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.

<table>
<thead>
<tr>
<th>no</th>
<th>node</th>
<th>Data</th>
<th>nochild</th>
<th>Parentno</th>
<th>ex</th>
<th>size</th>
<th>fieldsel</th>
<th>criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Normclasinfo</td>
<td>2</td>
<td>0</td>
<td>Yes</td>
<td>345 selector</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>selector1</td>
<td>2</td>
<td>1</td>
<td>Yes</td>
<td>145 gamma1</td>
<td>selector1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>selector2</td>
<td>2</td>
<td>1</td>
<td>Yes</td>
<td>200 sgt</td>
<td>selector2</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 1 database structure to store cluster information.
Finding Attribute Selection Measures by Computing Loss of Information and Ambiguity for Data Clustering

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. Criteria. 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 \( \hat{\mu}_{ij}^a \) 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 it as,

\[
\hat{\mu}_{ij}^a = g \left( \frac{1-d_{ij}}{d_{\max}} \right)
\]

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,

\[
g(x) = \begin{cases} 
  x & \text{for } 0 \leq x \leq 1 \\
  0 & \text{otherwise}
\end{cases}
\]

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,

\[
H_a = - \left( \hat{\mu}_{ij}^a \left( 1-\hat{\mu}_{ij}^a \right) + \hat{\mu}_{ij}^a \left( 1-\hat{\mu}_{ij} \right) \right)
\]

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

\[
\hat{\mu}_{ij}^a = \hat{\mu}_{ij} \left( 1-\hat{\mu}_{ij}^a \right) 
\]

\( \hat{\mu}_{ij}^a \) 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,

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

Where \( | z | \) represents the absolute value of z, \( x \) is the first point...a 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 (continuous) attribute.
As per this method distance between two values x and y of a given attribute a as ,
\[ f = \arg \max_{y \in \mathcal{F}} \left\{ H_a(D_y^x) + H_a(D_y^y) - H_a(D) \right\} \]
\[ D_a(x,y) = \begin{cases} 
1 & \text{if } x \text{ or } y \text{ is unknown , else} \\
\text{overlap}(x,y) & \text{if } a \text{ is nominal} \\
\text{diff}(x,y) & \text{else}
\end{cases} \]  
(6)

Unknown attribute values are handled by returning an attribute distance 1 ( i.e. a maximum distance ) if either of the attribute value is unknown. The function overlap and range-normalized difference \( \text{m}_\text{diff} \) are defined as,
\[ \text{Overlap}(x,y) = \begin{cases} 
0 & \text{if } x = y \\
1 & \text{otherwise}
\end{cases} \]  
(7)
\[ \text{m}_\text{diff}(x,y) = \frac{|x-y|}{\text{range}_a} \]
The value range\(_a\) is used to normalize the attribute , and is defined as ,
\[ \text{range}_a = \text{max}_a - \text{min}_a \]
Where max\(_a\) and min\(_a\) are the maximum and minimum values respectively. The overall distance between input vectors x and y is given by,
\[ \text{HOEM} = \sum \sqrt{d_a(x_a,y_a)}^2 \]  
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
\[ e_i = \min \left\{ q_i - v_i, q_{i+1} - v_i \right\} \]
\[ 1 + \lambda v_i \]
where \( v_i \) is the height of the valley, and \( q_i \) and \( q_{i+1} \) are heights of the peaks on either side of the valley. Based on the evaluated score \( e_i \), 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 \( c_1, c_2, \ldots, c_{k-1} \) be the cut-off points. For every pair \( (c_i, c_{i+1}) \) of consecutive cut-off points, the data records for which the value of the attribute falls between \( c_i \) and \( c_{i+1} \) constitute the \( i \)-th 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 ,
\[ \text{PeakLim} = \frac{\text{No. of data records available}}{\text{No. of bins in histogram}} \]  
(8)
4.3. Binary Partitioning of Data

Let $H(D)$ represent the inhomogeneity or information content of the data set $D$ with respect to some attribute $a$. 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,

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

if $\max_c \nabla H(c,D) > 0$, then we can find cut-off point $c_0$ as,

$$c_0 = \arg \max_c \{ \nabla H(c,D) \}$$

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 $1/5$th 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 $1/5$th of the data points and again calculate the histogram as discussed in 6.

6. RESULT AND ANALYSIS

**Table 1: Data Sets Used**

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Number of Samples</th>
<th>Number of Attributes</th>
<th>No. of Class Labels</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>150</td>
<td>4</td>
<td>2</td>
<td>ZERO</td>
</tr>
<tr>
<td>Ticin</td>
<td>349</td>
<td>86</td>
<td>2</td>
<td>ZERO</td>
</tr>
<tr>
<td>Ulysses</td>
<td>90</td>
<td>96</td>
<td>2</td>
<td>ZERO</td>
</tr>
<tr>
<td>Kansas</td>
<td>343</td>
<td>7</td>
<td>2</td>
<td>ZERO</td>
</tr>
<tr>
<td>Iris</td>
<td>150</td>
<td>3</td>
<td>-</td>
<td>Standard</td>
</tr>
</tbody>
</table>

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 $\lambda$ to 1.

![Fig. 2 unsupervised decision tree generated by measure I _ K_ary on IRIS data set](image-url)
Finding Attribute Selection Measures by Computing Loss of Information and Ambiguity for Data Clustering

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

<table>
<thead>
<tr>
<th>Sr.No</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
<th>S7</th>
<th>S8</th>
<th>S9</th>
<th>Total</th>
<th>score</th>
<th>% of classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDT1</td>
<td>00</td>
<td>01</td>
<td>03</td>
<td>01</td>
<td>01</td>
<td>02</td>
<td>00</td>
<td>00</td>
<td>11</td>
<td>10.33</td>
<td>84.63</td>
<td></td>
</tr>
<tr>
<td>UDT2</td>
<td>00</td>
<td>02</td>
<td>00</td>
<td>03</td>
<td>04</td>
<td>02</td>
<td>00</td>
<td>00</td>
<td>14</td>
<td>13.33</td>
<td>92.86</td>
<td></td>
</tr>
<tr>
<td>UDT3</td>
<td>00</td>
<td>01</td>
<td>01</td>
<td>02</td>
<td>02</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>12</td>
<td>11.00</td>
<td>82.00</td>
<td></td>
</tr>
<tr>
<td>UDT4</td>
<td>00</td>
<td>02</td>
<td>00</td>
<td>02</td>
<td>03</td>
<td>02</td>
<td>01</td>
<td>02</td>
<td>13</td>
<td>12.00</td>
<td>82.00</td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Short forms for Measures-Partitioning Methods

<table>
<thead>
<tr>
<th>UDT1</th>
<th>Measure I_K_ary</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDT2</td>
<td>Measure I_Binary</td>
</tr>
<tr>
<td>UDT3</td>
<td>Measure II_K_ary</td>
</tr>
<tr>
<td>UDT4</td>
<td>Measure II_Binary</td>
</tr>
</tbody>
</table>

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

REFERENCES

[15] file://C:\NEW Folder\literature\Improved Heterogeneous Distance Function.htm
[16] On Clustering Validation Techniques, Maria Halkidi, Yannis Batistakis, Michalis Vazirgiannis.