine out of the ten domains, more accurate than IB3 in seven domains (but it also generally stores more exemplars than IB3), and more accurate than TIBL in eight domains.

Looking at the domains in which Proto-IB2 is less accurate reveals that in the ‘1984 U.S. congressional voting’ domain, and in the ‘Tic-Tac-toe endgame’ one Proto-IB2 and Proto-TIBL are less successful than IB3 and TIBL. The latter domain was discussed earlier, and the failure of the prototypicality-based algorithm in it was expected, and is understood; concerning the former domain, it is a wonder to me why on this domain prototypicality fails; I wonder whether an expert in the American political system could foresee this result; and more generally, should it be possible for an expert in a field to foresee whether a prototypicality-based algorithm would be successful on a given domain.

The results clearly evidence that storing the more prototypical exemplars is a relatively successful storing criterion. It is ‘relatively’ successful in the sense that it truly produces better results than other storing filters; yet, the accuracies of Proto-IB2 or Proto-TIBL are generally inferior to those of the basic IB1 algorithm; thus, at least theoretically, it is possible to achieve even better accuracies than those achieved by Proto-IB2, Proto-TIBL—If the storage reduction algorithm would store all the exemplars that are used for the classification of probe objects by IB1 then it would achieve the same accuracy as IB1 does.

6 Conclusion

This paper examines the role of prototypicality in exemplar-based concept learning methods. It proposes two approaches to prototypicality, and suggests measures that implement the different approaches. The proposed measures were

tested in a set of experiments. The results of the experiments show that prototypicality serves as a good storing filter in storage reduction algorithms, combining it in algorithms that store all the training set does not cause a meaningful improvement in the accuracy of the algorithm. It was also found that the relative accuracy of an algorithm that utilizes prototypicality varies across domains: while in some domains (e.g. the ‘LED display’ one) it improves the performance, in other domains (e.g., the ‘Tic-Tac-Toe endgame’) it even degrades the accuracy. While in some of the domains this result is understood, in other its reasons are not clear enough and require further examination. Future research should also address questions like: how can a prototypicality measure be combined in a learning algorithm, what is the similarity and difference between the way humans utilize prototypicality versus the manner computers (should) do it.

References

1. D. W. Aha. Generalizing from case studies: A case study. In *Proc. of the 8th int. workshop on machine learning*, pages 1–10. Morgan Kaufmann, 1991.
2. D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.
3. J. R. Anderson and M. Matessa. Explorations of an incremental, Bayesian algorithm for categorization. *Machine Learning*, 9:275–308, 1992.
4. F. Bergadano, S. Matwin, R. S. Michalski, and J. Zhang. Learning two-tiered descriptions of flexible concepts: The POSEIDON system. *Machine Learning*, 8:5–43, 1992.
5. Y. Biberman. A context similarity measure. In *Proc. of the 7th European Conf. on Machine Learning*, pages 49–63. Springer-Verlag, 1994.
6. S. Cost and S. Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 10:57–78, 1993.
7. S. Markovitch and P. D. Scott. Information filters and their implementation in the syllog system. In *Proc. of the 6th int. workshop on machine learning*, pages 404–407. Morgan Kaufmann, 1989.
8. R. S. Michalski, I. Mozetic, J. Hong, and N. Lavrac. The multi-purpose incremental system AQ15 and its testing application to three medical domains. In *Proc. of the American Association for AI Conference*, Philadelphia, 1986.
9. P. G. Neumann. An attribute frequency model for the abstraction of prototypes. *Memory and Cognition*, 2:241–248, 1974.
10. B. W. Porter, R. Bareiss, and R. C. Holte. Concept learning and heuristic classification in weak-theory domains. *Artificial Intelligence*, 45:229–263, 1990.
11. M. I. Posner and S. W. Keele. Retention of abstract ideas. *Journal of Experimental Psychology*, 83:304–308, 1970.
12. J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.
13. L. Rendell. A general framework for induction and a study of selective induction. *Machine Learning*, 1:177–226, 1986.
14. E. Rosch. On the internal structure of perceptual and semantic categories. In T. E. Moore, editor, *Cognitive development and the acquisition of language*. Academic Press, 1973.
15. E. Rosch and C. B. Mervis. Family resemblance studies in the internal structures of categories. *Cognitive Psychology*, 7:573–605, 1974.

16. S. L. Salzberg. *Learning with nested generalized exemplars*. Kluwer Academic Publishers, Norwell: MA, 1990.
17. J. C. Schlimmer and R. H. Granger. Incremental learning from noisy data. *Machine Learning*, 1(1):317-354, 1986.
18. L. Wittgenstein. *Philosophical investigations*. Macmillan, New York, 1953.
19. J. Zhang. Selecting typical instances in instance-based learning. In *Proc. of the 9th intl. conf. on machine learning*, pages 470-479. Morgan Kaufmann, 1992.