Selecting Prototypes in Mixed Incomplete Data

Milton García-Borroto[1] and José Ruiz-Shulcloper[2]

[1] Bioplants Center, UNICA, C. de Ávila, Cuba
mil@bioplantas.cu
http://www.bioplantas.cu
[2] Advanced Technologies Applications Center, MINBAS, Cuba
jshulcloper@cenatav.co.cu
http://www.cenatav.co.cu/

Abstract. In this paper we introduce a new method for selecting prototypes with Mixed Incomplete Data (MID) object description, based on an extension of the Nearest Neighbor rule. This new rule allows dealing with functions that are not necessarily dual functions of distances. The introduced compact set editing method (CSE) constructs a prototype consistent subset, which is also subclass consistent. The experimental results show that CSE has a very nice computational behavior and effectiveness, reducing around 50% of prototypes without appreciable degradation on accuracy, in almost all databases with more than 300 objects.

1 Introduction

Supervised classifiers need a good training matrix for classifying with effectiveness. This "goodness" is usually achieved by expert criterion, but sometimes even experts make this selection arbitrarily. These classifiers typically compare a new unclassified object with all stored classified ones to make a decision. This can make them prohibitively costly for large training sets. One possible solution to these problems is to reduce the cardinality of the object descriptions sample, while simultaneously insisting that the decisions based on the reduced data set perform as well, or nearly as well, as the decisions based on the original data set. This process is known as finding prototypes.

There are two different goals approached while finding prototypes:

- Minimize the size of the training set (condensing methods).
- Reduce the size of the training set obtaining classification accuracy never worse than with the initial training matrix (editing methods).

On the other hand, in order to solve practical real problems, especially in soft sciences, we have to deal frequently with description of objects that are *non-classical*, that is, the features are not exclusively numerical or categorical. Both kinds of values can appear simultaneously, and a special symbol is necessary to denote the absence of values (missing values). A *mixed and incomplete description* of objects should be used in this case (MID). Many examples of real problems with this sort of objects can be found [1, 2] and also in the UCI Repository of Machine Learning Databases [3].

M. Lazo and A. Sanfeliu (Eds.): CIARP 2005, LNCS 3773, pp. 450 – 459, 2005.

Although the terms distance and dissimilarity have been widely exchanged, it is not true that a dissimilarity function is always dual to a distance function. There are many practice applications that use non-reflexive and/or non-symmetrical dissimilarities, which their duals are evidently not distances [4, 5].

Most prototype selection algorithms were developed to deal with distances defined in metric spaces, which almost never is possible to use while working with MID. Some of them may be trivially extended to work with MID (Hart's CNN [6], Wilson's ENN [7], Random [8]) and many others do not, because use properties of distances and metric spaces for working (Construction of new prototypes [9], proximity graphs [10]).

2 Basic Concepts

Let U a universe of objects, structured in $K_1, \ldots, K_r$ classes, described in terms of a finite set of features $R=\{x_{1,\ldots}\ x_n\}$. Each of these features has associated a set of admissible values M_i, which include de value '*' for the case of unknown value. Over M_i no algebraic, topologic of logic structure is assumed. Then be $U=M_1\times\ldots\times M_n$, the Cartesian product of the admissible values sets of features of R. Let $O=(x_1(O), x_2(O), \ldots, x_n(O))$, where $x_i:U\rightarrow M_i$. A *comparison criterion* $\varphi_i:M_i\times M_i\rightarrow L_i$ is associated to each x_i, where L_i is a totally ordered set. A *similarity function* is a function Γ as be defined in [11]. $\Gamma(O_1,O_2)$ is an evaluation of the degree of similarity between any two descriptions of objects belonging to U. Any restriction of Γ to any subset of R will be called a *partial similarity function*. Besides, this function is characterized by the following properties: the partial similarity relationships between any pair of objects are preserved when the total similarity between these objects is considered. Also, the maximum value of similarity is reached when the same part of the same object for any non-empty subset of R is considered, including the case of whole R.

There are many pattern recognition algorithms for either numerical data processing or categorical data processing, that can be extended for the case of MID. These extensions are scarce and non trivial because it is necessary to face several problems. One of the simplest is the assumption of a distance for the comparison of MID.

Nearest neighbor rule can not be applied with similarities which are non-dual to distances because the term "near" is associated with distances, while the term "most similar" is associated with analogies.

Let $\alpha(O) =(\alpha_1(O),\ldots, \alpha_r(O))$ the membership t-uple of O in which $\alpha_i(O)$ means the grade of membership of O to the class K_i, $i=1,..,r$. For example, it could have $\alpha_i(O) =\{0,1\}$ with the obvious interpretation. Let $Q=\bigcup_{i=1}^{r} K_i^{'}$, $K_i^{'}\subset K_i$, $i=1,\ldots r$, a training set.

Let $O\in U\backslash Q$, the *most similar neighbor rule* (MSN) for classifying O is to assign it the membership t-uple $\alpha(O)$ in the following way:

A) Assuming Γ as just a similarity function

If $\max\left\{\max_{O_i\in Q}\{\Gamma(O,O_i)\},\max_{O_i\in Q}\{\Gamma(O_i,O)\}\right\}=\Gamma(O,O')$ or $\Gamma(O',\ O)$ then $\alpha(O)=\alpha(O')$

B) Assuming Γ as symmetric similarity function
If $\max_{O_i \in Q} \{\Gamma(O, O_i)\} = \Gamma(O, O')$ then $\alpha(O) = \alpha(O')$, with $O' \in Q$

Observe that in these cases MSN rule does not require that K be a partition neither a hard structuralization of U.

We say that O_i, $O_j \in U$ are *β_0-similar objects* if $\Gamma(O_i, O_j) \geq \beta_0$. In the same way O_i is a *β_0-isolated object* if $\forall O_j \neq O_i \in U \;\; \Gamma(O_j, O_i) < \beta_0$ .The β_0 threshold value can be used to control how similar a pair of objects must be in order to be considered β_0-similar.

Definition. $NU \subseteq U$, $NU \neq \varnothing$ is a *compact set* if: [11]

- $\forall O_j \in U \left[O_i \in NU \wedge \max_{\substack{O_i \in U \\ O_i \neq O_i}} \left\{\Gamma(O_i, O_t)\right\} = \Gamma(O_i, O_j) \geq \beta_0 \right] \Rightarrow O_j \in NU$
- $\left[\max_{\substack{O_i \in U \\ O_i \neq O_{pi}}} \left\{\Gamma(O_p, O_i)\right\} = \Gamma(O_p, O_t) \geq \beta_0 \wedge O_t \in NU \right] \Rightarrow O_p \in NU$
- $|NU|$ is minimal.
- Every β_0-isolated object is a compact set (degenerated).

The compact set criterion induces a unique partition for a given data set, which has the property that one object x and all its most similar neighbors belongs to the same cluster and also, those objects for which x is its most similar neighbor.

In many classification problems, a class is not uniformly formed. Consider, for example, in the universe of all humans we can define two classes: *S* is the class of all who are sick, and *H* is the class of all who are healthy. In the class *S* are grouped together many different objects with many different diseases, which compose *subclasses* inside the outer class. Intuitively, if an object belongs to a subclass its most similar neighbor must be in the same set, so it is obvious that a subclass should be considered as a union of compact sets.

Consider now the problem of selecting a set of prototypes which describes this problem. We face two important difficulties:

1. Selecting the number of prototypes per subclass (and obviously per class) can not be done *a priori*, because it depends on the inner structure of each subclass, which can only safely be inferred from data.
2. If the subclass structure of the class is not preserved someway it may be a serious degradation on accuracy, and it may be whole subclasses without a single representative. That is why it is important to introduce a new kind of consistency.

Let $Q \subset U$ a training matrix of a set of classes $K=\{K_1, K_2, \ldots, K_r\}$, $Cf(LM, x)$ a classifier with learning matrix LM and $MSN_R(x)$ the most similar neighbor of object x in set R.

Following Hart [6] $R \subset Q$ is a *prototype consistent* subset with respect to (wrt) *Cf* and *Q* iff $\forall x \in Q[Cf(Q,x) = Cf(R,x)]$

Definition. *Let* Φ *a partition of Q in subclasses, such that* $\forall i \in I[\Phi_i \in \Phi\ x_1, x_2 \in \Phi_i \Rightarrow \alpha(x_1) = \alpha(x_2)]$ *and* $R_i \subset \Phi_i$, *a set of representatives associated to each subclass.* $R = \bigcup_{i \in I} R_i$ *is subclass consistent wrt* Φ iff:

$$\forall i \in I\ \forall x \in Q[(x \in \Phi_i \Rightarrow MSN_R(x) \in \Phi_i]$$

3 Compact Set Editing (CSE) Algorithm

Inputs:

- β_0-compact sets in a maximal similarity graph (oriented graph each edge from vertex *a* to vertex *b* means that *b* is the most β_0-similar element of *a*) described by a set of edges C and a set of vertexes V.
- $\alpha(x)$: class associated with vertex *x*

Output:

- Subset of selected prototypes *R*

Notations:

- $S(x) = \{b \in V / (x, b) \in C\}$, set of the successors of vertex *x* in graph *V*. The presence of these elements in *K* guarantee the good classification of *x*
- $A(x) = \{a \in V / (a, x) \in C\}$, set of the predecessors of vertex *x* in graph *V*.

0. $R = \varnothing$
1. Let associate each vertex *x* in *V* with a quadruple (S'_x, E_x, S_x, $Flags_x$), where:
   $S'_x = |\{y \in S(x) / \alpha(x) \neq \alpha(y)\}|$
   $E_x = |\{y \in A(x) / \alpha(x) = \alpha(y)\}|$
   $S_x = |\{y \in S(x) / \alpha(x) = \alpha(y)\}|$
   $Flags_x \subset V$, $Flags_x = \varnothing$
2. $R' = \{x \in V / S'_x > 0\}$
3. If $R' = \varnothing$ go to step 6
4. $C \leftarrow C \setminus \{(x, y) \in C / x \in R' \wedge \alpha(x) \neq \alpha(y)\}$
5. For each element $x \in R'$ execute Move(*x*).
6. $\forall x \in V\ [(S_x = 0) \Rightarrow$ execute Move(*x*)]
7. $\forall x \in V\ \forall y \in Flags_x \left[y \notin \bigcup_{z \in V \setminus \{x\}} Flags_z \Rightarrow \text{execute Move(x)} \right]$
8. Sort the elements of V with the following order relation:
   $x \prec y \Leftrightarrow E_x < E_y \vee (E_x = E_y \wedge S_x < S_y) \vee (E_x = E_y \wedge S_x = S_y \wedge |Flags_x| > |Flags_y|)$
9. Execute Discard(x_1), where x_1 is the first vertex of V ($\prec$)
10. If $V = \varnothing$ end, else go to step 6

Move(x)	**Discard(x)**
m1. Calculate A(x) and S(x) with the current set of vertexes V.	d1. Calculate A(x) and S(x) with the current set of vertexes V.
m2. $\forall\, y \in \text{A}(x)\ [S_y \leftarrow \infty]$	d2. $\forall\, y \in \text{S(x)}\ [Flags_y \leftarrow Flags_y \cup \{x\}]$
m3. $\forall\, y \in \text{S}(x)\ [E_y \leftarrow E_y - 1]$	d3. $\forall\, y \in \text{S(x)}\ [E_y \leftarrow E_y - 1]$
m4. $\forall\, y \in V\ [Flags_y \leftarrow Flags_y \setminus Flags_x]$	d4. $\forall\, y \in \text{A(x)}[S_y \neq \infty \Rightarrow S_y \leftarrow S_y - 1$
m5. $V \leftarrow V - \{x\}, R \leftarrow R \cup \{x\}$	d5. $V \leftarrow V - \{x\}$
m6. $C \leftarrow C \setminus \{(a, b) \in C \,/\, a = x \vee b = x\}$	d6. $C \leftarrow C \setminus \{(a, b) \in C \,/\, a = x \vee b = x\}$

The indexes calculated in step 1 are the core of the later decision of which vertex to select and which to discard. Steps 2-6 break the compact sets eliminating the edges connecting vertexes with different classes, leaving "pure" components (*v. gr.* in figure 1b eliminating edges c-d, d-c and e-d). To guarantee consistency predecessors nodes in this edges are moved to R, because they would be bad classified if do not (its MSN have a different class).

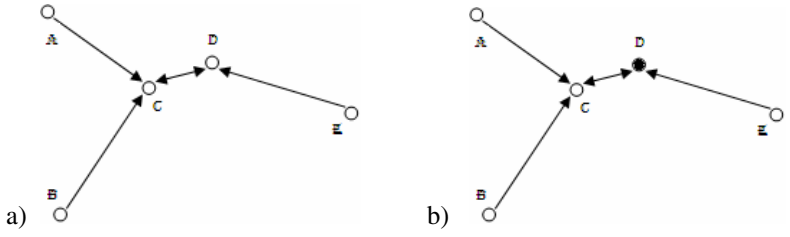

Fig. 1. Maximal similarity graph with a single class (a) and a couple of classes (b)

Let see how the algorithm decides what objects should be included in the result set. Suppose that the graph represented in Figure 1 is associated with a real problem. As you can see nodes C and D are more important than nodes A, B and E, because their presence in the result set guarantee the good classification of the rest of the nodes. The order relation defined assures that nodes with low importance are removed first from the set, and an additional process is done to keep consistency: if a node is discarded one of its MSN must to stay. This is done "flagging" all the successors of x with a non-simultaneous elimination mark (step d2). In step 7 if an object is the last having such "flag", it is automatically moved to R. After each modification in the graph, the indexes are updated. If the good classification of some node x is already assured, its S_x is assigned the value infinite, meaning that this information is no longer necessary for that object.

In the example, node A is the first discarded, flagging C as its only successor. Node C is moved to result set because is the only one to have the "A" flag. So, nodes B and D have S_x equal infinite. All indexes are recalculated, and the process is repeated again. Finally the result set is nodes C and D. Note that this set is prototype consistent, no matter the distribution of the other objects in the space, because of the use of maximal similarity graph.

Let demonstrate some properties of the algorithm.

Proposition 1. *Let Cf the classifier defined by the MSN rule. We have:*
($R \subset Q$ is prototype consistent) $\Leftrightarrow \forall x \in Q\left[\alpha(x) = \alpha(MSN_R(x))\right]$

Proof. If $R \subset Q$ is prototype consistent, then by definition we have $\forall x \in Q[Cf(Q, x) = Cf(R, x)]$ (1).
$Cf(Q, x) = \alpha(MSN_Q(x)) = \alpha(x)$, because x is its own MSN (2).
$Cf(R, x) = \alpha(MSN_R(x))$, because the definition of MSN (3).
Substituting (2) and (3) in (1) we have $\forall x \in Q\left[\alpha(x) = \alpha(MSN_R(x))\right]$.
The back implication is also obvious.

Proposition 2. *If a set of prototypes $R \subset Q$ is subclass consistent wrt a partition Φ, then it is prototype consistent wrt Q.*
Proof.
This is obvious based on the fact that the partition Φ is such that two elements in the same subclass have the same class.

Theorem 1. *The result set of the algorithm CSE is subclass consistent wrt the partition induced by the β_0-connected subgraphs.*
Proof. Basically the CSE algorithm, for each $x \in Q$ decides if $x \in R$ or not (and then its most similar neighbor $MSN_Q(x) \in R$), so we have
$\forall x \in Q\left[x \in R \vee MSN_Q(x) \in R\right]$

Let $x \in R$ and $x \in V_i$, then $x \in R \cap V_i = R_i$ (1)
$x \in R$ implies that $MSN_R(x) = x$, because x is its own MSN in R. (2)
By (1) and (2) we have that $MSN_R(x) \in R_i$, and then $MSN_R(x) \in V_i$ (3)
Let $MSN_Q(x) \in R$ and $x \in V_i$ (4)
$x \in V_i$ implies that $MSN_Q(x) \in V_i$, because V_i is a β_0-compact set (5).
From (4) and (5) we have:
$MSN_Q(x) \in R$ and $MSN_Q(x) \in V_i$, so $MSN_Q(x) \in R \cap V_i = R_i$, and then, because $R_i \subset R$ and $R \subset Q$, we have $MSN_R(x) \in R_i$, and finally $MSN_R(x) \in V_i$. (6)
By (3) and (6) we have:
$\forall x \in Q\left[x \in R \vee MSN_Q(x) \in R\right] \Rightarrow \forall i \in I \forall x \in Q\left[x \in V_i \Rightarrow MSN_R(x) \in V_i\right]$
what prove the theorem.

4 Experimental Results

Traditionally all prototypes selection methods have been defined in $\Re^n$ with distances functions. Many of them can not be extended to deal with MID, because they need properties of metric spaces, for example, the existence of an addition and multiplication operator. We have trivially extended some methods, originally enounced for working in metric spaces, allowing the comparisons with CSE. These methods are: AllKnn [12], Hart's CNN [6], IB2 [13], Dasarathy's MCS [14], Random [8], Relative neighbor editing [10], Random Mutation Hill Climbing (RMHC) [15], Shrink [16] and Wilson's ENN [7].

We use 27 databases from UCI Repository of Machine Learning with mixed and incomplete object description. Each database was split randomly, taking 30% for training (training matrix) and 70% for testing (control matrix). To reduce the influence of the randomness in partition, we repeat the process 5 times, and average the results. We measure the accuracy (# correct classification / # of objects) of each method by the difference of the accuracies over the training matrix and the edited matrix with respect to the control matrix, respectively.

A MSN classifier was used for testing, without weighing the features, because we are only interested in the differences between the selection methods, more than finding a best classifier for a particular example.

Table 1 shows the databases used in the experiments, the size of those databases and the list of methods that outperforms CSE in both, compression ratio and accuracy

Table 1. Databases used in the experiments

Number	UCI name	Objects	Outperforms CSE
1.	Annealing	257	MCS
2.	Audiology	64	-
3.	Breast cancer 1	230	Random
4.	Breast cancer 2	182	Random
5.	Breast cancer 3	69	RMHC
6.	Credit-screening	228	Random, RMHC
7.	Heart-disease Cleveland	94	Random, RMHC
8.	Heart-disease Hungarian	91	Random, RMHC
9.	Heart-disease Long Beach	63	MCS, ENN
10.	Heart-disease Switzerland	37	AllKnn, MCS, Random, RMHC, Shrink, ENN
11.	Hepatitis	56	-
12.	Horse-colic	96	MCS, Random
13.	Monks-problems 1	186	Shrink
14.	Monks-problems 2	194	AllKnn, IB2, Random, RMHC
15.	Monks-problems 3	184	MCS
16.	Mushroom	2655	MCS
17.	Soybean large	95	MCS
18.	Thyroid-disease Allbp	903	-
19.	Thyroid-disease ann	2399	-
20.	Thyroid-disease dis	1243	-
21.	Thyroid-disease hyper	936	-
22.	Thyroid-disease hypo	1233	-
23.	Thyroid-disease hypothyroid	1049	-
24.	Thyroid-disease new-thyroid	72	-
25.	Thyroid-disease rep	1246	-
26.	Thyroid-disease sick	1268	-
27.	Thyroid-disease sick-euthyroid	1055	-

difference. In the 27 databases evaluated, twelve of them CSE had the best behavior, in eight cases was outperformed by only one of the nine methods (not always the same) and in the remainder cases in which was outperformed by other methods, the databases were small.

We can also observe that gaining in compression ratio by other classifiers above CSE will lead to a drastic reduction in classification accuracy, as shown in **Table 2** and **Table 3** (bolded rows). Random based and evolutive methods (Random and RMHC) have a good performance in small databases [8], but are usually slow and inaccurate for big ones. MCS exhibit good performance for medium size database, but for big ones is always worse than CSE.

Table 2. Results of prototype selection for "thyroid-disease ann" database

Method Name	Acc. Difference & Comp. Ratio		Time(sec)
CSE	-1,75	53,9	151,93
RMHillClimb	-0,43	50,1	670,91
RelativeNeighborEditor	**-5,10**	**79,37**	**3738,04**
MCS	**-5,54**	**83,74**	**487,32**
IB2	**-13,99**	**85,16**	**10,95**
Shrink	**-28,45**	**88,70**	**81,07**
AllKnn	1,98	8,71	1173,69
WilsonENN	1,70	5,04	148,54
CNN	-6,99	8,63	115,64

We have to note than the Time result shown in tables are only useful for comparisons, because the absolute value is highly dependant on the computer where they are executed.

Table 3. Results of prototype selection for "thyroid-disease dis" database

Method Name	Acc. Difference & Comp. Ratio		Time
CSE	-0,42	58,25	57,1418
RMHillClimb	-0,22	50,52	248,1506
MCS	**-2,54**	**95,58**	**121,0522**
IB2	**-6,33**	**95,09**	**1,5004**
CNN	-2,31	35,32	38,7486
Wilson ENN	0,25	1,21	55,376
AllKnn	0,28	1,85	443,863
Shrink	**-21,71**	**97,35**	**28,3314**

The compression ratio of the method is around 50% of the prototypes in almost all databases, and the reduction of accuracy for medium and big databases is usually lower than 1. The behavior of the remainder methods is not so stable, and is more dependent to the data nature.

5 Conclusions

Many practical pattern recognition problems, especially many of those appearing in soft sciences (medicine, geosciences, criminology, and others), make a necessity to work with MID. Training set prototype selection is a core issue for improving the efficiency and efficacy of many supervised classifiers. To face those problems, firstly we have extended the well known NN rule to MSN, for allowing to work with similarity functions non necessarily dual to distances and with object representation spaces different to metric spaces, which is usual while working with MID. We have defined subclass consistency property, to preserve the subclass structure of the data set while selecting a subset of prototypes.

A new prototype selection method has been introduced (CSE). It works with MID and more general similarities (even non-symmetric or non-positive defined). It produces a subclass consistent subset. We have shown that this algorithm has a good performance compared to other prototype selection algorithms that can be used also with MID after a trivial extension. The new method is neither a pure condensing method nor a pure editing method, having desirable properties of both. Also the method leverages the user to spend time in selecting the training matrix, doing the selection automatically.

Based on preliminary experiments and the results shown, CSE seems to be very adequate for synergy of editing methods with mixed incomplete data, in which we are actually working.

References

1. F. Martínez-Trinidad and A. Guzmán-Arenas. The logical combinatorial approach to Pattern Recognition, an overview through selected works. *Pattern Recognition*, 34: 741-751, 2001.
2. J. Ruiz-Shulcloper and M. A. Abidi. Logical combinatorial pattern recognition: A Review. In S. G. Pandalai, editors, Recent Research Developments in Pattern Recognition. Transword Research Networks, USA.
3. C. J. Merz and P. M. Murphy. UCI Repository of Machine Learning Databases. Technical report, University of California at Irvine, Department of Information and Computer Science, 1998.
4. M. Sato and Y. Sato. Extended fuzzy clustering models for asymmetric similarity. In B. Bouchon-Meunier, R. Yager, and L. Zadeh, editors, Fuzzy logic and soft computing. World Scientific.
5. H. Chen and K. J. Lynch. Automatic construction of networks of concepts characterizing document databases. *IEEE Transactions on systems, man and cybernetics.*, 22: 885-902, 1992.
6. P. E. Hart. The condensed nearest neighbor rule. *IEEE Trans. on Information Theory*, 14: 515-516, 1968.
7. D. L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on systems, man and cybernetics*, SMC-2: 408-421, 1972.
8. L. I. Kuncheva and J. C. Bezdek. Nearest prototype classification: clustering, genetic algorithms or random search. *IEEE transactions on systems, man and cybernetics. Part C*, 28: 160-164, 1998.

9. S.-W. Kim and J. B. Oommen. A brief taxonomy and ranking of creative prototype reduction schemes, in IEEE SCM Conference, 2002.
10. G. T. Toussaint. Proximity Graphs for Nearest Neighbor Decision Rules: Recent Progress, in 34 Symposium on Computing and Statistics INTERFACE-2002, 2002.
11. J. F. Martínez-Trinidad, J. Ruiz-Shulcloper, and M. S. Lazo-Cortés. Structuralization of universes. *Fuzzy sets and systems*, 112: 485-500, 2000.
12. I. Tomek. Two modifications of CNN. *IEEE Transactions on systems, man and cybernetics*, SMC-6: 769-772, 1976.
13. D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6: 37-66, 1991.
14. B. D. Dasarathy. Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design. *IEEE Transactions on systems, man and cybernetics.*, 24: 511-517, 1994.
15. D. B. Skalak. Prototype and Feature Selection by Sampling and Random Mutation Hill Climbing Algorithms, in Eleventh International Conference on Machine Learning, 1994.
16. D. Kibler and D. W. Aha. Learning representative exemplars of concepts: An initial case study., in Fourth international workshop on Machine learning, pages 24-30, 1987.