

## **Clustering data without distance functions**

Traditional clustering algorithms use distance functions to measure similarity between elements.

Problem with using distance functions:

Technique is difficult to use with applications that do not have natural distance measures; e.g. data sets with elements that have many categorical attributes.

This paper suggests a solution to this problem. The new method is based on two important principles.

- Elements that share many attribute values are similar.
- Want to group elements that share a common value for a specified attribute or label.

Method uses a combination of the two methods: the exact combination depends on a user-defined parameter.

Two important concepts:

- **Co-occurrence maximization**: clustering technique that tries to group together records that have frequently co-occurring items.
- **Label minimization**: clustering technique that tries to group together records with the same label.

These two methods can co-exist in hierarchical partitioning of the data. At a node of partition, the choice of a particular method depends on:

1. Quality measures of the two methods.
2. User defined numeric parameter that allows continuous variation between the two methods.

### **Co-occurrence maximization**:

- Let the input consist of set of data records  $R$ .
- Each record  $r \in R$  defines a set of (attribute, value) pairs.
- Define the set of items  $I$  to be the set of all possible (attribute, value) pairs that occur in  $I$ .
- Threshold frequency of an item/combination of items is given by the support  $s$ .

- Pairs of items present in greater than a fraction  $s$  of the records in  $R$  are called frequent item sets  $F \subseteq I^2$ .

### **Calculation of frequent item sets:**

- Get frequent item sets of cardinality 2 from  $R$ , using an association mining algorithm such as Apriori algorithm.
- Construct a graph  $G$  whose vertices are elements of  $I$ , and frequent item sets  $J$  as the edges. The weight on the edge is the support for the item set.
- Partition  $I$  into subsets  $I_A$  and  $I_B$  so that:
  1. weight of the edges across the partition is minimized. Such a partition is denoted as  $P_{AB}$ .
  2. number of vertices in the larger partition is close to  $n/2$ . This is done using the MinCut algorithm recursively, stopping when (2) is satisfied.
- Partition  $R$  into subsets  $R_A$  and  $R_B$  so that a record  $r$  of  $R$  goes into  $R_A$  if its overlap with  $I_A$  is greater than  $I_B$ . The effect of a single item frequency is accounted for by weighting each item's contribution with the inverse of its

support. The partitioning induced in  $R$  due the partitioning in  $I$  is denoted as  $q_{AB}$ .

### **Quality of partitioning:**

Quality of a partitioning is defined as a numeric between 0 & 1.

For any  $r \subseteq R$ , if  $r$  falls entirely in  $I_A$ , then its partition quality is 1; if it overlaps  $I_A$  and  $I_B$  equally, then its partition quality is 0.

Weighted cardinality of an item set is  $\sum_{i \in I} (1/s_i)$ , where  $s_i$  is the support of the item in  $R$ .

Quality of a record  $r$  over  $p_{AB}$  is:

$$Q_c(r) = \frac{\text{abs}(s(r \cap I_A) - s(r \cap I_B))}{|I_A| + |I_B|}$$

$s(r \cap I_A)$  represents a set of items, and the support is over the data set  $R$ .

Quality of the data set  $R$  over  $p_{AB}$  is:

$$Q_c(R) = \frac{\sum Q_c(r)(r, p_{AB})}{|R|}$$

### **Label minimization:**

Partition  $R$  based on the class labels.

$C$  is the set of class labels of  $R$ .

**Technique:**

Minimize the number of labels in each partition; ideally have one class label for a partition. Partitioning is achieved using candidate tests from (attribute, value) pairs.

For categorical attributes, candidate tests are equality tests for each value of the attribute, for numerical attributes are range tests, i.e. whether the value for an attribute lies in a specified range.

For each test  $t$ , the data records are divided into two partitions  $R_A$  and  $R_B$ , and the gain in information measured. After partition, there is less uncertainty about the type of records in a particular cluster; i.e. the information content of the cluster increases.

For each class label  $c \in C$ , let  $n(c, R)$  denote the frequency of occurrence of  $c$  in  $R$ . The information content  $i(R)$  is defined as:

$$i(R) = \frac{\sum_{c \in C} n(c, R) \cdot (-\log_2(n(c, R)/|R|))}{|R|}$$

The combined information content of  $R_A$  and  $R_B$  is:

$$i(R_A, R_B) = \frac{i(R_A|R_A) + i(R_B|R_B)}{|R_A| + |R_B|}$$

The information gained by partitioning  $R$  into  $R_A$  and  $R_B$  using a test  $t$  is  $i(R_A, R_B) - i(R)$ .

The quality of the label minimization method is the relative information gain:

$$Q_l = i(R_A, R_B) - i(R)$$

The partitioning process is done recursively. To determine which of the two methods to choose at each node of partition, use user-defined parameter  $w$ . At each node:

if  $w * Q_c(R) > (1 - w) * Q_l$  use co-occurrence maximization;

else use label minimization

if tie, use the partitioning technique that has the best quality among the two.

### **Experiments:**

Sample data sets from UC Irvine ML repository.

- Zoo data set of animals.

## **Results from zoo data set:**

The class attribute is the attribute: type; distinguishes between mammals, fish, insects, birds and reptiles etc.

The schema attributes are:

- Animal name : string.
- Hair, feathers, eggs, milk, airborne, aquatic, predator, toothed, backbone, breathes, venomous and fins: boolean
- Legs: numeric(set of values: 0,2,4,5,6,8)
- Tail, domestic, catsize: all boolean.
- Type: numeric(integer values in range[1,7]).

User defined parameter  $w = 0.35$  and frequent item set threshold 0.06.

The test at the root node was label minimization with milk attribute; all mammals to the left. The set of mammals was then divided into clusters A and B using co-occurrence maximization, using legs as the attribute.

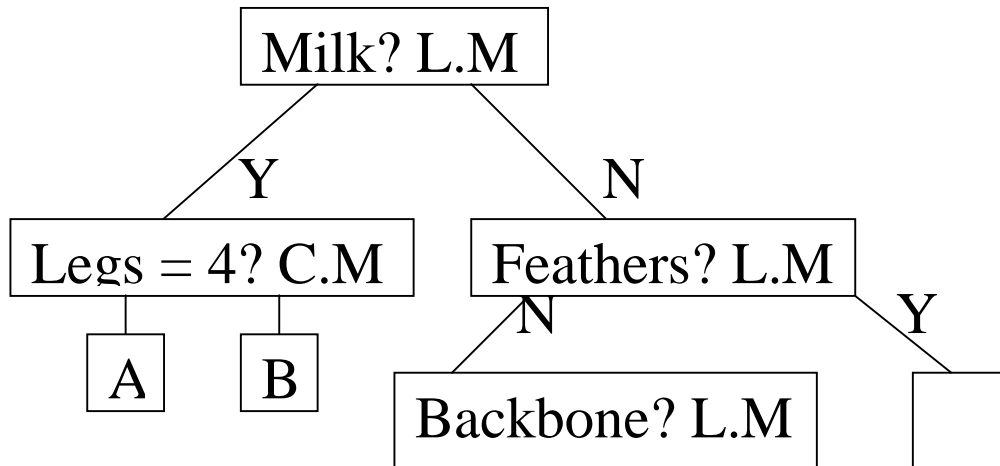
A: Bear, boar, buffalo, cheetah, deer, elephant, leopard.

B: Dolphin, fruitbat, gorilla, girl, platypus, seal, sealion.

Characteristics obtained by the partition:

A: none of these animals fly.

B: none of these are fast-moving animals on land.



### **Conclusion:**

Uses techniques from association rules and classification.

The use of user defined parameter in deciding between the two methods can be used to favor one method over the other.

The quality function used depends linearly on user defined parameter and the quality functions of the two methods.