Gonzalez Rodriguez, I., Lawry, J., & Baldwin, J. F. (2002). A Hierarchical Linguistic Clustering Algorithm for Prototype Induction. In Proc. of IPMU 2002.

IPMU 2002, July 1-5, 2002
Annecy, France

A Hierarchical Linguistic Clustering Algorithm for Prototype Induction

I. González Rodríguez, J. Lawry, J.F. Baldwin
A.I. Group, Dept. of Engineering Mathematics, University of Bristol, Bristol BS8 1DJ (U.K.)
{ines.gonzalez, j.lawry, jim.baldwin}@bristol.ac.uk

Abstract

A clustering algorithm is described which learns fuzzy prototypes to represent data sets and decides the number of prototypes needed. This algorithm is based on a modified hierarchical clustering scheme and incorporates ideas taken from mass assignment theory. It is illustrated using a model classification problem and its potential is shown by its application to a benchmark problem for glass identification.

Keywords: Fuzzy Prototypes, Clustering, Mass Assignment.

1 Introduction

In many of the emerging information technologies, there is a clear need for automated learning from databases. Data mining methods attempt to extract useful general knowledge from the implicit patterns contained in databases. Machine learning approaches learn models of complex systems capable of accurate prediction. Such methods have applications to classification problems, as well as vision, function approximation and control. For example, supermarkets are interested in learning prototypical descriptions of customers with certain purchasing behaviour; these models can then be used to learn descriptions of certain types of customers and to make informed decisions about levels of certain goods to stock, as well as pricing.

It is this need for automated learning that motivates our clustering algorithm, which tries to learn fuzzy prototypes to represent data sets and also decides the number of prototypes needed. Here, prototypes correspond to tuples of fuzzy sets on words over attribute universes and as such represent amalgams of similar objects sharing particular properties. It is this idea of grouping similar object that is central to the model of prototype induction proposed in this paper.

2 Basic Mass Assignment Theory

The mass assignment for a fuzzy concept, first introduced by Baldwin [1, 4], can be interpreted as a probability distribution over possible crisp definitions of the concept. We might think of these varying definitions as being provided by a population of voters where each voter is asked to give his or her crisp definition of the concept.

Definition (Mass Assignment) Let f be a fuzzy set on a finite universe Ω such that the range of the membership function of f, χ_f, is $\{y_1, \ldots, y_n\}$ where $y_i > y_{i+1} > 0$. Then, the *mass assignment* of f, denoted m_f, is a probability distribution on 2^Ω satisfying

$$\begin{aligned} m_f(\emptyset) &= 1 - y_1 \\ m_f(F_i) &= y_i - y_{i+1} \text{ for } i = 1, ..., n-1 \\ m_f(F_n) &= y_n \end{aligned}$$

where $F_i = \{x \in \Omega \,|\, \chi_f(x) \geq y_i\}$ for $i = 1, \ldots, n$. $\{F_i\}_{i=1}^n$ are referred to as the *focal elements(sets)* of m_f.

The notion of mass assignment suggests a means of conditioning a variable X relative to a fuzzy constraint 'X is f' to obtain a probability distribution, by redistributing the mass associated with every focal set uniformly to the elements of that set. The probability distribution on X generated in this way is referred to as the *least prejudiced distribution* of f [1].

Definition (Least Prejudiced Distribution) For f a fuzzy subset of a finite universe Ω such that f is normalised, the *least prejudiced distribution of* f, denoted lp_f, is a probability distribution on Ω given by

$$lp_f(x) = \sum_{F_i : x \in F_i} \frac{m_f(F_i)}{|F_i|}$$

The idea of least prejudiced distribution provides us with an alternative definition of the conditional probability of fuzzy events [1].

Definition (Conditional Probability) For f and g fuzzy subsets of a finite universe Ω where g is normalised assuming no prior knowledge we define

$$\Pr(f|g) = \sum_{x \in \Omega} \chi_f(x) lp_g(x)$$

The least prejudiced distribution allows us, in a sense, to convert a fuzzy set into a probability distribution. That is, in the absence of any prior knowledge, we might on being told f naturally infer the distribution lp_f. Now, if we can find a method by which, when presented with a probability distribution, we can infer the fuzzy constraint generating that distribution, we can use fuzzy sets as descriptors of probability distributions.

Theorem Let Pr be a probability distribution on a finite universe Ω taking as a range of values $\{p_1, \ldots, p_n\}$ where $0 \leq p_{i+1} < p_i \leq 1$ and $\sum_{i=1}^{n} p_i = 1$. Then, Pr is the least prejudiced distribution of a fuzzy set f if and only if f has a mass assignment given by

$$\begin{aligned} m_f(F_i) &= y_i - y_{i+1} \text{ for } i = 1, \ldots, n-1 \\ m_f(F_n) &= y_n \end{aligned}$$

where

$$\begin{aligned} F_i &= \{x \in \Omega \mid \Pr(x) \geq p_i\} \\ y_i &= |F_i| p_i + \sum_{j=i+1}^{n} (|F_j| - |F_{j+1}|) p_j \end{aligned}$$

Proof (See [2])

It is interesting to note that this transformation algorithm is identical to the bijective transformation method proposed by Dubois and Prade [6], although the motivation here is quite different. A further justification for this transformation can be found in [10].

Definition (Fuzzy Description) For a probability distribution Pr on a finite universe Ω, we refer to the fuzzy set generated from Pr according to the previous theorem as the *fuzzy description* of Pr, denoted $FD(\Pr)$.

3 Fuzzy Prototypes

We now use ideas from mass assignment theory to infer a number of prototypes representing a set of instances and where each prototype corresponds to a grouping of similar points (sometimes called a granule [11]). Unlike many current clustering methods, we do not intend to define single instances as being prototypical. Instead, a prototype is taken to be an amalgam of points represented by a tuple of fuzzy sets on each of the attributes describing an instance.

Definition (Fuzzy Prototype) A *fuzzy prototype* in $\Omega_1 \times \cdots \times \Omega_n$ is a n-tuple of fuzzy sets $\langle f_1, \ldots, f_n \rangle$ where f_i is a fuzzy subset of Ω_i.

In particular, we are interested in fuzzy prototypes generated from a set of elements so that they constitute a description of these elements. In order to determine which vectors should be associated with which prototypes, we need to define a notion of similarity. Furthermore, since data vectors can be viewed as a special case of prototypes, we need to define a similarity relation between fuzzy prototypes. Tversky's statement that similarity "may be better described as a comparison of

features rather than as a computation of metric distance between points" [8] inspires the following definition, based on similarity measures between fuzzy sets.

Definition (Prototype Similarity) Let s_i be a similarity measure between fuzzy subsets on Ω_i and Π the class of all prototypes in $\Omega_1 \times \cdots \times \Omega_n$. The *prototype similarity measure* is a function $Sim : \Pi \times \Pi \mapsto [0,1]$ where, given the prototypes $P_1 = \langle f_1, \ldots, f_n \rangle$ and $P_2 = \langle g_1, \ldots, g_n \rangle$ in Π,

$$Sim\,(P_1, P_2) = \frac{1}{n} \sum_{i=1}^{n} s_i(f_i, g_i)$$

There is wide variety of similarity measures proposed in the literature which are possible candidates for s_i. For example, in [7, p24] there is a list of similarity indices which are a generalisation of the classical set-theory similarity functions. It is from this list that we have chosen a specific similarity measure to obtain the results shown in sections 6 and 7. Given f, g fuzzy sets on a finite universe Ω, we will take the similarity between them to be:

$$\begin{aligned} s(f,g) &= \frac{\sum_{x \in \Omega} \min(\chi_f(x), \chi_g(x))}{\sum_{x \in \Omega} \max(\chi_f(x), \chi_g(x))} \\ &= \frac{|f \cap g|}{|f \cup g|} \end{aligned}$$

Once it is clear what we understand by similar points (or prototypes), we need to define a means of grouping them. We do so by defining prototype addition.

Definition (Prototype Addition) Let $P_1 = \langle f_1, \ldots, f_n \rangle$ and $P_2 = \langle g_1, \ldots, g_n \rangle$ be prototypes in $\Omega_1 \times \cdots \times \Omega_n$ representing a granule of k and c data points respectively. Then, $P_1[+]P_2$ is a prototype in $\{\Omega_1, \ldots, \Omega_n\}$ such that

$$P_1[+]P_2 = \langle FD(r_1), \ldots, FD(r_n) \rangle$$

where r_i is the probability distribution in Ω_i given by

$$\forall x \in \Omega_i \; r_i(x) = \frac{k\, lp_{f_i}(x) + c\, lp_{g_i}(x)}{k + c}$$

4 Linguistic Variables

In the above, we have only considered finite universes, but most real-world problems involve continuous attributes. In this case, we need some way of converting infinite universes into finite ones so that the methods described in section 2 can be applied. Thus, we require some way of partitioning the universes associated with the continuous attributes. Fuzzy sets can be used to divide such universes into information granules, a term defined by Zadeh [11] as a group drawn together by similarity which can be viewed as corresponding to the meaning of words from natural language. Then, a linguistic variable can be defined which is associated with the original continuous attribute and takes as it values the words. Fuzzy sets on words can then be inferred.

Definition (Linguistic Variable) [1] A *linguistic variable* is a quadruple $\langle L, T(L), \Omega, M \rangle$ in which L is the name of the variable, $T(L)$ is a finite term set of labels or words (i.e. the linguistic values), Ω is a universe of discourse and M is a semantic rule.

The semantic rule M is defined as a function that associates a normalised fuzzy subset of Ω with each word in $T(L)$. In other words, the fuzzy set $M(w)$ can be viewed as encoding the meaning of w so that for $x \in \Omega$ the membership value $\chi_{M(w)}(x)$ quantifies the suitability or applicability of the word w as a label for the value x. This generates a fuzzy subset of $T(L)$ describing the value of x.

Definition (Linguistic Description of a Value) For $x \in \Omega$, the *linguistic description* of x relative to the linguistic variable L is the fuzzy subset of $T(L)$

$$des_L(x) = \sum_{w \in T(L)} w / \chi_{M(w)}(x)$$

In cases where the linguistic variable is fixed,

[1] Zadeh [12] originally defined linguistic variable as a quintuple by including syntactic rule according to which new terms (i.e. linguistic values) could be formed by applying quantifiers and hedges to existing words. In this context, however, we shall asume that the term set is predefined and finite

the subscript L is dropped and we write $des(x)$.

Once we have a fuzzy set on words for a given $x \in \Omega$, $des(x)$, which is a fuzzy subset of a finite universe $T(L)$, the basic mass assignment theory is applicable and it is possible to consider the least prejudiced distribution of $des(x)$, $lp_{des(x)}$, this being a probability distribution in the set of labels $T(L)$.

Now recall that the least prejudiced distribution is defined only for normalised fuzzy sets and, hence, it is desirable that linguistic descriptions as defined above be normalised. This will hold if and only if the semantic function generates a linguistic covering defined as follows.

Definition (Linguistic Covering) A set of fuzzy sets $\{f_i\}_{i=1}^n$ forms a *linguistic covering* of the universe Ω if and only if

$$\forall x \in \Omega \ \max_{i=1}^{n} (\chi_{f_i}(x)) = 1$$

Having described how fuzzy sets can be used as descriptors for probability distributions, we shall introduce a prototype induction algorithm based on these ideas. The prototypes induced belong to an specific type of fuzzy prototype (as described in section 3) where each attribute is described by a fuzzy set on words, with labels provided by the linguistic covering of the attribute's universe. We may think of each of these fuzzy sets as a possibility distribution on the set of linguistic labels describing the attribute's universe. In this way we can evaluate the possibility that a particular label describes an attribute for a certain prototype.

5 A Hierarchical Linguistic Clustering Algorithm

Traditional hierarchical clustering algorithms produce a series of partitions of the data, $P_n, P_{n-1}, \ldots, P_1$. The first, P_n, consists of n single-member clusters, the last P_1 consists of a single group containing all n individuals. At each particular stage, the methods fuse individuals or groups of individuals which are closest (or most similar), it being the chosen definition of 'closeness' that differentiates one method from another. A description of these methods can be found in [5, pp55–90].

However, it is often the case, when hierarchical clustering techniques are used, that what is of interest is not the complete hierarchy obtained by the clustering, but only one or two partitions obtained from it that contain an 'optimal' number of clusters. Therefore, it is necessary to select one of the solutions in the nested sequence of clusterings that comprise the hierarchy. This in itself is a very challenging problem.

Another issue that should be addressed is the computational complexity of exhaustively searching for the pair of most similar elements at each stage of the clustering. This may become an important issue when the number of data points n is 'large', which is usually the case in practical applications.

In our algorithm, we try to overcome these problems by introducing several changes in the scheme described above. First of all, we have already mentioned that each prototype corresponds to a grouping of similar points. Therefore, if two prototypes are not 'similar enough', this implies that the elements in the clusters they represent are not close and should not be merged into a single group. For this reason, we define a similarity threshold $\sigma \in [0, 1]$, according to which the grouping of clusters will terminate once the similarity between prototypes falls below σ.

Also, we will introduce a heuristic search for pairs of similar prototypes and we will allow more than one pair to fuse at each level of the clustering. The search for similar elements in one partition P_i will start with the first cluster in P_i, according to some arbitrary ordering, and will go through the elements in this partition to select the most similar one to it. If the similarity between these two clusters is high enough according to the threshold σ, then they should be merged into only one cluster and this new group should be added to the next partition P_{i-1}. If, on the contrary, they are not similar enough, the first cluster alone will be added to the next partition. We re-

peat this process with the remaining clusters in P_i until all of them have been considered.

This reduces the complexity of the original algorithms in two ways. First of all, by using this heuristic we do not need to compare all the elements to find the most similar pair. Secondly, if we have m clusters, our search will allow us to find up to $m/2$ pairs of similar prototypes as our candidates to merge. Whilst the standard hierarchical algorithms reduce the number of elements from one partition, P_i, to the next one, P_{i-1}, by only one cluster, our search allows us to reduce the number of clusters by up to $m/2$.

Let $S = \{\langle i, \vec{x}(i)\rangle \mid i = 1, \ldots, N\}$, where $\vec{x}(i) \in \Omega_1 \times \cdots \times \Omega_n$, be a data set. For each continuous attribute $j \in \{1, \ldots, n\}$, let us suppose that we have a linguistic covering of the universe Ω_i. We can rewrite the attribute's value for each data point in S, $x_j(i)$ as a fuzzy set of words, namely, its linguistic description $des(x_j(i))$. For the sake of a simpler notation, let us identify $x_j(i)$ with $des(x_j(i))$. Finally, let us suppose that we have Sim the similarity measure between fuzzy objects defined in section 3. Then, having set a threshold $\sigma \in [0, 1]$ for the similarity, our linguistic hierarchical clustering algorithm is described as follows:

C= $\{\vec{x}(i) | \langle i, \vec{x}(i)\rangle \in S\}$
CHANGED=true
while CHANGED==true **do**
  NEWC= $\emptyset$
  CHANGED=false
  TO-MERGE= $\{i \mid \langle i, \vec{x}(i)\rangle \in$ C$\}$
  MERGED= $\emptyset$
  while TO-MERGE$\neq \emptyset$ **do**
    pick $i \in$ TO-MERGE
    pick $j \in$ TO-MERGE such that
      $Sim(\vec{x}(i), \vec{x}(j)) =$
      $\max_{k \in \text{TO-MERGE}} Sim(\vec{x}(i), \vec{x}(k))$
    if $Sim(\vec{x}(i), \vec{x}(j)) > \sigma$ **then**
      add $\vec{x}(i)[+]\vec{x}(j)$ to NEWC
      delete $\{i, j\}$ from TO-MERGE
      add $\{i, j\}$ to MERGED
      CHANGED=true
    else
      add $\vec{x}(i)$ to NEWC
      delete $\{i\}$ from TO-MERGE
      add $\{i\}$ to MERGED
    end if
  end while
  C=NEWC
end while
NEWC contains the final clusters for S

It may be worth noting that, in the case that new data points are obtained, there is no need to re-run the clustering from the beginning with the enhanced data set in order to update the final set of prototypes. Instead, it is enough to run the clustering algorithm with the union of the old prototypes and the new data points as our new data set.

For supervised learning where the data set is partitioned according to class, $S = \bigcup_{i=1}^{k} S_i$ where S_i is the set of data points in S of class C_i, the hierarchical linguistic clustering algorithm is applied to the subsets of data formed by each of the classes.

Having learnt a set of fuzzy prototypes describing the set S and given any tuple of values $\vec{x} \in \Omega_1 \times \cdots \times \Omega_n$, we can determine the support for it belonging to or being associated with a particular fuzzy prototype $P = \langle f_1, \ldots, f_n\rangle$ as follows:

$$supp(P|\vec{x}) = \prod_{i=1}^{n} \Pr(f_i | des(x_i))$$

In particular, for supervised learning, if we are given a tuple of values $\vec{x} \in \Omega_1 \times \cdots \times \Omega_n$ and we are asked to classify it, we can evaluate the support for each of the prototypes learned. Then, the vector is classified as belonging to the class associated with the prototype with highest support.

6 Application to a Model Classification Problem

To illustrate the above algorithm and its potential, let us consider a toy problem. In this problem, the data set consists of 916 data points in $[-1, 1] \times [-1, 1]$. The data set is divided in two classes, *legal* and *illegal*. If we consider two concentric circles, then the 345 points in the inner circle and the exterior circular crown are labelled as legal; the other

616 points in $[-1,1] \times [-1,1]$ are labelled as illegal. A plot of the legal points can be seen in Figure 1.

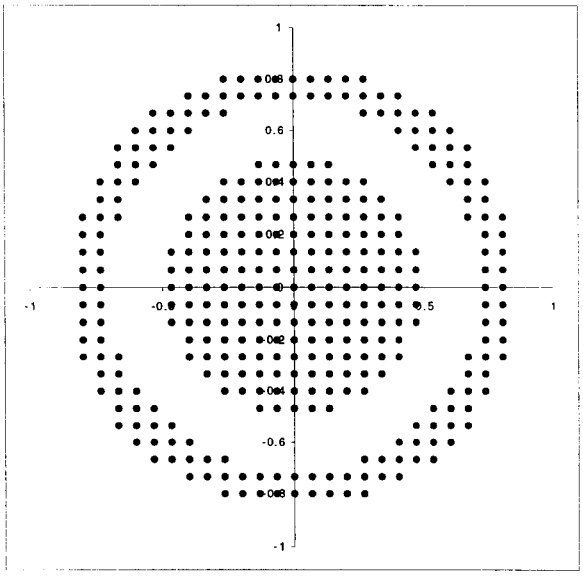

Figure 1: Legal points

As we have continuous attributes, we use linguistic coverings with 12 trapezoidal fuzzy sets uniformly distributed over each universe. If we set our similarity threshold to $\sigma = 0.5$, the clustering terminates with 36 prototypes to represent the *illegal* points and 31 to represent the *legal* ones. Even though there has been a considerable reduction in the number of clusters from our initial partitions of 616 and 345 data points respectively, we might want to merge more and reduce even further the number of final clusters. If we lower the similarity threshold to $\sigma = 0.4$, we allow clustering at one partition level higher and reduce the number of final prototypes to 17 for each class. Therefore, with only one more intermediate partition, we have approximately halved the number of final clusters for each class. This information is sumarised in Table 1.

Table 1: Number of Clustering Stages (NoCS) and Number of Prototypes (NoP) for each class

threshold	NoCS		NoP	
	illegal	legal	illegal	legal
$\sigma = 0.5$	6	5	36	31
$\sigma = 0.4$	7	6	17	17

Of course, we are interested in knowing how well these prototypes represent our original data. For this purpose, we ran a classifier through the same data and obtained a predictive accuracy of 95.4% for the threshold $\sigma = 0.5$ and of 93% for the threshold $\sigma = 0.4$. Obviously, the reduction in the number of prototypes gives some loss of predictive accuracy, although both results can be considered as good. Figure 2 is a plot of those points predicted *legal* according to prototypes learned with $\sigma = 0.4$.

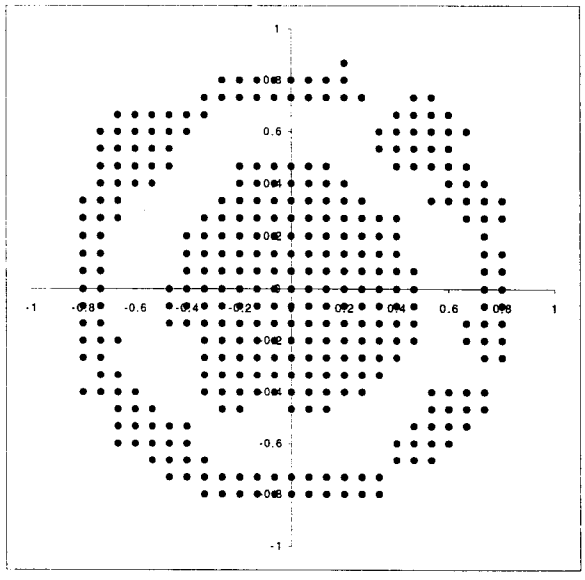

Figure 2: Learned legal points

7 Application to a Real-World Classification Problem

This database was taken from the UCI machine learning repository [9] and originates from a project carried out by the British Home Office Forensic Science Service Central Research Establishment on the identification of glass fragments found at crime scenes [4]. The study is motivated by the fact that in a criminal investigation the glass found at the scene of the crime can only be used as evidence if it is correctly identified. Glass fragments are divided into 7 possible classes, although the database only contains examples of six (there are no instances of class 4). These are:

1. Building windows—float processed
2. Building windows—non float processed

3. Vehicle windows—float processed
4. Vehicle windows—non float processed
5. Containers
6. Tableware
7. Headlamps

The classification is to be made on the basis of the following 9 attributes, relating to certain chemical properties of the glass (the unit measurement for attributes 2–9 is the weight percent of the corresponding oxide):

1. RI—refractive index
2. Na—sodium
3. Mg—magnesium
4. Al—aluminium
5. Si—silicon
6. K—potasium
7. Ca—calcium
8. Ba—barium
9. Fe—iron

The database, consisting of 214 instances, was split into a training and test set of 107 instances each in such a way that the instances of each class were divided equally between the two sets. A linguistic covering of 5 trapezoidal fuzzy sets was then defined for each attribute where a percentile approach was used to determine the exact position of the fuzzy sets (see Figure 3).

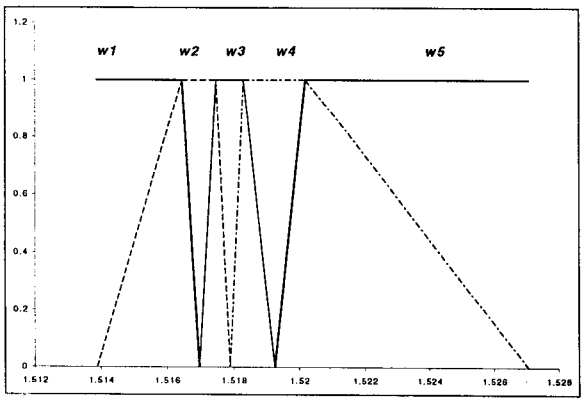

Figure 3: Linguistic covering for attribute 1, RI

The threshold parameter for the similarity was set at $\sigma = 0.5$ and the linguistic hierarchical clustering was applied to the data. The number of clustering stages (consecutive partitions of the data) used and the number of 'optimal' prototypes for each class can be seen in Table 2.

Table 2: Number of Clustering Stages (NoCS) and Number of Prototypes (NoP) for each class

CLASS	NoCS	NoP
1	5	5
2	5	4
3	2	4
5	3	2
6	2	1
7	4	5

Figure 4 shows the first attribute, RI, of the 5 prototypes obtained for the first class, *building windows—float processed.*

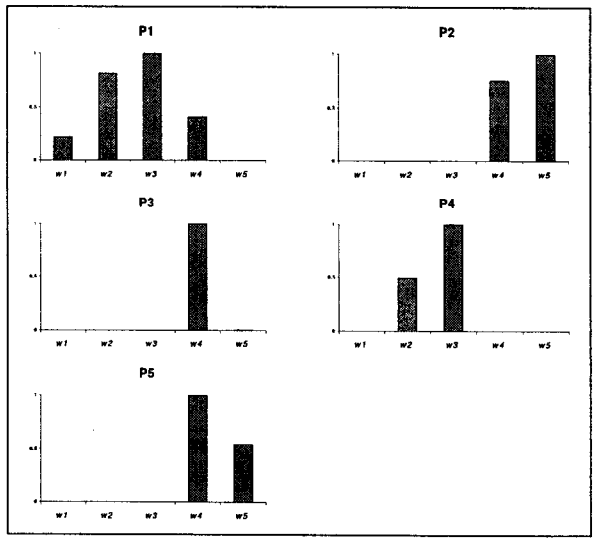

Figure 4: RI in the prototypes for class 1

The accuracy obtained using the prototypes for classification was 98% on the training set and 77.5% on the test set. This compares favourably with other learning algorithms. For instance, a previous mass assignment based prototype induction algorithm [4] gave an accuracy of 71% on the test set. Also, mass assignment ID3 [3] gave an accuracy of 68% on the test set and a neural network with topology 9–6–6 gave 72% on a smaller test set where the network was trained on 50% of the data and validated on 25% and tested on 25% [4].

8 Conclusions

A linguistic hierarchical clustering algorithm has been described for learning fuzzy prototypes to represent a data set as well as the number of prototypes needed. This algorithm incorporates ideas from mass assignment theory and similarity relations between fuzzy objects. The potential of the linguistic hierarchical clustering has been illustrated with both a toy example and a real world problem.

Acknowledgements

This work is partially funded by an E.P.S.R.C. studentship.

References

[1] J.F. Baldwin, J. Lawry, T.P. Martin, *Mass Assignment Theory of the Probability of Fuzzy Events*, Fuzzy Sets and Systems, Vol. 83, pp353–367, 1996.

[2] J.F. Baldwin, J. Lawry, T.P. Martin, *The Application of Generalised Fuzzy Rules to Machine Learning and Automated Knowledge Discovery*, International Journal of Uncertainty, Fuzzyness and Knowledge-Based Systems, Vol. 6, No. 5, pp459–487, 1998.

[3] J.F. Baldwin, J. Lawry, T.P. Martin, *Mass Assignment Based Induction of Decision Trees on Words*, Proceedings of IPMU98, Paris, France, 1998.

[4] J.F. Baldwin, J. Lawry, T.P. Martin, *A Mass Assignment Method for Prototype Induction*, International Journal of Intelligent Systems, Vol. 14, No. 10, pp1041–1070, 1999.

[5] B. S, Everitt, *Cluster Analysis*, Edward Arnold (Ed.), third edition, 1993.

[6] D. Dubois, H. Prade, *Unfair Coins and Necessity Measures: Towards a Possibilistic Interpretations of Histograms*, Fuzzy Sets and Systems, Vol, 10, pp15–20, 1979.

[7] D. Dubois, H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York, 1980.

[8] A. Tversky, *Features of similarity*, Psychological Review, Vol. 84, No. 4, pp327–353, 1977.

[9] WWW, *UCI Machine Learning Repository*, http://www.ics.uci.edu/~mlearn-/MLRepository.html.

[10] K. Yamada, *Probability-Possibility Transformation Based on Evidence Theory*, Proceedings of IFSA-NAFIPS'2001, Vancouver, Canada, 2001.

[11] L. A. Zadeh, *Fuzzy Sets and Information Granularity* in M. Gupta, R. Ragade, R. Yager (Eds), Advances in Fuzzy Set Theory and Applications, pp3–18, North-Holland, Amsterdam, 1979.

[12] L. A. Zadeh, *The concept of a linguistic variable and its applications to approximate reasoning*, Part I: Information Sciences, Vol. 8, pp199–249, 1975; Part II: Information Sciences, Vol. 8, pp301–357; Part III: Information Sciences, Vol. 9, pp43–80, 1976.