Finding Attribute Selection Measures by Computing Loss of Information and Ambiguity for Data Clustering

Mr. Mane S.G.[1], Mr. Powar R.V.[2]

[1]*(Senior Lecturer in Computer Engineering, Institute of Civil and Rural Engineering, D.T.E. Mumbai, India)*
[2]*(Senior Lecturer in Computer Engineering, Institute of Civil and Rural Engineering, D.T.E. Mumbai, India)*

ABSTRACT: *In this paper, we propose a method for hierarchical clustering based by using measures for attribute selection and data partitioning algorithms after selecting the proper attribute. We have find these attributes by computing the loss of information and ambiguity. In this we have generated the decision tree (unsupervised) which will maintain the data available at each node , name of the attribute selected for partitioning available data, and rule used to partition data. We present two different measures for selecting the most appropriate attribute to be used for splitting the data at every branching node (or decision node), and two different algorithms for splitting the data at each decision node. At the last we have shown the performance of these measures and partitioning algorithms by using one sample labeled database.*

Index Terms
Unsupervised decision tree, entropy, data set segmentation, valley detection.

1. INTRODUCTION

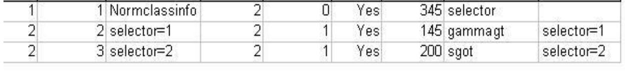
One major advantage of the decision tree is its interpretability, i.e., the decision can be represented in terms of a rule set. The branching decision at each node is determined by the value of a certain attribute or combination of attributes, and the choice of the attribute(s) is based on a certain splitting criterion that is consistent with the objective of the classification process. Each leaf node of the tree represents a class and is interpreted by the path from the root node to the leaf node in terms of a rule.

Unsupervised decision trees are structurally similar to hierarchical clustering methods. Algorithms for hierarchical clustering are generally of two kinds, namely, top-down and bottom-up. In the bottom-up algorithms for hierarchical clustering, each data point is considered to be a separate cluster and then these are progressively combined depending on certain criteria to generate the hierarchy. The structure generated by this process commonly referred to as a dendrogram. Different distance measures give rise to different cluster structures at the end of the algorithm. In top-down hierarchical clustering algorithms, to begin with, all data points are considered to belong to the same cluster. The data set (consisting of set of patterns) is then divided into a certain number clusters at a coarse level. Then, each of these coarse clusters is further segmented into finer levels in the subsequent levels until a stopping criterion is satisfied.

2. OVERALL ALGORITHM AND CLUSTER INFORMATION STORING

As we are going to cluster the data in hierarchical fashion , we have to assign the whole data (initial data) to the root node of the tree that we are going to construct. After that the task of the finding the most important field for data partitioning starts. This can be done by any one of the two measures. After finding the important fields, if that field if categorical then partition the data into no. of cluster into no. of distinct values of that selected attribute appears. If selected attribute is numeric then we have to use valley detection algorithm for data partitioning. Another type of algorithm for data partitioning after selecting the important attribute is binary partitioning. Ending criteria used in this algorithm may be number of levels , size of the leaf node or both. The overall algorithm for data clustering is as below,

As we go on finding the important attribute and partition, we need to store the information of each and every node that generated for future use . i.e. for generating rules, construction of tree and analysis. Figure shows sample of structure which is used to store the clustering data.



elno	nodeno	Data	nochild	Parentno	ex	size	fieldsel	criteria
1	1	Normclassinfo	2	0	Yes	345	selector	
2	2	selector=1	2	1	Yes	145	gammagt	selector=1
2	3	selector=2	2	1	Yes	200	sgot	selector=2

Fig. 1 database structure to store cluster information.

1. *Level No.* is used to store the level no. of the generated node or cluster.
2. *Node No.* is used to give unique number to every node.
3. *Data* is used to store the rule generated for the generated node.
4. *Nochild* describes the no. of child node for that node.
5. *ParentNo* gives the number of the parent node.
6. *Ex.* Indicates whether this node is partitioned or not.
7. *Size*. Size is the no. of data records available at this node.
8. *Fieldsel.* Is used to store the selected field for partitioning data.
9. *Critera.* Is used to use the criteria used to partition the data.

3. MEASURES FOR FINDING THE IMPORTANT ATTRIBUTE

At each node of the unsupervised decision tree, we select an attribute in such a way that the inhomogeneity of the data set is maximum with respect to that attribute. We used various entropy measures in selecting the attribute at each level of the tree. We describe four different measures of inhomogeneity of the data set with respect to an attribute.

Measure I

Let μ_{ij}^{a} be the degree of similarity of two data points x_i and x_j in the data set available at a given node. We define as ,

$$\mu_{ij} = g\left(1 - \frac{d_{ij}}{d_{\max}}\right) \qquad (1)$$

where d_{ij} is the distance d(x_i , x_j) between the data points x_I and x_j , d_{max} is the maximum of all interpoint distances, and the function g(.) is a monotonically nondecreasing function,

$$g(x) = \begin{cases} x & \text{for } 0 <= x <= 1 \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

Equation (2) indicates that $\hat{\mu}_{ij}^{a}$ lies between 0-1 for all i and j. The measure of importance for an attribute fa is computed as,

$$Ha = -\left(\mu_{ij}\left(1-\mu_{ij}^{a}\right) + \mu_{ij}^{a}\left(1-\mu_{ij}\right) \right) \qquad (3)$$

Measure II.

Instead of computing the loss of information when an attribute is dropped from the set of attributes, we can also compute ambiguity when an attribute is considered alone. In that case, the importance of an attribute is computed as

$$\hat{H}a = -\hat{\mu}_{ij}^{a}(1-\hat{\mu}_{ij}^{a}) \qquad (4)$$

$\hat{\mu}_{ij}^{a}$ Indicates the degree of similarity between two data points x_i and x_j in terms of the attribute f_a only.

3.1 Calculating Euclidean Distance between data points

The Euclidean distance between two Points/objects/items in dataset , defined by point x and y is defined by equation ,

Euclidean Distance

$$d(x,y) = (|x_1-y_1|^2 + |x_2-y_2|^2 + |x_3-y_3|^2 + \ldots\ldots\ldots + |x_n-y_n|^2)^{1/2} \qquad (5)$$

Where | z | represents the absolute value of z , x is the first data point , y is the second data point , n is the number of characteristics or attributes in data mining terminology or fields in database terminology. The Euclidean distance works well for continuous type attribute but not for the combination of the continuous and categorical type of attribute . So we use the alternative distance calculating function called Heterogeneous Euclidean-Overlap Metric (HEOM) for this purpose. In this function one approach that has been used is to use the overlap metric for nominal (categorical) and normalized Euclidean distance calculation for linear (continues) attribute.

As per this method distance between two values x and y of a given attribute a as ,

$$f = \arg\max_{\forall fa \in F} \left\{ H_a(D_1^{c_1^a}) + H_a(D_2^{c_1^a}) - H_a(D) \right\}$$

$$D_a(x,y) = \begin{cases} 1 \text{ if x or y is unknown , else} \\ \text{overlap(x,y) if a is nominal} \\ \text{else } \mathrm{diff}_a(x,y) \end{cases} \qquad (6)$$

Unknown attribute values are handled by returning an attribute distance 1 (i.e. a maximum distance) if eighter of the attribute value is unknown. The function overlap and range-normalized difference rn_diff are defined as,

$$\text{Overlap}(x,y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{otherwise} \end{cases} \qquad (7)$$

$$\text{m\_diff}_a(x,y) = |x-y| / \text{range}_a$$

The value rangea is used to normalize the attribute , and is defined as ,

range_a=maxa-mina

Where maxa and mina are the maximum and minimum values respectively. The overall distance between input vectors x and y is given by,

$$HOEM = \sum \sqrt{d_a(x_a, y_a)^2}$$

For a=0 to m, where m are the no. of attributes.

4. DATA PARTITIONING ALGORITHMS

In this section, we present two different algorithms for splitting the data based on a single attribute.

4.1 Data Set Partitioning Based on Valley Detection

1. For categorical (or nominal) attributes, we partition the data set into a number of segments that is equal to the number of different attribute values in the data set.
2. For numerical attributes, we compute the histogram of the data along the attribute dimension.
3. We then determine the valley points of the smoothed histogram by computing the change in direction of the gradient.
4. We evaluate each valley point as

$$e_i = \frac{\min\{q_i - v_i, q_{i+1} - v_i\}}{1 + \lambda v_i}$$

where vi is the height of the valley, and qi and q_{i+1} are heights of the peaks on either side of the valley. Based on the evaluated score ei, we consider the top k - 1 valleys (for k-ary tree) as potential cut-off points (if the number of detected valleys is less than k, then we consider all detected valleys).

5. Let c_1, c_2, _ _ _ c_{k-1} be the cut-off points. For every pair (ci,c_{i+1}) of consecutive cut-off points, the data records for which the value of the attribute falls between ci and c_{i+1} constitute the ith segment. If the number of data records in a segment is less than a certain predefined count, then we merge the records into the nearest segment

4.2. Problems in valley detection algorithm

1. If no. of valleys are more than k-1 (predefined value) , then we evaluate each valley by using peak values of the valley. After evaluating the valleys if n. of valley selected are more than k-1 due to equal evaluation we have to consider more than k-1 valleys in this condition.
2. If no. of valleys are less than k-1 but some valleys are such that their peaks are so small as compared to other peaks. In this condition we can consider the valleys which has any one peak (left or right) has value greater than the average of data points.

PeakLimit can be calculated as ,

$$\text{PeakLim} = \frac{\text{No. of data records available}}{\text{No. of bins in histogram}} \qquad (8)$$

4.3. Binary Partitioning of Data

Let H(D) represent the inhomogenity or information content of the data set D with respect to some attribute fa. If we perform a binary partition of the data at a certain cutoff point c into two subsets D_1 and D_2, then the resulting change in information is ,

$$\nabla H(c, D) = H(D_1^c) + H(D_2^c) - H(D) \qquad (9)$$

if maxc ∇H(c,D) >0 , then we can find cut-off point c_0 as ,

c_0 =argmaxc{∇H(c,D) }

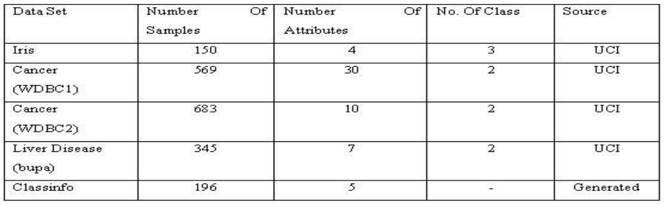
Note that, in this partitioning process, we always split the data set into two segments with O(n) complexity where n is the number of data points. However, it is possible to find a k partition using brute force search with O(n $^{(k-1)}$) complexity.

5. RELATIVE BEHAVIOR OF THE MEASURES

We can observed that the construction of UDTs using Measure I is the most expensive among all four measures. This is because Measure I finds the most important attribute by considering the information content (or the ambiguity) of the attribute with respect to the entire attribute set. In general, this kind of feature selection techniques are subtractive methods where the importance of an attribute is judged based on the loss of information content when the attribute is dropped from the set of attributes. On the other hand, construction of UDT using Measure II is less costly than that using Measure I. Thus, in a situation where data records with different class labels (note that, we do not use class labels in the construction of UDT) are highly overlapped, Measure I may provide certain better estimates for selecting the important attributes. The bin size of the histogram should not be so small that very few points are contained in a bin. I observed that so long as the number of bins is approximately less than 1/5th of the number of available data records, the behavior of the measure is stable. Due to this if no. of bins are greater than we make no. of bins equal to 1/5th of the data points and again calculate the histogram as discussed in 6.

6. RESULT AND ANALYSIS

Table 1: Data Sets Used



Data Set	Number Of Samples	Number Of Attributes	No. Of Class	Source
Iris	150	4	3	UCI
Cancer (WDBC1)	569	30	2	UCI
Cancer (WDBC2)	683	10	2	UCI
Liver Disease (bupa)	345	7	2	UCI
Classinfo	196	5	-	Generated

Table shows the data sets that we have used in our work. The size of the unsupervised decision tree is always controlled by the minimum size of the cluster or the no. of data points available at the node under consideration. For every data base we set the minimum size of the node to 10% of the total number of data points in the data set. Again the value of k in k_ary if set to the value five. I.e. if no. of valleys are more than four , then only four out of the valleys detected will be selected by the evaluating the valleys by using the peaks available for the valley by expression (12) discussed in valley detection algorithm. While evaluating the valleys by expression (12) we set the value of λ to 1.

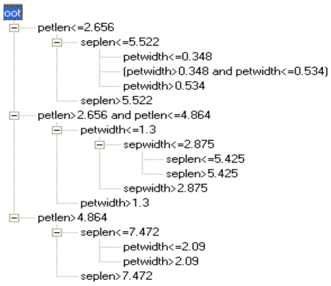


Fig. 2 unsupervised decision tree generated by measure I _ K_ary on IRIS data set

Table 2: Error numbers and 10-fold cross validation scores for Iris Data set.

Sr.No.	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	Total	score	% of classifi cation
UDT1	00	01	03	01	01	01	01	01	02	00	11	07.33	92.67
UDT2	00	02	00	01	04	02	01	01	02	01	14	09.33	90.67
UDT3	00	01	03	01	01	02	02	00	02	00	12	08.00	92.00
UDT4	00	02	00	02	03	02	02	01	02	01	15	10.00	90.00

Table 3. Short forms for Measures-Partitioning Methods

UDT1	MeasureI_K_ary
UDT2	MeasureI_Binary
UDT3	MeasureII_K_ary
UDT4	MeasureII_Binary

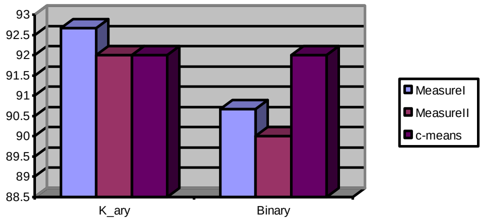


Fig. 3 performance comparison with c-means for IRIS data set for various measures..

REFERENCES

[1] Jayanta Basak and Raghu Krishnapuram, Interpretable Hierarchical Clustering by Constructing an Unsupervised Decision Tree, IEEE Transactions On Knowledge And Data Engineering, Vol. 17, No. 1, January 2005

[2] U.M. Fayyad and K.B. Irani, On the Handling of Continuous values Attributes in Decision Tree Generation, Machine Learning, vol. 8, pp. 87-102, 1992.

[3] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, Classification and Regression Trees. New York: Chapman & Hall, 1993.

[4] C.E. Brodley and P.E. Utgoff, Multivariate Decision Trees, Machine Learning, vol. 19, pp. 45-77, 1995.

[5] Y. Yang and J.O. Pedersen, A Comparatative Study on Feature Selection in Text Categorization, Proc. 14th Int'l Conf. Machine Learning (ICML '97), pp. 412-420, 1997.

[6] P. Bellot and M. El-Beze, Clustering by Means of Unsupervised Decision Trees or Hierarchical and k-Means Like Algorithms, Proc. RIAO 2000 Conf., pp. 344-363, 2000.

[7] B. Liu, Y. Xia, and P. Yu, Clustering through Decision Tree Construction, Technical Report RC 21695, IBM Research Report, IBM, 2000.

[8] M.H. Law, A.K. Jain, and M.A.T. Figueiredo, Feature Selection in Mixture-Based Clustering, Proc. Advances in Neural Information Processing, vol. 15, 2003.

[9] H.H. Bock, Information and Entropy in Cluster Analysis, Proc. First US/Japan Conf. Frontiers of Statistical Modeling, 1994.

[10] F.B. Baulieu, "A Classification of Presence/Absence Based Dissimilarity Coefficients," J. Classification, vol. 6, pp. 233-246, 1989.

[11] J. Basak, R.K. De, and S.K. Pal, Unsupervised Feature Selection Using Neuro-Fuzzy Approach, Pattern Recognition Letters, vol. 19, pp. 997-1006, 1998.

[12] http://www.ics.uci.edu/mlearn/MLRepository.html, 2003.

[13] ftp://ftp.cs.cornell.edu/pub/smart, 2004.

[14] Histogram Smoothing via the Wavelet Transform, Ian Kaplan , September 2002.

[15] file://\\Com1\NEW Folder \ literature \Improved Heterogeneous Distance Function.htm

[16] On Clustering Validation Techniques, Maria Halkidi , Yannis Batistakis , Michalis Vazirgiannis.

[17] Data Clustering: A Review, A.K. JAIN Michigan State University, M.N. MURTY ,Indian Institute of Science AND P.J. FLYNN The Ohio State University