

Knowledge Discovery and Data Mining

Unit # 13

Distance Computation

- Interval-Scaled Variables
- Binary Variables
- Categorical Variables
- Ordinal Variables

Interval-Scaled Variables

- *Interval-scaled variables are continuous measurements* of a roughly linear scale.
- Typical examples include weight and height, latitude and longitude coordinates (e.g., when clustering houses), and weather temperature.
- Both Euclidean distance and Manhattan distances are generally used for distance computation.

Binary Variables

- One approach involves computing a dissimilarity matrix from the given binary data.
- If all binary variables are thought of as having the same weight, we have the 2-by-2 contingency table, where
 - q is the number of variables that equal 1 for both objects i and j ,
 - r is the number of variables that equal 1 for object i but that are 0 for object j ,
 - s is the number of variables that equal 0 for object i but equal 1 for object j , and
 - t is the number of variables that equal 0 for both objects i and j .
- The total number of variables is p , where $p = q+r+s+t$.

Symmetric Binary Variables

- A *binary* variable is symmetric if both of its states are equally valuable and carry the same weight; that is, there is no preference on which outcome should be coded as 0 or 1.
- One such example could be the attribute *gender having the states male and female*.

$$d(i, j) = \frac{r+s}{q+r+s+t}$$

Asymmetric Binary Variables

- A binary variable is asymmetric if the outcomes of the states are not equally important, such as the *positive and negative outcomes of a disease test*.
- *By convention*, we shall code the most important outcome, which is usually the rarest one, by 1
- (e.g., *HIV positive*) and the other by 0 (e.g., *HIV negative*).

$$d(i, j) = \frac{r+s}{q+r+s}, \quad \text{sim}(i, j) = \frac{q}{q+r+s} = 1 - d(i, j).$$

- The coefficient $\text{sim}(i, j)$ is call the Jaccard coefficient, which is popularly referenced in the literature.

Categorical Variable

- A categorical variable is a generalization of the binary variable in that it can take on more than two states.
- For example, map color is a categorical variable that may have, say, five states: red, yellow, green, pink, and blue.
- The dissimilarity between two objects i and j can be computed based on the ratio of mismatches:

$$d(i, j) = \frac{p - m}{p},$$

- where m is the number of matches (i.e., the number of variables for which i and j are in the same state), and p is the total number of variables.

Ordinal Variables

- The treatment of ordinal variables is quite
- similar to that of interval-scaled variables when computing the dissimilarity between objects.
- The dissimilarity computation with respect to f involves the following steps:
 1. The value of f for the i th object is x_{if} , and f has M_f ordered states, representing the ranking $1, \dots, M_f$. Replace each x_{if} by its corresponding rank, $r_{if} \in \{1, \dots, M_f\}$.
 2. Since each ordinal variable can have a different number of states, it is often necessary to map the range of each variable onto $[0,0,1.0]$ so that each variable has equal weight. This can be achieved by replacing the rank r_{if} of the i th object in the f th variable by
 3. Dissimilarity can then be computed using any of the distance measures described for interval-scaled variables

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}.$$

Mixed Type Variables

Suppose that the data set contains p variables of mixed type. The dissimilarity $d(i, j)$ between objects i and j is defined as

$$d(i, j) = \frac{\sum_{f=1}^p \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^p \delta_{ij}^{(f)}}, \quad (7.15)$$

where the indicator $\delta_{ij}^{(f)} = 0$ if either (1) x_{if} or x_{jf} is missing (i.e., there is no measurement of variable f for object i or object j), or (2) $x_{if} = x_{jf} = 0$ and variable f is asymmetric binary; otherwise, $\delta_{ij}^{(f)} = 1$. The contribution of variable f to the dissimilarity between i and j , that is, $d_{ij}^{(f)}$, is computed dependent on its type:

- If f is interval-based: $d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{\max_h x_{hf} - \min_h x_{hf}}$, where h runs over all nonmissing objects for variable f .
- If f is binary or categorical: $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$; otherwise $d_{ij}^{(f)} = 1$.
- If f is ordinal: compute the ranks r_{if} and $r_{jf} = \frac{r_{if} - 1}{M_f - 1}$, and treat r_{if} as interval-scaled.

Vector Objects

- In some applications, such as information retrieval, text document clustering, and biological taxonomy, we need to compare and cluster complex objects (such as documents) containing a large number of symbolic entities (such as keywords and phrases).

There are several ways to define such a similarity function, $s(\mathbf{x}, \mathbf{y})$, to compare two vectors \mathbf{x} and \mathbf{y} . One popular way is to define the similarity function as a cosine measure as follows:

$$s(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^t \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}, \quad (7.16)$$

where \mathbf{x}^t is a transposition of vector \mathbf{x} , $\|\mathbf{x}\|$ is the Euclidean norm of vector \mathbf{x} ,¹ $\|\mathbf{y}\|$ is the Euclidean norm of vector \mathbf{y} , and s is essentially the cosine of the angle between vectors \mathbf{x} and \mathbf{y} .

¹The Euclidean normal of vector $\mathbf{x} = (x_1, x_2, \dots, x_p)$ is defined as $\sqrt{x_1^2 + x_2^2 + \dots + x_p^2}$. Conceptually, it is the length of the vector.

Recap of K-Means

- The K-Means node provides a method of cluster analysis.
- It can be used to cluster the data set into distinct groups when you don't know what those groups are at the beginning.
- Instead of trying to predict an outcome, K-Means tries to uncover patterns in the set of input fields.
- Records are grouped so that records within a group or cluster tend to be similar to each other, but records in different groups are dissimilar.
- Note: The resulting model depends to a certain extent on the order of the training data. Reordering the data and rebuilding the model may lead to a different final cluster model.

Recap of K-Means (Cont'd)

- K-Means works by defining a set of starting cluster centers derived from data.
- It then assigns each record to the cluster to which it is most similar, based on the record's input field values.
- After all cases have been assigned, the cluster centers are updated to reflect the new set of records assigned to each cluster.
- The records are then checked again to see whether they should be reassigned to a different cluster, and the record assignment/cluster iteration process continues until either the maximum number of iterations is reached, or the change between one iteration and the next fails to exceed a specified threshold.

Fuzzy c-Means

- The fuzzy c -means algorithm is very similar to the k -means algorithm:
 - Choose a number of clusters.
 - Assign randomly to each point coefficients for being in the clusters.
 - Repeat until the algorithm has converged (that is, the coefficients' change between two iterations is no more than , the given sensitivity threshold) :
 - Compute the centroid for each cluster, using the formula on the next slide.
 - For each point, compute its coefficients of being in the clusters, using the formula on the next slide.
 - The algorithm minimizes intra-cluster variance as well, but has the same problems as k -means, the minimum is a local minimum, and the results depend on the initial choice of weights.

Fuzzy c-Means (Cont'd)

$$\forall x \left(\sum_{k=1}^{\text{num. clusters}} u_k(x) = 1 \right).$$

With fuzzy c -means, the centroid of a cluster is the mean of all points, weighted by their degree of belonging to the cluster:

$$\text{center}_k = \frac{\sum_x u_k(x)^m x}{\sum_x u_k(x)^m}.$$

The degree of belonging is related to the inverse of the distance to the cluster center:

$$u_k(x) = \frac{1}{d(\text{center}_k, x)},$$

then the coefficients are normalized and fuzzyfied with a real parameter $m > 1$ so that their sum is 1. So

$$u_k(x) = \frac{1}{\sum_j \left(\frac{d(\text{center}_k, x)}{d(\text{center}_j, x)} \right)^{2/(m-1)}}.$$

KNIME Demo (Clustering)