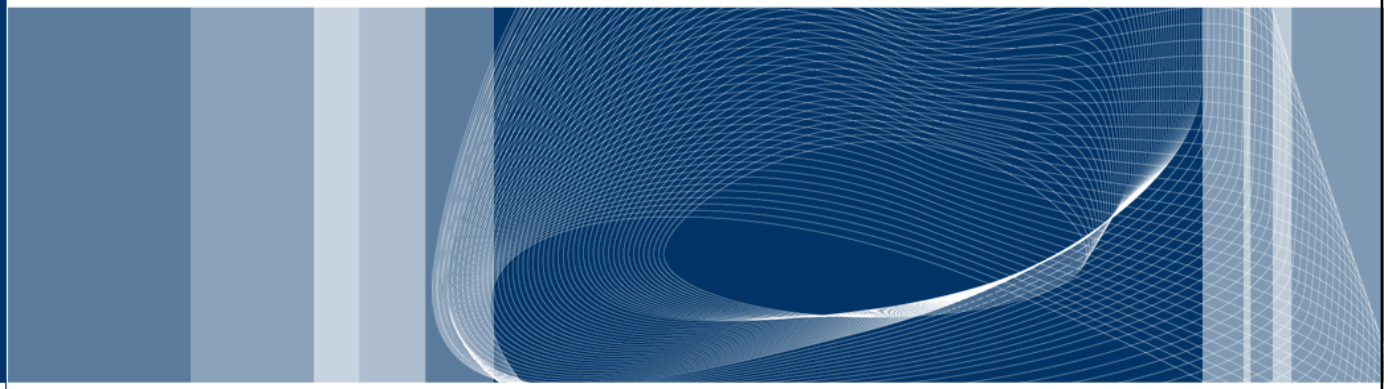
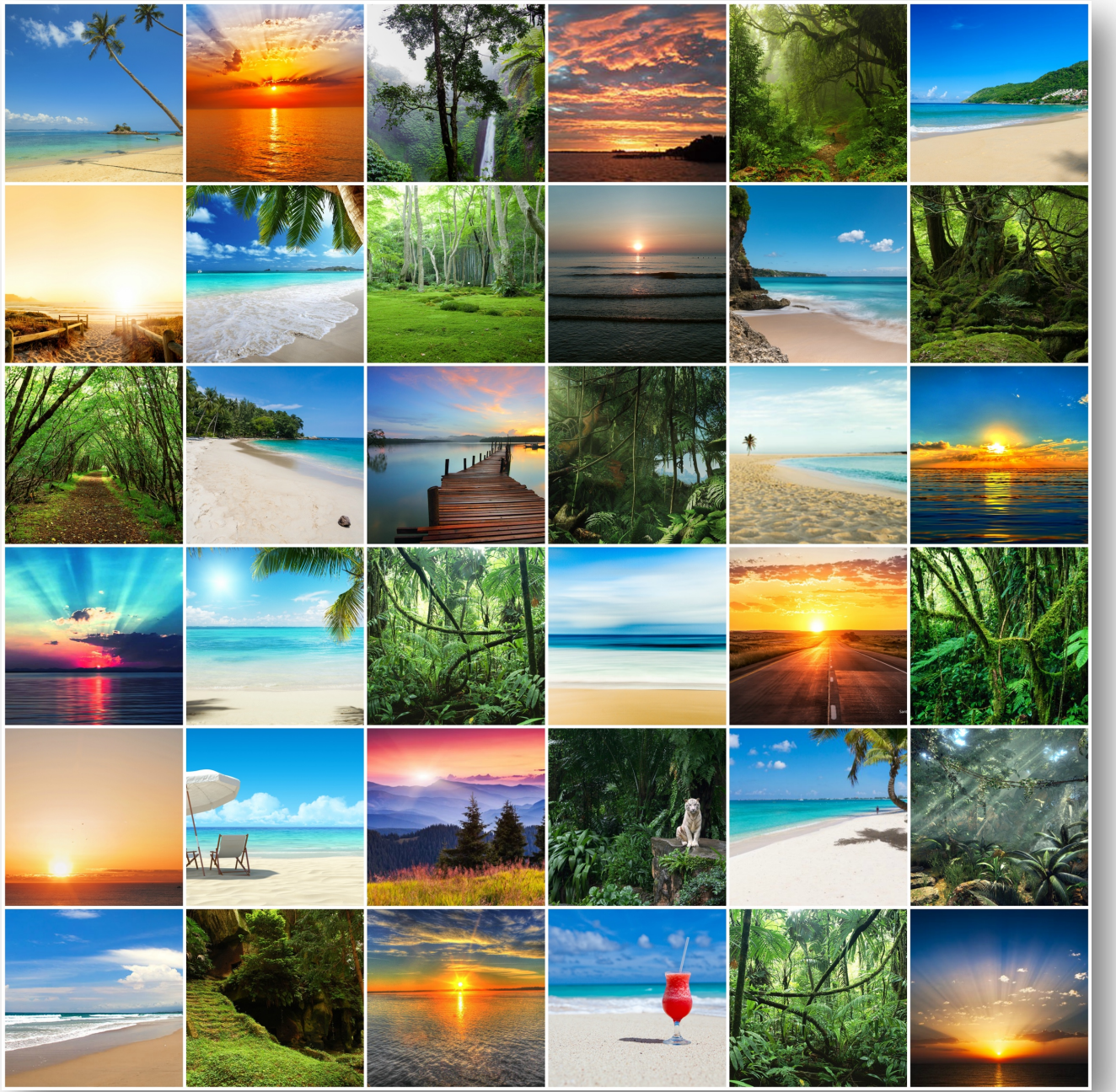
POLITECNICO DI MILANO



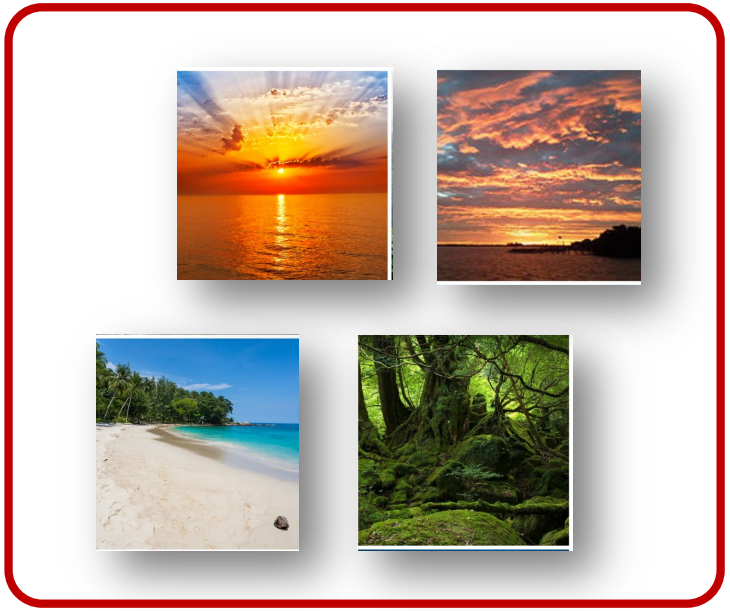
Clustering

Data Mining and Text Mining

Prof. Pierluca Lanzi



Clustering searches for “natural” grouping/structure in un-labeled data



Clustering algorithms group a collection of data points into "clusters" according to some distance measure

Data points in the same cluster should have a small distance from one another

Data points in different clusters should be at a large distance from one another

Clustering Applications

- Marketing
  - Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Land use
  - Identification of areas of similar land use in an earth observation database
- Insurance
  - Identifying groups of motor insurance policy holders with a high average claim cost
- City-planning
  - Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies
  - Observed earthquake epicenters should be clustered along continent faults

What is Cluster Analysis?

- A cluster is a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters

- Cluster analysis
  - Given a set data points try to understand their structure
  - Finds similarities between data according to the characteristics found in the data
  - Groups similar data objects into clusters
  - It is unsupervised learning since there is no predefined classes

What Is Good Clustering?

- A good clustering consists of high-quality clusters with
  - High intra-class similarity
  - Low inter-class similarity
- The quality of a clustering result depends on both
  - The similarity measure used by the method and
  - Its implementation (the algorithms used to find the clusters)
- The quality of a clustering method is also measured by its ability to discover some or all the hidden patterns
- Evaluation
  - Various measure of intra/inter cluster similarity
  - Manual inspection
  - Benchmarking on existing labels

Measure the Quality of Clustering

- Dissimilarity/Similarity metric
  - Similarity expressed in terms of distance function, typically a metric, d(i, j)
  - Definitions of distance functions are usually very different for interval-scaled, Boolean, categorical, ordinal ratio, and vector variables
  - Weights can/should be associated with different variables based on applications and data semantics
- Cluster quality measure
  - Separate from distance, there is a "quality" function that measures the "goodness" of a cluster
  - It is hard to define "similar enough" or "good enough" as the answer is typically highly subjective

Clustering Strategies

- Hierarchical vs point assignment
- Numeric and/or symbolic data
- Deterministic vs. probabilistic
- Exclusive vs. overlapping
- Hierarchical vs. flat
- Top-down vs. bottom-up

Data Structures

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
…	…	…	…	…

Data Matrix

$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix}$$

Dis/Similarity Matrix

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

Distance and Similarity Measures

Distance Measures

- Given a space and a set of points on this space, a distance measure d(x,y) maps two points x and y to a real number, and satisfies three axioms
- $d(x,y) \geq 0$
- $d(x,y) = 0$ if and only $x=y$
- $d(x,y) = d(y,x)$
- $d(x,y) \leq d(x,z) + d(z,y)$

What Similarity Measure?

- Euclidian distance is the typical function used to compute the similarity between two examples

$$d([x_1, x_2, \ldots, x_n],\ [y_1, y_2, \ldots, y_n]) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

- Another popular metric is city-block (Manhattan) metric, distance is the sum of absolute differences

$$d([x_1, x_2, \ldots, x_n], [y_1, y_2, \ldots, y_n]) = \sum_{i=1}^{n} |x_i - y_i|$$

Jaccard Distance

- Jaccard distance is a measure of how dissimilar two sets are. Examples are represented by binary variables and are viewed as representations of set
- Jaccard distance is defined as

$$d(x, y) = 1 - J(x, y)$$

- J is the Jaccard similarity which is computed as the number of

$$J(x, y) = \frac{|x \cap y|}{|x \cup y|}$$

- Which can also be interpreted as the percentage of identical attributes

Examples of Jaccard Distance

- Consider the following example consisting of two examples

Outlook_overcast	Outlook_rainy	Outlook_sunny	Humidity	Windy
0	0	1	0	0
0	0	1	0	1

- They are at Jaccard distance of 0.5 since the first example "contains" only Outlook_sunny whereas the second example "contains" both Outlook_sunny and Windy so intersection has size one while union has size two. The Jaccard distance is 0.5 so the similarity is 1-0.5 that is 0.5

Outlook_overcast	Outlook_rainy	Outlook_sunny	Humidity	Windy	Temperature
0	0	1	0	0	1.0
0	0	1	0	1	1.0

Outlook_overcast	Outlook_rainy	Outlook_sunny	Humidity	Windy	Temperature
0	0	1	0	0	2.0
0	0	1	0	1	1.0

Jaccard distance is 0.333 in the first case and 0.667 in the second case

Hamming Distance

- Hamming distance between two vectors is the number of components in which they differ

- Or equivalently, given the number of variables p, and the number m of matching components, we define

$$d(x, y) = \frac{p - m}{p}$$

Examples of Hamming Distance

- Consider the same example

Outlook_overcast	Outlook_rainy	Outlook_sunny	Humidity	Windy
0	0	1	0	0
0	0	1	0	1

- They are at Jaccard distance of 0.2 since all their attribute values are the same except for Windy so they differ for one value over 5, that is 0.2

Outlook_overcast	Outlook_rainy	Outlook_sunny	Humidity	Windy	Temperature
0	0	1	0	0	1.0
0	0	1	0	1	1.0

Outlook_overcast	Outlook_rainy	Outlook_sunny	Humidity	Windy	Temperature
0	0	1	0	0	2.0
0	0	1	0	1	1.0

Hamming distance is 1/6 in the first case and 1/3 in the second case.

Cosine Distance

- The cosine distance between x, y is the angle that the vectors to those points make

$$d(x,y) = \arccos \frac{\sum_1^n x_i y_i}{\sqrt{\sum_i^n x_i^2}\sqrt{\sum_i^n y_i^2}}$$

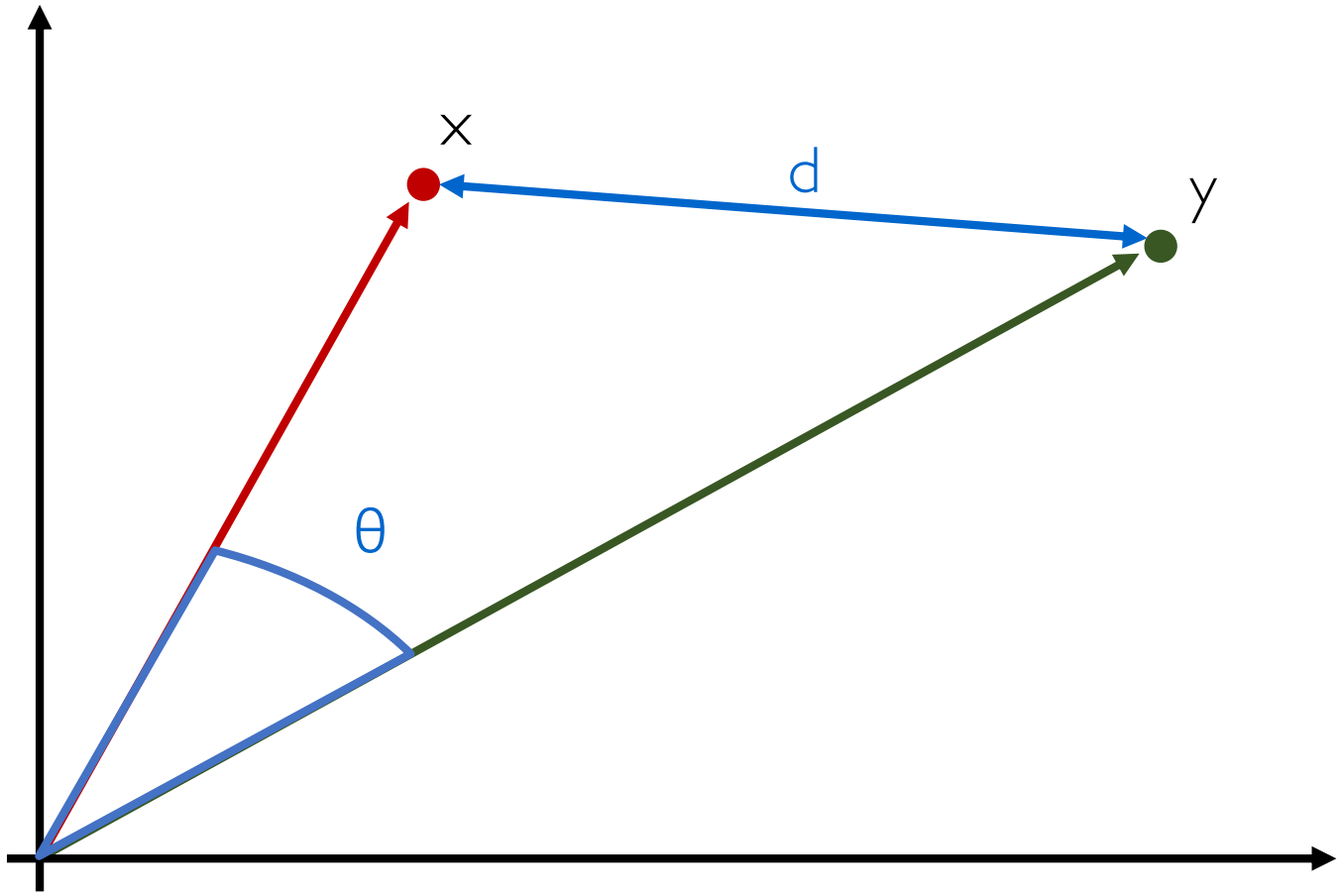
- This angle will be in the range 0 to 180 degrees, regardless of how many dimensions the space has.

- Example: given x = (1,2,-1) and y = (2,1,1) the angle between the two vectors is 60

Cosine Similarity

- The cosine similarity between x, y is simply computed as,

$$s(x,y) = \frac{\sum_1^n x_i y_i}{\sqrt{\sum_i^n x_i^2}\sqrt{\sum_i^n y_i^2}}$$



Cosine Distance θ compared to the Euclidean distance d

Edit Distance

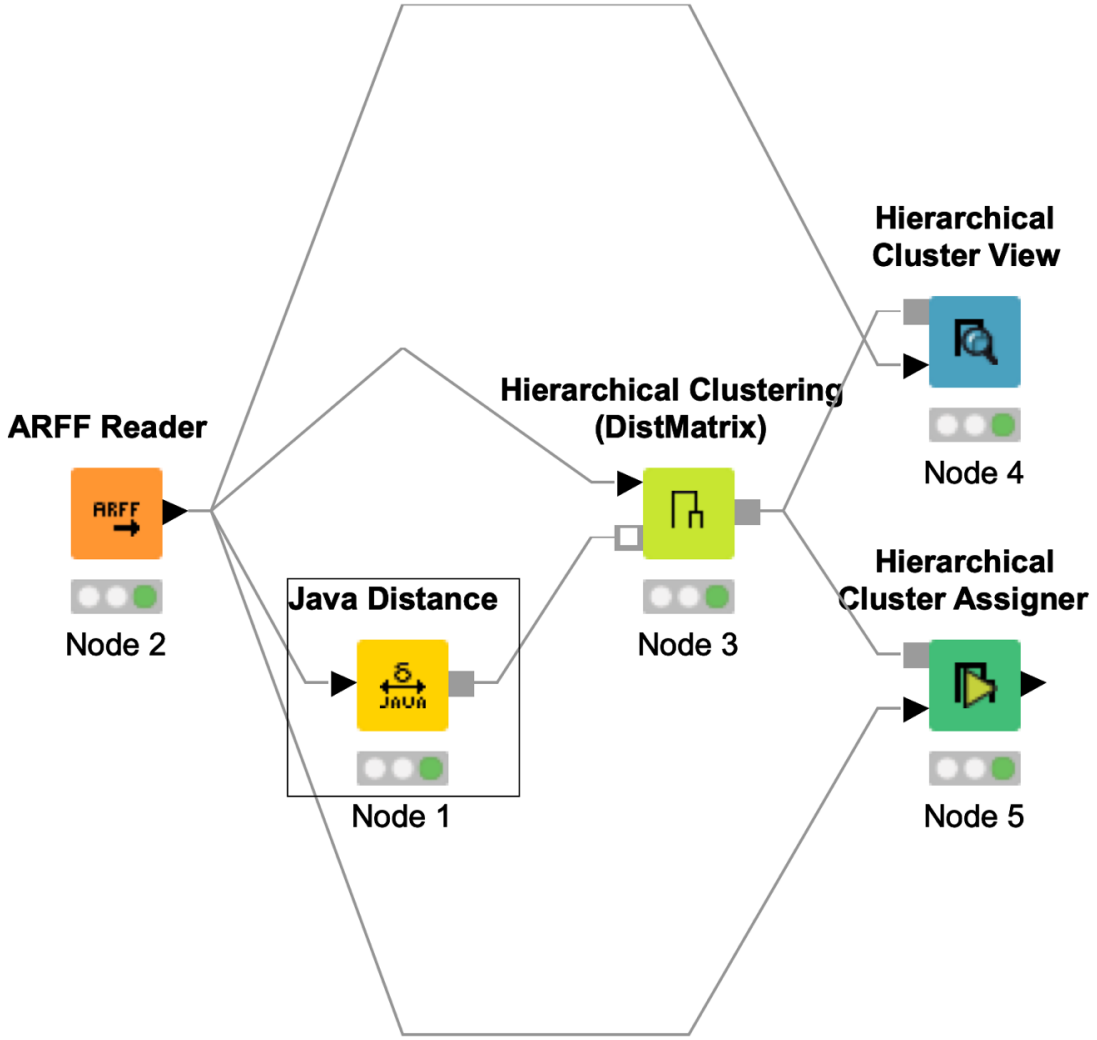
- The distance between a string x=x1x2…xn and y=y1y2…ym is the smallest number of insertions and deletions of single characters that will transform x into y

- Alternatively, the edit distance d(x, y) can be computed as the longest common subsequence (LCS) of x and y and then,

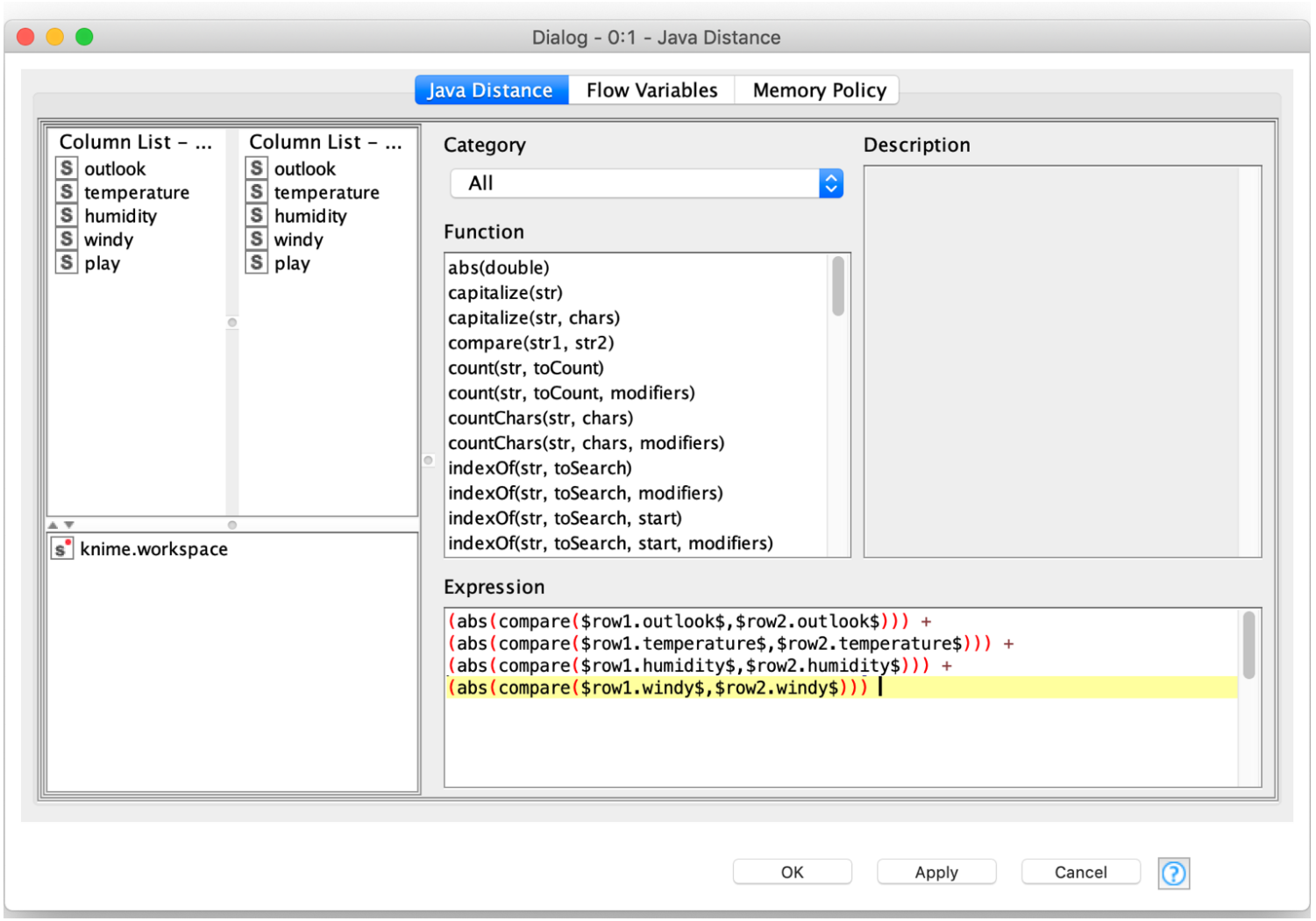
$$d(x,y) = |x| + |y| - 2|LCS|$$

- Example
  - The edit distance between **x=abcde** and **y=acfdeg** is 3 (delete b, insert f, insert g), the LCS is acde which is coherent with the previous result

Some tools allow users to define their own distance function



KNIME workflow applying hierarchical clustering with
a custom distance function implemented in Java



The definition of the custom distance function used in the previous KNIME workflow.

Other tools provide only the usual distance function (Euclidean, Manhattan, etc.)

When a custom function is not an option, we can try to transform the data

Normalization

Normalization and Other Issues

- Different attributes are measured on different scales, thus we often need to normalize them, using for instance range normalization o standard score normalization

$$x'_i = \frac{x_i - \min_i x_i}{\max_i x_i - \min_i x_i} \qquad x'_i = \frac{x_i - \mu}{\sigma}$$

- For nominal attributes, the distance either 0 (the value is the same) or 1 (the value is different)

- Missing values are usually assumed to be maximally distant (given normalized attributes)

Requirements on Clustering Algorithms

Requisites for Clustering Algorithms

- Scalability
- Ability to deal with different types of attributes
- Ability to handle dynamic data
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

Curse of Dimensionality

Curse of Dimensionality

in high dimensions, almost all pairs of points are equally far away from one another

almost any two vectors are almost orthogonal