Dynamic Feature Selection in Incremental Hierarchical Clustering

Luis Talavera

Dept. Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya
Campus Nord, Mòdul C6, Jordi Girona 1-3, 08034 Barcelona, Spain
talavera@lsi.upc.es

Abstract. Feature selection has received a lot of attention in the machine learning community, but mainly under the supervised paradigm. In this work we study the potential benefits of feature selection in hierarchical clustering tasks. Particularly we address this problem in the context of incremental clustering, following the basic ideas of Gennari [8]. By using a simple implementation, we show that a feature selection scheme running in parallel with the learning process can improve the clustering task under the dimensions of accuracy, efficiency in learning, efficiency in prediction and comprehensibility.

1 Introduction

The performance of inductive learning algorithms heavily depends on the features used to describe the training data. As widely reported in the literature, most algorithms are known to degrade in performance when faced with features that are not useful for the task at hand. Ideally, one would like to provide the algorithm only with features containing useful information . However, there are many applications where experts make arbitrary choices or simply there are too many features to be processed by hand, so that automatic *feature selection* methods are needed.

Feature selection has received a lot of attention in the machine learning community as reflected in the huge number of works in the area as reviewed for instance in [1,2]. However, most of these works address the problem of supervised learning, and we can find only a few works devoted to unsupervised learning.

In this paper we address the particular problem of unsupervised feature selection in incremental hierarchical clustering. Given the nature of this sort of algorithms, we study a *dynamic* feature selection method that runs in parallel with learning. We follow the general guidelines proposed by Gennari [8] who proposed a dynamic feature selection method but only evaluated its merits with few datasets. We extend the testing procedure by adding some new dimensions to evaluate the benefits of feature selection and using several UCI datasets typically used for these purposes in supervised learning. Rather than develop an optimal method for feature selection, our work aims to explore the potential benefits of dynamic feature selection in clustering by using a more simple scheme that Gennari's but powerful and flexible enough to draw some interesting conclusions.

R. López de Mántaras, E. Plaza (Eds.): ECML 2000, LNAI 1810, pp. 392–403, 2000.

2 Supervised and Unsupervised Learning

The most common type of inductive learning problems are supervised *classification* problems. Given a set of labeled training examples the goal is to build a classification model that is able to correctly classify new, unseen instances. Evaluation of supervised learning systems is done by measuring the *accuracy* of the system. The system is provided with a separate set of instances usually called the test set, that is used to predict the label of each instance. Accuracy is estimated from the proportion of correct predictions over the total number of instances.

Unlike supervised learners, unsupervised algorithms do not have access to labeled examples. In unsupervised learning there are no target outputs associated with the inputs, and systems must resort to internal biases to decide which relationships should be represented in the output. This makes very difficult to define a widely accepted method for evaluating unsupervised learners and, particularly, clustering systems.

A first and widely used method for evaluating clustering systems is to compute predictive accuracy as is done for supervised classifiers. In order to apply this procedure a dataset with a known class structure must be used. The system is provided with a training set with the labels masked out from which a model is built. After learning, each cluster created by the system is labeled with the majority value for the class attribute and then the model is used to predict the label of instances in a test set. The resulting accuracy serves as a measure of how well the system has discovered the (known) underlying structure in the dataset. Alternatively, instead of using the system for prediction, after labeling the clusters, the proportion of incorrectly placed instances is computed as the accuracy of the system. This method is commonly used in statistics, since most statistical clustering approaches are not intended to make predictions.

A not so popular but not less interesting evaluation criterion is *flexible prediction* or *pattern completion* [6,11].Since in unsupervised learning there is no *a priori* target feature, it appears natural to consider that clustering or unsupervised systems in general, may support inference of any unknown feature value. A performance measure for this task is the average prediction accuracy over all the features. In order to compute this measure, each feature present in the data is masked out and the discovered model is used to predict its value from the information provided for the rest of features as if it was a “label”. The average of each individual accuracy for all the features is then taken as the overall predictive accuracy of the system. Note that flexible prediction is a much more complex task that label prediction, since multiple targets must be predicted from a single model. As an example, if you consider that a supervised classifier might be given the concept of “tiger” and then recognize this animal from observed features, from the viewpoint of flexible prediction, a clustering system should be able to predict, for example, that the animal is dangerous from the other observed features without even knowing the concept of “tiger” in advance.

Of course, other concerns may be particularly interesting in unsupervised learners, such as the *comprehensibility* of the results for humans, given that they discover completely new knowledge. In some cases users may be interested in the *descriptive* aspect of the results rather in their predictive power. Additionally, for particular applications there can be task-specific constrains. However, these aspects are usually very difficult to evaluate with numerical measures and often evaluation is very subjective.

3 Feature Selection in Hierarchical Clustering

At a conceptual level, the feature selection problem is similar for both supervised and unsupervised learners. Considering feature selection as a heuristic search in a space of feature subsets, any method, supervised or unsupervised, requires an starting point in the space, a search strategy, an evaluation function and a stopping criterion [1]. Under this view, unsupervised feature selection methods could be designed by adapting existing supervised methods and adding a few task-specific modifications. However, in practice, the adaptation of the evaluation function is not straightforward, since all the existing criteria rely on assessing how well a given feature subset discriminates among a set of predefined classes that are not available for unsupervised learners. In fact, the problem stems from a more general issue related to the performance task associated with each type of learning. As we have mentioned, in supervised learning, the predictive accuracy over class labels is a widely accepted performance task, so it is relatively easy to design evaluation functions. On the contrary there is a lack of a generally accepted performance task for clustering systems. For the rest of the paper we will focus on the three predictive tasks presented above: accuracy over labels, flexible prediction and comprehensibility.

Typically, the primary goal of feature selection is intended to make inductive learning algorithms more robust in the face of irrelevant features. There are a number of formal definitions of the relevance of features in the literature [9], although all of them are addressed to supervised tasks. There is no standard definition of irrelevance for flexible prediction tasks and, in fact, it is not clear if a system can built a clustering using a reduced feature set and still predict all the features originally present in the data. Therefore, for the rest of the paper we will resort to an intuitive notion of relevance, considering that a feature is relevant if it cannot be removed without loss of prediction accuracy of any kind.

There is an important factor related to the organization of the knowledge base in hierarchical clusterers. Commonly, hierarchical clusterings are *polythetic* classifiers, that is, they divide objects based on their values along multiple features. Particularly, they tend to use the full set of features at each node to decide how to classify a new object. Note that, while in *monothetic* classifiers such as decision trees, a redundant feature adds one additional test when classifying a new observation, in polythetic classifiers it adds a test for each node in the classification path. Clearly, improving performance may be a motivation for applying feature selection to clustering tasks, but not the only one. In general, the hierarchical organization, the polythetic nature of clusterings and the performance task determine several dimensions for evaluating the particular benefits of feature selection in conceptual clustering. Following [14,15] we can summarize these dimensions as follows:

- *Performance.* The set of features used in an inductive learning task is a powerful representational bias that determines the performance of a learning system. Irrelevant features may be particularly harmful in unsupervised systems, leading the system to form wrong patterns and having an impact in prediction that may be especially significant in a multiple inference task.
- *Efficiency in the learning task.* We have noted that hierarchical clusterings are polythetic classifiers. Since the decision of how to classify a new object has to be made along several nodes in the tree, the number of features present in the data strongly influences the complexity of the clustering process. If we apply feature selection to reduce this complexity, we should expect to obtain clusterings with at least similar performance that we would had obtained by using all the available features.
- *Efficiency in the performance task.* When using a hierarchical clustering to classify unobserved objects in order to infer unknown properties, the number of features has a strong influence in the complexity of the process in the same manner we have described above. Again, selecting an appropriate subset of features may reduce this complexity while maintaining the original performance level.
- *Comprehensibility of the results.* Clustering systems usually make use of all the available features at each node of the hierarchy. Reducing the number of features used in the clustering process allow to provide shorter cluster descriptions to the user. Short descriptions tend to be more readable and, hence more comprehensible.

4 Dynamic Feature Selection in Incremental Clustering

Typically, supervised feature selection methods are *static* in the sense that they are applied just once before the final induction task is carried out. The set of features obtained from the selection procedure is then fixed and never changes during learning. An alternative is to implement feature selection as a procedure that runs in parallel with learning. This approach allows the feature selection mechanism to *dynamically* adapt the set of selected features in the light of the knowledge gathered during the learning process. The dynamic feature selection procedure is then triggered at each *learning step*, that may differ from system to system. For example, in an incremental system, a learning step may be the incorporation of a new object, while in a batch agglomerative algorithm it may be a local merging operation. Interestingly, dynamic methods are the only methods that do not compromise the incremental operation of clustering systems that work in this way. Dynamic feature selection schemes may be very sensitive to wrong initial decisions biasing the system towards bad learning paths. However, potentially, they are a very attractive alternative since they can improve the clustering task on all the four dimensions presented in Section 3.

On the other hand, the hierarchical organization of the knowledge base in clustering allows to represent relatively complex descriptions of the environment at several levels of abstraction. By dividing the object space into local regions of variable generality, hierarchical clusterers provide a more expressive representation that flat clusterers. This property suggest that features that could be relevant at certain parts of the object space, might be useless at other regions. Thus, *local* feature selection methods that select different subsets of features for different nodes in the hierarchy appear particularly interesting, since they can be applied even when all the features in the data set are necessary for the clustering task. As polythetic classifiers, hierarchical clusterings may obtain great benefits from a local feature selection scheme even when none of the features in the original set is definitely removed. Dynamic feature selection naturally suggests to employ a local feature selection scheme, since they take local decisions at each learning step. It is worth noticing that local feature selection methods are more expensive than global ones. A local method must perform the same process that a global one as many times as nodes are in the tree. Moreover, if the are dynamically applied as well, one must take care of not employing very expensive procedures.

5 A Simple Dynamic Feature Selection Mechanism

In this section we will propose a simple implementation for a dynamic feature selection scheme. We implemented this method on the top of the well-known incremental clustering system Cobweb.

Cobweb is a hierarchical clustering system that constructs a tree from a sequence of objects. The system follows a strict *incremental* scheme, that is, it learns from each object in the sequence without reprocessing previously seen objects. An object is assumed to be a vector of nominal values V_{ij} along different features A_i. Cobweb employs *probabilistic concept* descriptions to represent the learned knowledge. In this sort of representation, in a cluster C_k, each feature value has an associated conditional probability $P(A_i = V_{ij} \mid C_k)$ reflecting the proportion of objects in C_k with the value V_{ij} along the feature A_i.

The strategy followed by Cobweb is summarized in Table 1. Given an object and a current hierarchical clustering, the system categorizes the object by sorting it through the hierarchy from the root node down to the leaves. At each level, the learning algorithm evaluates the quality of the new clustering resulting from placing the object in each of the existing clusters, and the quality resulting from creating a new cluster covering the new object. In addition, the algorithm considers two more actions that can restructure the hierarchy in order to improve its quality. *Merging* attempts to combine the two sibling clusters which were identified as the two best hosts for the new object; *splitting* can replace the best host and promote its children to the next higher level. The option that yields the high quality score is selected and the procedure is recursed, considering the best host as the root in the recursive call. The recursion ends when a leaf containing only the new object is created.

Table 1. The control strategy of Cobweb

```
Function Cobweb(object,root)
    1) Incorporate object into the root cluster.
    2) If root is a leaf then
           return expanded leaf with the object.
       else choose the best of the following operators:
           a) Incorporate the object into the best host
           b) Create a new disjunct based on the object
           c) Merge the two best hosts
           d) Split the best host
    3) If a), c) or d) recurse on the chosen host.
```

In order to choose among the four available operators, Cobweb uses a cluster quality function called *category utility* defined for a partition $P = \{C_1, C_2, ..., C_n\}$ of n clusters as

$$\frac{\sum_k P(C_k) \sum_i \sum_j [P(A_i = V_{ij} \mid C_k)^2 - P(A_i = V_{ij})^2]}{n} \tag{1}$$

This function measures how much a partition P promotes inference and rewards clusters C_k that increase the predictability of feature values within C_k. By using this metric, the system should be biased towards the construction of clusters allowing accurate predictions along any unobserved features.

As in supervised feature selection, feature selection in clustering can be done by using the so-called *filter* or *wrapper* models [9]. Briefly, filter models are independent of the induction algorithm that will use their output and they employ some metric dependent on intrinsic properties of the data. Typically, they measure the correlation of each feature with the class label by using distance, information or dependence measures. Obviously, the absence of class labels makes infeasible to compute these sort of measures in unsupervised learning and, therefore, alternative measures not using class information need to be defined.

On the other hand, in the wrapper model, the feature selection algorithm works as a wrapper around the induction algorithm. Alternative feature subsets are evaluated by using the induction algorithm as a black box over the training data in order to obtain an estimate of future performance. Usually, performance is estimated by measuring the predictive accuracy over class labels. Note that unsupervised learners cannot use these methods in the label prediction performance task, since they have no access to the labels during learning. Wrappers can be used for flexible prediction, albeit at the price of a high computational cost to estimate accuracy over the full feature set.

We propose a filter method of feature selection based on an *ordering* scheme. A weight is individually computed for each feature and features are ordered

Table 2. A method for feature selection based on an ordering scheme

```
Let $\mathcal{A}$ be a set of features
Let $\tau$ be the feature selection threshold
**Function** select_features($\mathcal{A}$,$\tau$)
    compute_feature_weights($\mathcal{A}$)
    $max\_w = max\{weight(A_i) \mid A_i \in \mathcal{A}\}$
    **return** $\{A_i \mid weight(A_i) \geq max\_w \times \tau\}$
```

according to these weights. We define a *feature selection threshold* τ in the [0,1] range such that the weight required for a feature to be selected is higher for higher τ values. Our method uses the maximum computed weight as a baseline to determine which features are selected as shown in Table 2. Note that, if we assume relevances to be positive, when $\tau = 0$ there is no feature selection at all, so reducing the original algorithm to a special case of our approach.

This method can be easily incorporated into COBWEB by slightly modifying the control strategy showed in Table 1. First, we need to add an additional step between steps 2 and 3 of the existing algorithm. In this step a call to the *select_features* function is performed, obtaining a subset of relevant features to be stored in the current root node. Second, at each classification step, the computation of the quality function must be modified in such a way that only the subset of relevant features stored in the current root node is used.

The weighting function we use is the one proposed by Gennari [8] in the context of his CLASSIT system, an extension of COBWEB to deal with numeric features. Gennari refers to this measure as *salience*. He defines the relative salience of a feature as its contribution to category utility (see equation 1) in a clustering. More formally, for a given feature A_i, salience is defined as follows:

$$\frac{\sum_k P(C_k) \sum_j [P(A_i = V_{ij} \mid C_k)^2 - P(A_i = V_{ij})^2]}{n} \tag{2}$$

6 Experiments

In order to evaluate dynamic feature selection, we ran experiments on six datasets from the UCI repository: cleveland, crx, horse colic, hypothyroid, pima and wisconsin diagnostic breast cancer. Our aim was to test the potential of feature selection regarding the dimensions presented in Section 3. As regards to performance, we performed separated experiments for label and flexible prediction. In order to evaluate the efficiency of the learning and prediction processes, we computed the *average number of feature tests* needed to sort the instances in the training or testing set. This number is calculated by summing the total number of features involved in evaluating the category utility metric for the different clustering choices. For instance, if an observation is being clustered in a root node with three children and using a subset of n features, we need to perform $3n$ feature tests to evaluate the CU of incorporating the observation to each of the siblings. In learning, additional feature tests are needed to evaluate creating a new cluster, merging and splitting. We think that this way of measuring efficiency give us a better empirical approximation of the complexity of the clustering process than, for instance, the average of features per node. On the other hand, we use this later measure as a measure of *comprehensibility* of the obtained clusterings, since fewer features per node indicate simpler cluster descriptions.

Table 3. Accuracy in label prediction, average number of tests in learning and prediction, features per node and number of nodes for several datasets and τ values

Dataset	τ	Accuracy	Tests learning	Tests pred	Feat./node	Nodes
cleve	n/a	74.73 (5.05)	1619.94 (42.94)	720.40 (45.02)	13.00 (0.00)	107.60 (2.91)
	0.1	75.27 (5.06)	1161.23 (96.60)	450.81 (59.67)	5.58 (0.09)	126.30 (2.26)
	0.2	76.04 (4.60)	928.64 (117.88)	363.80 (45.76)	5.22 (0.17)	129.30 (4.08)
	0.3	75.93 (3.17)	712.32 (55.35)	246.75 (24.73)	4.90 (0.12)	134.30 (2.71)
	0.4	74.18 (3.08)	531.78 (40.33)	190.97 (14.89)	4.53 (0.14)	135.60 (5.58)
	0.5	73.19 (3.56)	396.53 (18.12)	130.61 (7.05)	3.99 (0.20)	141.80 (2.74)
crx	n/a	80.24 (2.89)	2226.49 (59.25)	950.22 (35.62)	15.00 (0.00)	255.30 (5.96)
	0.1	78.74 (4.57)	1070.76 (66.12)	388.48 (31.81)	4.56 (0.08)	297.80 (7.44)
	0.2	79.86 (2.93)	852.89 (62.22)	286.32 (23.67)	4.20 (0.13)	302.60 (6.92)
	0.3	80.87 (3.35)	653.60 (33.76)	211.09 (15.19)	3.91 (0.15)	307.10 (4.77)
	0.4	78.89 (3.28)	486.00 (25.49)	156.69 (8.43)	3.69 (0.10)	314.00 (5.58)
	0.5	78.41 (2.75)	392.04 (23.29)	121.83 (8.00)	3.35 (0.08)	327.40 (5.13)
horse	n/a	74.23 (4.60)	3108.48 (79.31)	1371.85 (126.94)	22.00 (0.00)	124.30 (3.71)
	0.1	72.52 (2.66)	2011.35 (78.56)	797.53 (54.58)	9.18 (0.23)	149.40 (1.96)
	0.2	75.95 (3.85)	1548.66 (102.93)	602.92 (49.33)	8.52 (0.26)	150.60 (3.34)
	0.3	75.68 (3.89)	1151.87 (45.39)	420.08 (33.19)	7.95 (0.16)	158.40 (3.98)
	0.4	72.16 (3.05)	882.24 (86.90)	304.25 (24.66)	7.33 (0.17)	161.30 (3.33)
	0.5	72.61 (3.30)	612.11 (31.35)	200.78 (17.39)	6.54 (0.21)	170.50 (2.99)
hypo	n/a	97.65 (0.48)	8448.86 (248.21)	3952.57 (234.33)	25.00 (0.00)	1825.50 (13.95)
	0.1	97.65 (0.34)	3867.14 (229.60)	2024.41 (243.95)	18.46 (0.18)	1912.00 (14.45)
	0.2	97.54 (0.43)	3769.29 (311.04)	2035.68 (327.15)	18.33 (0.24)	1928.40 (14.03)
	0.3	97.61 (0.44)	3467.92 (220.02)	1873.79 (338.49)	18.13 (0.25)	1954.10 (11.51)
	0.4	97.47 (0.46)	3407.59 (250.65)	1868.41 (321.50)	17.89 (0.21)	1979.80 (10.67)
	0.5	97.46 (0.44)	3183.58 (343.08)	1643.02 (348.78)	17.60 (0.23)	2011.50 (8.66)
pima	n/a	65.11 (2.62)	1135.25 (21.92)	470.92 (15.74)	8.00 (0.00)	321.80 (7.32)
	0.1	64.94 (2.70)	723.84 (51.84)	256.16 (24.38)	3.61 (0.08)	347.70 (5.85)
	0.2	66.06 (2.94)	571.35 (45.10)	195.13 (19.45)	3.35 (0.08)	358.80 (5.49)
	0.3	64.85 (3.26)	448.13 (22.87)	150.23 (10.38)	3.24 (0.07)	365.70 (6.93)
	0.4	66.32 (2.10)	365.34 (30.23)	117.43 (8.09)	3.10 (0.10)	372.10 (6.28)
	0.5	65.93 (3.40)	274.59 (16.04)	89.56 (4.89)	2.93 (0.07)	384.40 (7.90)
wdbc	n/a	91.93 (1.55)	4287.82 (79.96)	1881.39 (70.54)	30.00 (0.00)	198.00 (6.73)
	0.1	91.93 (1.80)	2839.01 (196.73)	1123.98 (62.42)	12.67 (0.11)	235.10 (4.18)
	0.2	92.57 (1.20)	2249.09 (70.47)	864.46 (57.95)	11.68 (0.30)	240.30 (5.50)
	0.3	91.17 (2.69)	1858.66 (95.20)	663.93 (75.65)	10.97 (0.29)	247.70 (4.52)
	0.4	90.99 (2.34)	1362.12 (93.71)	472.08 (36.45)	9.95 (0.25)	256.00 (5.33)
	0.5	90.99 (1.66)	1013.66 (40.44)	335.47 (21.03)	8.91 (0.23)	266.70 (8.25)

In all the experiments, we used a 70% of the instances for training and a 30% for testing. All the results shown are averages over 20 independent runs using random splits. For each dataset, several values of the τ parameter were used to gain some insight into the effect of different degrees of selection on the performance of the system. The system never had access to the labels neither in label nor flexible prediction, although in the later case they were used for evaluation purposes. In the case of flexible prediction, accuracy is always computed as the overall accuracy over all the original features in the data, regardless of the features removed during the feature selection process. Since the CU measure is only applicable to nominal features, all numeric features were discretized.

Table 3 shows the results for the label prediction performance task. At a glance, we can observe that in all datasets accuracy can be maintained or improved while reducing the number of features per node used to an average of the 40% of the original number of features. As expected, this reduction implies an improvement in the efficiency of the system in both learning and prediction. In average, dynamic feature selection provides an efficiency improvement of about the 50% in learning and prediction. Note that the feature selection scheme produces changes in the structure of the hierarchies created by increasing the number of inner nodes. This increment reduces partially the efficiency gains obtained with the removal of features, since the complexity of sorting an instance depends on the depth of the hierarchy as well.

As a conclusion, we can say that dynamic feature selection can provide benefits in the clustering task along the four proposed dimensions for evaluation as regards the label prediction task. Our results for this task as regards efficiency agree with the results of Gennari[8]. The potential for creating accurate clusterings for this task is also shown in [13], although using a more classical preprocessing approach. As we have noted, obtaining such results with a dynamic incremental scheme is quite impressive given the greedy nature of the method.

Table 4 shows the results for the flexible prediction task. Again, the feature selection scheme is able of creating simpler clusterings without hurting accuracy. These results are even more remarkable than the previous ones, given the multiple inference task now required for the clusterings. Efficiency in prediction is improved in a similar amount that for label prediction. Obviously, the improvements in the efficiency in learning and in the number of features per node remain the same, since the learning task is identical for both performance tasks.

Therefore, we can conclude also that dynamic feature selection is able to improve the clustering task along the four evaluation dimensions as regards flexible prediction. We have obtained similar results by using a postprocessing approach [15]. Although such an approach is less prone to bad decisions while simplifying the hierarchy, it cannot improve the efficiency of learning as dynamic feature selection does.

7 Related Work

The idea of focusing on particular features in incremental unsupervised learning can be traced back to early influential work by Kolodner [10] on Cyrus and Lebowitz [12] on Unimem. As we have pointed out, Gennari [8] proposed a more general and principled mechanism that inspired this work. Fisher *et al* [5] adapted

Table 4. Accuracy in flexible prediction, average number of test in learning and prediction, features per node and number of nodes for several datasets and τ values

Dataset	τ	Accuracy	Tests learning	Avg. tests pred	Feat./node	Nodes
cleve	n/a	48.78 (1.93)	1619.94 (42.94)	8631.64 (502.74)	13.00 (0.00)	107.60 (2.91)
	0.1	49.10 (2.44)	1161.23 (96.60)	5417.78 (719.98)	5.58 (0.09)	126.30 (2.26)
	0.2	49.04 (1.65)	928.64 (117.88)	4342.31 (515.53)	5.22 (0.17)	129.30 (4.08)
	0.3	48.83 (1.64)	712.32 (55.35)	3019.06 (295.23)	4.90 (0.12)	134.30 (2.71)
	0.4	48.86 (1.87)	531.78 (40.33)	2400.76 (171.30)	4.53 (0.14)	135.60 (5.58)
	0.5	50.23 (2.52)	396.53 (18.12)	1750.43 (84.59)	3.99 (0.20)	141.80 (2.74)
	0.6	49.88 (1.12)	312.49 (33.26)	1440.42 (86.37)	3.72 (0.15)	145.60 (5.21)
crx	n/a	60.72 (1.23)	2226.49 (59.25)	13302.06 (491.47)	15.00 (0.00)	255.30 (5.96)
	0.1	60.74 (1.28)	1070.76 (66.12)	5427.68 (442.28)	4.56 (0.08)	297.80 (7.44)
	0.2	61.25 (0.88)	852.89 (62.22)	4028.86 (307.44)	4.20 (0.13)	302.60 (6.92)
	0.3	61.17 (1.20)	653.60 (33.76)	3059.85 (202.27)	3.91 (0.15)	307.10 (4.77)
	0.4	60.99 (1.27)	486.00 (25.49)	2353.99 (81.73)	3.69 (0.10)	314.00 (5.58)
	0.5	60.80 (0.92)	392.04 (23.29)	1930.33 (85.65)	3.35 (0.08)	327.40 (5.13)
horse	n/a	59.17 (1.05)	3108.48 (79.31)	28738.65 (2643.82)	22.00 (0.00)	124.30 (3.71)
	0.1	59.94 (1.21)	2011.35 (78.56)	16762.54 (1159.86)	9.18 (0.23)	149.40 (1.96)
	0.2	59.30 (1.85)	1548.66 (102.93)	12618.22 (983.83)	8.52 (0.26)	150.60 (3.34)
	0.3	58.89 (0.93)	1151.87 (45.39)	8865.25 (646.47)	7.95 (0.16)	158.40 (3.98)
	0.4	58.12 (1.17)	882.24 (86.90)	6562.92 (516.68)	7.33 (0.17)	161.30 (3.33)
	0.5	58.10 (0.97)	612.11 (31.35)	4473.22 (320.96)	6.54 (0.21)	170.50 (2.99)
hypo	n/a	83.05 (1.05)	8448.86 (248.21)	87296.71 (5996.60)	25.00 (0.00)	1825.50 (13.95)
	0.1	84.71 (0.51)	3867.14 (229.60)	45223.19 (5571.86)	18.46 (0.18)	1912.00 (14.45)
	0.2	85.12 (0.83)	3769.29 (311.04)	46351.35 (7007.69)	18.33 (0.24)	1928.40 (14.03)
	0.3	84.44 (0.90)	3467.92 (220.02)	41475.52 (7199.13)	18.13 (0.25)	1954.10 (11.51)
	0.4	84.59 (0.67)	3407.59 (250.65)	41225.04 (6659.03)	17.89 (0.21)	1979.80 (10.67)
	0.5	83.69 (1.08)	3183.58 (343.08)	36539.31 (7635.42)	17.60 (0.23)	2011.50 (8.66)
pima	n/a	45.61 (1.27)	1135.25 (21.92)	3299.04 (104.96)	8.00 (0.00)	321.80 (7.32)
	0.1	45.37 (1.35)	723.84 (51.84)	1801.95 (160.45)	3.61 (0.08)	347.70 (5.85)
	0.2	45.82 (1.46)	571.35 (45.10)	1379.61 (126.57)	3.35 (0.08)	358.80 (5.49)
	0.3	45.85 (1.33)	448.13 (22.87)	1107.40 (81.37)	3.24 (0.07)	365.70 (6.93)
	0.4	45.98 (0.99)	365.34 (30.23)	935.74 (39.89)	3.10 (0.10)	372.10 (6.28)
	0.5	46.20 (1.35)	274.59 (16.04)	794.64 (22.02)	2.93 (0.07)	384.40 (7.90)
	0.6	46.34 (1.32)	232.07 (13.29)	733.15 (24.28)	2.79 (0.06)	393.70 (7.60)
	0.7	46.38 (1.28)	191.36 (6.65)	707.61 (24.26)	2.56 (0.09)	398.60 (8.98)
wdbc	n/a	62.34 (1.57)	4287.82 (79.96)	54604.95 (2023.58)	30.00 (0.00)	198.00 (6.73)
	0.1	62.96 (0.99)	2839.01 (196.73)	32584.53 (1781.38)	12.67 (0.11)	235.10 (4.18)
	0.2	62.89 (0.64)	2249.09 (70.47)	25051.45 (1661.10)	11.68 (0.30)	240.30 (5.50)
	0.3	62.41 (0.92)	1858.66 (95.20)	19308.05 (2229.57)	10.97 (0.29)	247.70 (4.52)
	0.4	61.90 (1.04)	1362.12 (93.71)	13703.37 (1061.79)	9.95 (0.25)	256.00 (5.33)
	0.5	61.43 (1.49)	1013.66 (40.44)	9833.95 (582.14)	8.91 (0.23)	266.70 (8.25)

Gennari's procedure to a diagnosis task, where the intent was to minimize the number of probes necessary to diagnose a fault.

As in supervised learning, preprocessing approaches are more common as in [3], [4] or [13]. However, neither of these works have been extensively evaluated along all the dimensions proposed here. As regard the flexible prediction task, the only existing work is [16] with a weak evaluation and our own work in preprocessing and postprocessing methods [14,15]. Although the later works used different data sets to those that were used in this paper, at first sight, dynamic feature selection appears to be a good alternative to these methods.

8 Concluding Remarks

Feature selection methods have shown successful in supervised approaches and we have shown that they could be also useful in incremental hierarchical clustering despite the difficulties posed by unsupervised settings. Besides the traditional aim of increasing accuracy, we have proposed other dimensions for evaluating this benefits mainly concerned with efficiency and comprehensibility. In addition, hierarchical clusterings suggest a local feature selection scheme for each node in the hierarchy. Moreover, given that unsupervised systems support inference on more than one single dimension, we have shown the benefits of feature selection in both classical label prediction and flexible prediction, the later involving prediction of all the features in the data.

To maintain the incremental nature of the system we have applied a dynamic feature selection scheme that runs parallel to the clustering process instead of being a preprocessing step, as typically done in supervised learning. Results show that this mechanism can improve efficiency in learning, efficiency in prediction and comprehensibility while maintaining or improving performance in prediction.

All of these results have been obtained with a simple and rough implementation that can be clearly improved, although it has served well for the purpose of this study. Surely, a smarter feature selection method can be designed, ideally without having to set any thresholds. Gennari's original method is one possible alternative to test. Additionally, since the salience measure is derived from the objective function used in clustering (CU in this case), it would be interesting to test alternative objective functions to CU to see if they are better candidates not only for evaluating clusters but for evaluating features as well.

Finally, Fisher's work [7] suggests further implications about the relationship between feature selection and using feature frontiers for prediction. If a feature can be predicted accurately at a certain node without descending deeper into the hierarchy, this feature is not informative to discriminate between descendant nodes. We have noted the importance of the structure of the hierarchy only at a cursory level, but future work should explore this issue by including additional dimensions for cost assessment such as the branching factor or the depth of the hierarchies.

Acknowledgements

The author thanks one anonymous referee for providing important pointers to relevant work and its implications on feature selection. This work is partially supported by UPC grant PR99-09.

References

1. A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997. 392, 394
2. M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(3), 1997. 392
3. M. Dash, H. Liu, and J. Yao. Dimensionality reduction for unsupervised data. In *Ninth IEEE International Conference on Tools with AI, ICTAI'97*, 1997. 401

4. M. Devaney and A. Ram. Efficient feature selection in conceptual clustering. In *Machine Learning: Proceedings of the Fourteenth Intern ational Conference*, Nashville, TN, 1997. 401
5. D. Fisher, L. Xu, J. Carnes, Y. Reich, S. Fenves, J. Chen, R. Shiavi, G. Biswas, and J. Weinberg. Applying ai clustering to engineering tasks. *IEEE Expert*, 8:51–60, 1993. 400
6. D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987. 393
7. D. H. Fisher. Iterative optimization and simplification of hierarchical clusterings. *Journal of Artificial Intelligence Research*, 4:147–179, 1996. 402
8. J. H. Gennari. Concept formation and attention. In *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, pages 724–728, Irvine,CA, 1991. Lawrence Erlbaum Associates. 392, 398, 400
9. R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997. 394, 397
10. J. L. Kolodner. Reconstructive memory: A computer model. *Cognitive Science*, 7:281–328, 1983. 400
11. P. Langley. *Elements of machine learning*. Morgan Kaufmann, San Francisco, CA, 1995. 393
12. M. Lebowitz. Experiments with incremental concept formation: UNIMEM. *Machine Learning*, 2:103–138, 1987. 400
13. L. Talavera. Dependency-based feature selection for symbolic clustering. *Intelligent Data Analysis*. To appear. 400, 401
14. L. Talavera. Feature selection as a preprocessing step for hierarchical clustering. In *Proceedings of the Sixteenth International Conference on Machine Learning*, Bled, Slovenia, 1999. Morgan Kaufmann. 395, 401
15. L. Talavera. Feature selection as retrospective pruning in hierarchical clustering. In *Third International Symposium on Intelligent Data Analysis, IDA 99*, volume 1642 of *Lecture Notes in Computer Science*, Amsterdam, The Netherlands, 1999. Springer Verlag. 395, 400, 401
16. J. J. Furtado Vasco. Determining property relevance in concept formation by computing correlation between properties. In *Proceedings of the Tenth European Conference on Machine Learning, ECML98*, volume 1398 of *Lecture Notes in Artificial Intelligence*, pages 310–315, Chemnitz, Germany, 1998. Springer Verlag. 401