Data Mining, Lecture 9

Similarity and Distance II

S. Nõmm

[1]Department of Software Science, Tallinn University of Technology

16.11.2021

Match-based Similarity Computation

- How to select (in practice) relevant features for the similarity computations?
- For high dimensional data: impact of the noisy variation along individual attributes needs to be de-emphasized while counting the cumulative match across many dimensions.
- One of the methods is :*proximity thresholding* in a dimensionality-sensitive way.
- Perform discretization into k_d *equidepth buckets*. Each dimension is divided into k_d equidepth buckets, containing $1/k_d$ records. $k_d \propto d$ (proportional).

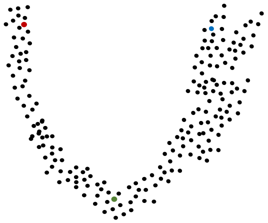
Definition

Two records of dimension d $\bar{X}$ and $\bar{Y}$ are said to be in proximity on dimension i if x_i and y_i belong to the same bucket.

Match-based Similarity Computation

- The subset of dimensions where $\bar{X}$ and $\bar{Y}$ map to the same bucket is referred as the proximity set, and denoted $\mathcal{S}(\bar{X}, \bar{Y}, k_d)$.
- Let m_i and n_i be the upper and lower bounds of the bucket in dimension i
- Similarity is defined as follows.
- $$S_p(\bar{X}, \bar{Y}, k_d) = \left[\sum_{i \in \mathcal{S}(\bar{X}, \bar{Y}, k_d)} \left(1 - \frac{| x_i - y_i |}{m_i - n_i} \right)^p \right]^{\frac{1}{p}}$$

Nonlinear distributions ISOMAP



- Let the data contain nonlinear distributions of arbitrary shape.
- Consider points A -blue, B - red and C - green. On the basis of the Euclidian distance A and B are the closest.
- According to the *geodesic distance*, where only small "jumps" are allowed A and B are not the closest.

Nonlinear distributions ISOMAP

- Compute the k-nearest neighbors of each point. Construct a weighted graph G with nodes representing data points, and edge weights (costs) representing distances of these k-nearest neighbors.
- For any pair of points X and Y, find $S(X, Y)$ as the shortest path between the corresponding nodes in G.

Impact of local data distributions

- Distribution of the data varies significantly with locality.
    - Density
    - Orientation
- Shared Nearest-Neighbor Similarity addresses the first problem.
- Generic Methods:
    - Partition the data into a set of local regions.
    - For any pair of objects, determine the most relevant region for the pair, and compute the pairwise distances using the local statistics of that region. For example, the local Mahalanobis distance may be used in each local region.

Dynamic Time Wrapping DTW

- DTW stretches the series along the time axis in a varying (or dynamic) way over different portions to enable more effective matching.
- Unlike L_p norms it allows many-to-one mappings. In turn allows to artificial creation of two equal length series.
- Temporal data is usually represented by continuous or discrete time-series.

DTW Example

- Consider L_1 Manhattan metric computed on the first n elements of two time series $\bar{X} = (x_1, \dots x_n)$ and $\bar{Y} = (y_1, \dots y_n)$.
- The value of Manhattan metric may be written recursively as follwos

$$M_(\bar{X}_i, \bar{Y}_i) = \mid x_i - y_i \mid + M(\bar{X}_{i-1}, \bar{Y_{i-1}})$$

- Let DTW will be optimal distance between the first i and the first j elements of two time series $\bar{X} = (x_1, \dots x_n)$ and $\bar{Y} = (y_1, \dots y_m)$. Then DTW is defines as follows

$$DTW(i,j) = s(x_i, y_i) + \min \begin{cases} DTW(i, j-1) & \text{repeat} \quad x_i \\ DTW(i-1, j) & \text{repeat} \quad y_i \\ DTW(i-1, j-1) & \text{repeat} \quad \text{neither} \end{cases}$$

Window-based methods

- Dropped readings (observation points) may cause a gap in the matching.
- If two sequences have contiguous matching segments, they should be considered similar.
- Let $\bar{X}$ and $\bar{Y}$ are two series, and let $\bar{X}_1, \ldots, \bar{X}_r$ and $\bar{Y}_1, \ldots, \bar{Y}_r$ be temporally ordered and nonoverlapping windows extracted from the respective series. Then overall similarity function may be defines as follows

$$S(\bar{X}\bar{Y}) = \sum_{i=1}^{r} M(\bar{X}_r\bar{Y}_r)$$

where $M(\bar{X}_r\bar{Y}_r)$ is some suitable matching function.

Discrete Sequence Similarity Measures: Edit distance

- Let us consider Levenshtein distance in a more detailed way.
- *Edit* distance defines the similarity between the two strings in terms of cost (effort) of transforming one sequence into another.
- Let $\bar{X} = (x_1, \ldots, x_m)$, $\bar{Y} = (y_1, \ldots, y_n)$ be the two sequences and let the edit is performed to transform X into Y.
- the optimal matching is defined as follwos

$$E(i,j) = \min \begin{cases} E(i-1,j) + C_D \\ E(i-1,j) + C_I \\ E(i-1,j-1) + I_{i,j} \cdot C_R \end{cases}$$

where C_D denotes deletion cost, C_I -insertion cost and C_R -replacement cost.

Discrete Sequence Similarity Measures: Longest Common Subsequence

- Optimal matching based on the longest common subsequence LCSS is defined as follows.

$$LCSS(i,j) = \max \begin{cases} LCSS(i-1,j-1) & \text{if} \quad x_i = y_i \\ LCSS(i-1,j) & \text{otherwise} \quad \text{no} \quad \text{match} \quad \text{on} \quad x_i \\ LCSS(i,j-1) & \text{otherwise} \quad \text{no} \quad \text{match} \quad \text{on} \quad y_i \end{cases}$$

Temporal Similarity Measures

- Temporal data contain a single contextual attribute representing time and one or more behavioral attributes that measure the properties varying along a particular time period.
- Temporal data shares some similarities with discrete sequences *strings*.
- Temporal data is usually represented by continuous or discrete time-series.
- Time-series similarity measures
  - Behavioral attribute scaling and translation.
  - Temporal (contextual) attribute translation.
  - Temporal (contextual) attribute scaling.
  - Noncontiguity in matching.
- L_p norms may be used to measure similarities of time series of equal length. But can not address distortions on temporal (contextual) attributes.

Text Similarity Measures

- Levenshtein or SED distance. SED - minimal number of single -charter edits required to change one string into another. Edit operations are as follows:
  - insertions
  - deletions
  - substitutions
- SED(delta, delata)$=1$ delete "a" or SED(kitten,sitting)$=3$: substitute "k" with "s",substitute "e" with "i", insert "g".
- Hamming distance Similar to Levenshtein but with substitution operation only. Frequently used with categorical and binary data.

Text Similarity Measures

- Text may be considered as the multidimensional data if it treated as the bag of words. Lexicon can be treated as the full set of attributes and word frequencies as the quantitative attributes.
- Many attributes would take the value "0", L_p norms do not adjust well to the different lengthes of the documents.
- The cosine measure computes the *angle* between two documents. Let $\bar{X} = (x_1, \ldots, x_d)$ and $\bar{Y} = (y_1, \ldots, y_d)$ will be two documents on a lexicon of size d. Then cosine measure is defines as follows:

$$\cos(\bar{X}, \bar{Y}) = \frac{\displaystyle\sum_{i=1}^{d} x_i \cdot y_i}{\displaystyle\sum_{i=1}^{d} x_i^2 \cdot \sum_{i=1}^{d} y_i^2}$$

This measure uses the raw frequencies between attributes.

Text Similarity Measures

- If two documents match in a uncommon word it is more indicative than if two documents match on a word that occurs very commonly.
- The *inverse document frequency* is defined as

$$id_i = \log(n/n_i).$$

- Sometimes a dumping function $f(\cdot)$ which is either logarithm or square root is used.
- The normalized frequency for the ith word is defined as follows

$$h(x_i) = f(x_i) \cdot id_i$$

- Combined together with the cosine distance this leads

$$\cos(\bar{X}, \bar{Y}) = \frac{\sum_{i=1}^{d} h(x_i) \cdot h(y_i)}{\sum_{i=1}^{d} h(x_i)^2 \cdot \sum_{i=1}^{d} h(y_i)^2}$$

Binary and Sparse Data

- Already known to you Jaccard distance may be rewritten in the following way

$$J(\bar{X},\bar{Y}) = \frac{\sum_{i=1}^{d} h(x_i)\cdot h(y_i)}{\sum_{i=1}^{d} h(x_i)^2 + \sum_{i=1}^{d} h(y_i)^2 - \sum_{i=1}^{d} h(x_i)\cdot h(y_i)}$$

- Let $\bar{X}$, $\bar{Y}$ are binary representations of the two sets H_X and H_Y respectively. Then

$$J(\bar{X},\bar{Y}) = \frac{\sum_{i=1}^{d} x_i\cdot y_i}{\sum_{i=1}^{d} x_i^2 + \sum_{i=1}^{d} y_i^2 - \sum_{i=1}^{d} x_i\cdot y_i}$$

Categorical data

- $$S(\bar{X}\bar{Y}) = \sum_{i=1}^{d} s(x_i, y_i).$$
- The choice of similarity function $s(x_i, y_i)$ defines the overall similarity function S
  - The simplest possible function is $s(x_i, y_i) = \mathbb{I}(x_i, y_i)$
  - *Inverse occurrence frequency*

$$s(x_i, y_i) = \begin{cases} \dfrac{1}{p_k(x_i)^2} & \text{if} \quad x_i = y_i \\ 0 & \text{otherwise} \end{cases}$$

    here $p_k(x)$ is the fraction of records where $x_i = y_i$
  - *Goodall measure*

$$s(x_i, y_i) = \begin{cases} 1 - p_k(x_i)^2 & \text{if} \quad x_i = y_i \\ 0 & \text{otherwise} \end{cases}$$

Mixed Quantitative and Categorical Data

$$d(x,y) = \lambda d_q(x_q, y_q) + (1-\lambda)(d_c(x_c, y_c))$$

here index q denotes quantitative and c categorical data.

- how to choose λ?
- data normalization ?

Gower's distance

$$D_G(x, y) = \frac{\sum_{k}^{n} w_{ijk} S_{ijk}}{\sum_{k}^{n} w_{ijk}}$$

where s_{ijk} is the contribution of the feature k, w_{ijk} takes the values 0 and 1 depending on the comparison results for the feature k

Gower's distance

Gower's distance function is designed for the datasets which features are of a mixed type. It is based on the range normalized Manhattan distance for continues features and dice distance for binary features.

$$D_G(x, y) = 1 - \left(\frac{1}{p} \sum_{j=1}^{p} s_j(x, y) \right)$$

where:

- **Case of numeric features** index s_j is the partial distance (similarity) function computed separately for each feature (dimension)

  $$s_j = (x, y) = 1 - \frac{|x_j - y_j|}{R_j}$$

  here R_j is range of the feature j
- **Case of categoric features** dice distance is used.

  $$S_{DSC} = \frac{2|x \cap y|}{|X| + |Y|}$$