

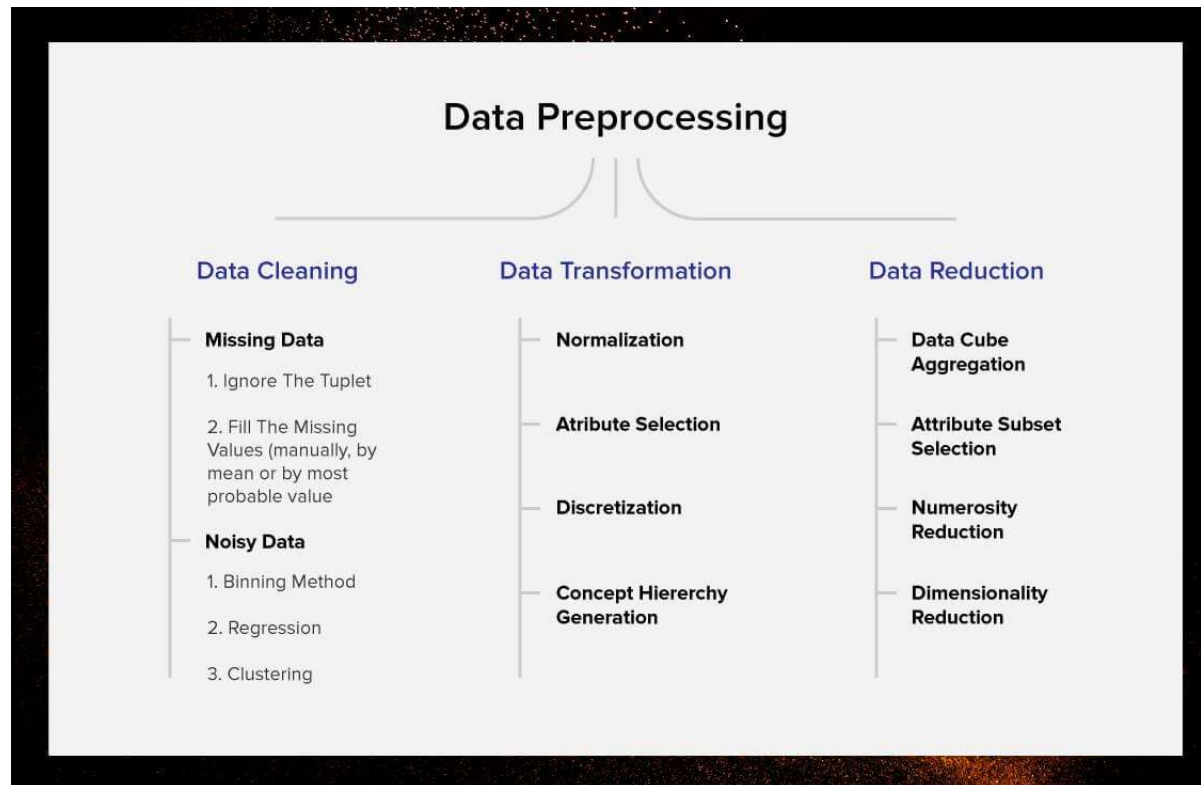


Business Analytics
M. Tech QROR – 2nd yr (2024)

Data Processing
Preparatory Analytics-02

Dr. Prasun Das
SQC & OR Unit
Indian Statistical Institute
e-mail: prasun@isical.ac.in

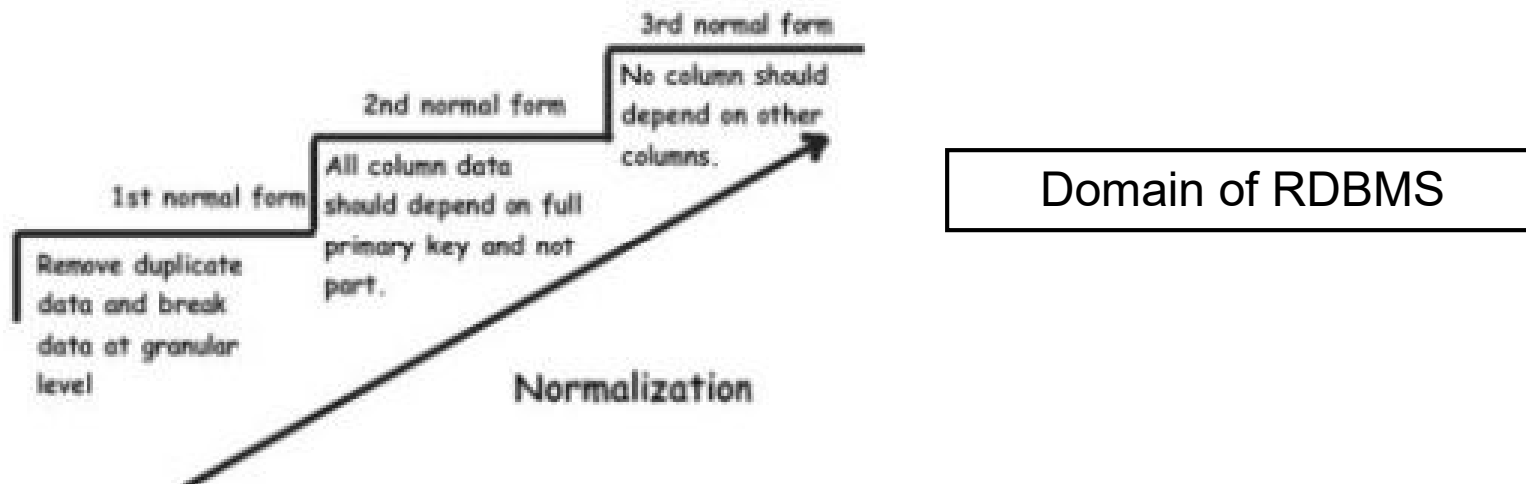
Data Transformation



Normalization

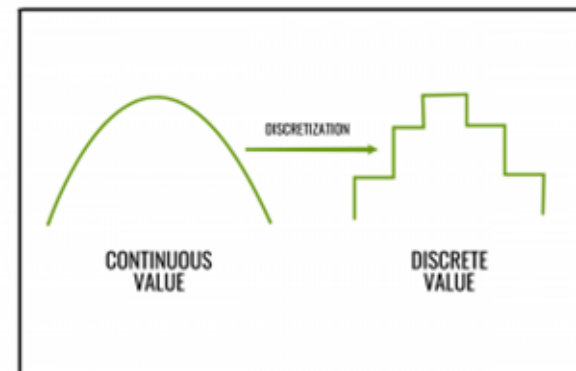
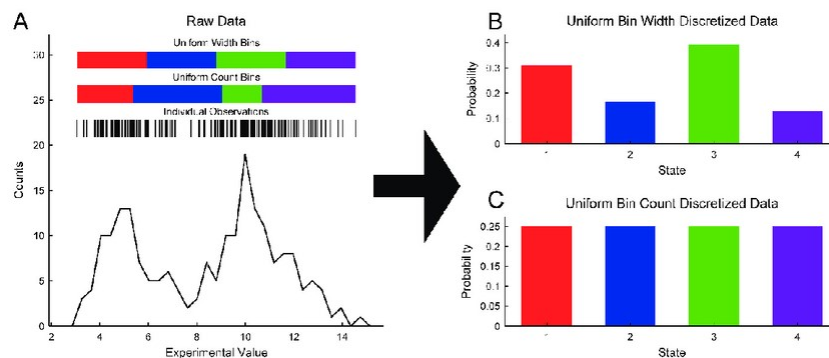
It is a database design technique used to design a **relational database** in accordance with a series of so-called **normal forms** to reduce **data redundancy (duplicacy)** and improve **data integrity**. Having data in unnormalized form and aiming to achieve the highest level of normalization, the stages of normal forms as **first normal form (1NF)**, **second normal form (2NF)**, **third normal form (3NF)** and so on. [Coded, E.F. (1970). *Communications of the ACM*, 13(6), 377-387]

Normalization entails organizing the **columns (attributes)** and **tables (relations)** of a database to ensure that their dependencies are properly enforced by database integrity.



Discretization (binning)

Binning or discretization is used **to transform a continuous or numerical variable into a categorical feature** by creating a set of contiguous intervals (or bins) that go across the range of our desired variable/model/function. Binning of continuous variables introduces non-linearity and tends to improve the performance of the model. Training a model with discrete data becomes faster and more effective than when attempting the same with continuous data. It can also be used to identify missing values or outliers.



Discretization (binning)

Two methods of Discretization:

1. Unsupervised Binning:

- a) **Equal Frequency Binning**: bins have an equal frequency.
- b) **Equal Width Binning** : bins have equal width with a range of each bin are defined as $[min + w]$, $[min + 2w]$ $[min + nw]$ where $w = (max - min) / (\# \text{ bins})$.

2. Supervised Binning:

Entropy-based Binning (a top down split point approach on entropy for bins and the best split provides majority of the values in a bin with same class label: decision tree, concept hierarchy)
[entropy(data tuples): $-\sum p_i \log p_i$ where, $i: 1, 2, \dots, m$: no. of classes]

Concept Hierarchy Generation

- Discretization can be performed rapidly on an attribute to provide a hierarchical partitioning of the attribute values, known as a **Concept Hierarchy**.
- Concept hierarchies can be used to reduce the data by replacing low-level concepts with higher-level concepts.
- In the multidimensional model, data are organized into multiple dimensions, and each dimension contains multiple levels of abstraction defined by concept hierarchies.
- Note: Data mining on a reduced data set means fewer input and output operations and is more efficient than mining on a larger data set.
- Because of these benefits, discretization techniques and concept hierarchies are typically applied before data mining, rather than during mining.

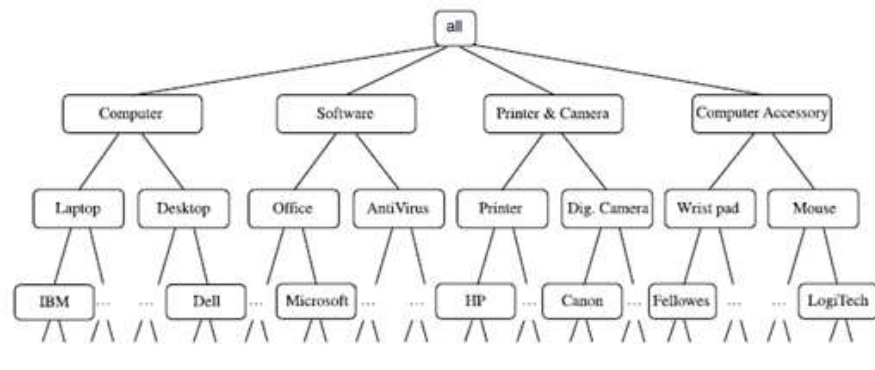
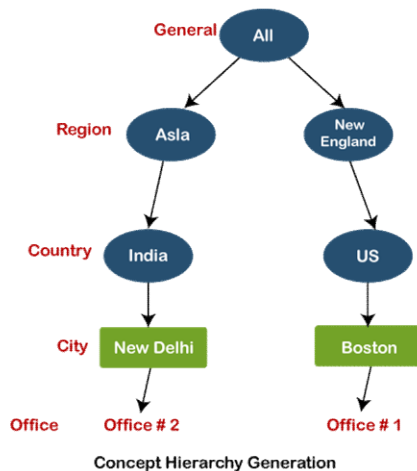


Figure 5.10 A concept hierarchy for AllElectronics computer items.

Data Reduction

Data reduction is **the process of reducing the amount of capacity (effective, not raw) required to store data** by reducing the size of data and makes it suitable and feasible for analysis. Data reduction can increase storage efficiency and reduce costs. In the reduction process, integrity of the data must be preserved and data volume is reduced. There are many techniques that can be used for data reduction.

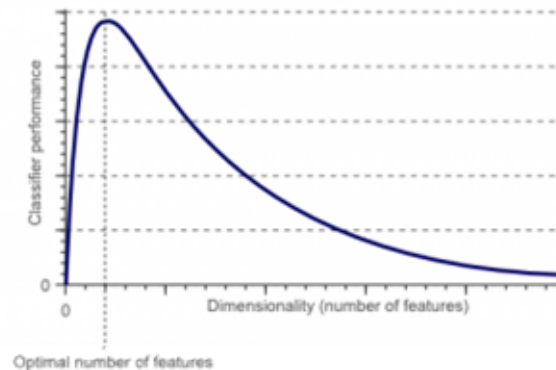
1. Dimensionality Reduction
2. Data Cube Aggregation
3. Data Compression

Dimensionality Reduction

Dimensionality reduction refers to the technique of reducing the dimension of a data feature set. Usually, machine learning datasets (feature set) contain hundreds of columns (i.e., features) or an array of points, creating a massive sphere in a three-dimensional space. By applying **dimensionality reduction**, the number of columns can be brought down to quantifiable counts, like transforming the three-dimensional sphere into a two-dimensional object (circle).

The Curse of Dimensionality

The curse of dimensionality is a phenomenon that arises when you work (analyze and visualize) with data in high-dimensional spaces that do not exist in low-dimensional spaces.



Dimensionality Reduction

(Importance)

The primary aim of dimensionality reduction is to avoid **overfitting**. A training data with considerably lesser features will ensure that your model remains simple – it will make smaller assumptions.

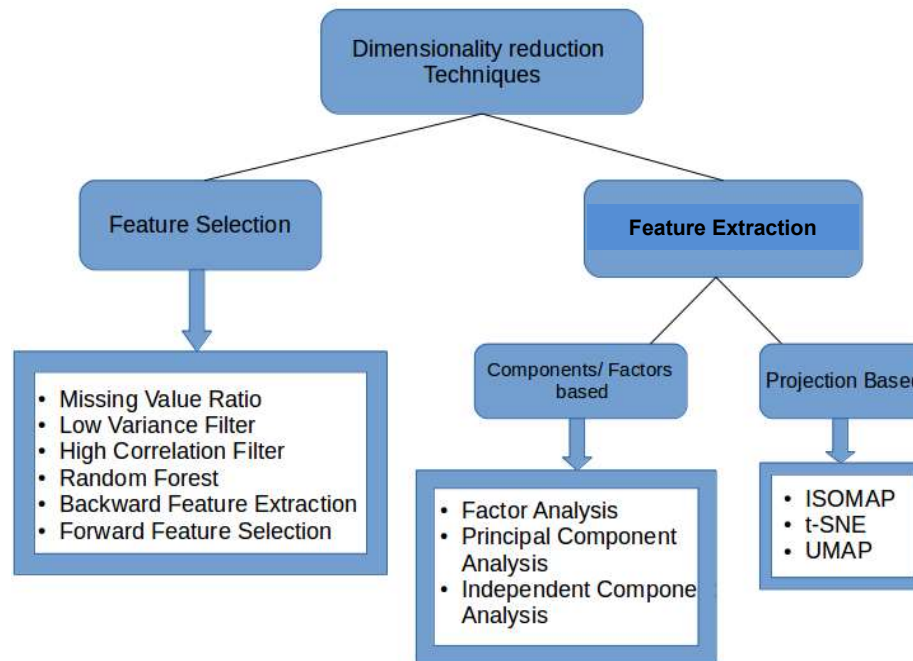
Benefits:

- It eliminates noise and takes care of multicollinearity by removing redundant (correlated) features.
- It helps improve the model's accuracy and performance.
- It facilitates the usage of algorithms that are usually unfit for more substantial dimensions.
- It reduces the amount of storage space required (less data needs lesser storage space).
- It compresses the data, which reduces the computation time and facilitates faster training.
- It helps in visualizing data at lower dimensions (2D or 3D) and allows us to plot and observe patterns more clearly.

Techniques: List

1. Missing Value Ratio
2. Low Variance Filter
3. High Correlation Filter
4. Random Forest (RF)
5. Backward Feature Elimination
6. Forward Feature Selection
7. Factor Analysis
8. Principal Component Analysis (PCA) – Linear/non-linear
9. Independent Component Analysis (ICA) – Linear/non-linear
10. Multi-dimensional Scaling (MDS) – non-linear
11. Isometric Feature Mapping (ISOMAP)
12. t-Distributed Stochastic Neighbor Embedding (t-SNE)
13. Uniform Manifold Approximation and Projection (UMAP)
14. Locally Linear Embedding (LLE) – non-linear
15. Linear Discriminant Analysis (LDA)
16. Kohonen's SOM
17. ANN (PNN, GRNN, PCNN)
18. Vector Quantization *and, many more*

Techniques: Categories



1. Missing Value Ratio

Set a **threshold value** and if the percentage of missing values in any variable is more than that threshold, we will **drop** the variable.

Else, **Impute** the missing data as described earlier.

2. Low Variance Filter

Consider a variable in the dataset where all the observations have the same value, say 1. This variable can't improve the target variable/model, because of having zero variance.

So, **calculate the variance of each variable/feature** and **drop the variables having low variance** as compared to other variables in the dataset.

3. High Correlation Filter

High correlation between two variables (likely to carry similar information) can bring down the performance of some models drastically (linear and logistic regression models, for instance). After calculating the correlation between independent numerical variables, if the **correlation coefficient** crosses a certain threshold value, we can drop one of the variables.

As a general guideline, we should keep those variables which show a decent or high correlation with the target variable. Generally, if the correlation between a pair of variables is greater than 0.5-0.6, we should seriously consider dropping one of those variables.

Keeping these variables will not affect the prediction performance much but reduces the precision of estimated coefficients of independent variables.

4. Backward Feature Selection

This method can be used while building Linear Regression or Logistic Regression models.

Steps:

- Compute the performance of the model, while training, after eliminating each of the n features every time (n times, in total) and training the model on the remaining $(n-1)$ features.
- Identify the feature whose removal indicates the minimum (or, no) change in the performance of the model, and then drop that feature
- Repeat this process until no variable can be dropped.

5. Forward Feature Selection

This is the reverse process of the Backward Feature Elimination, ie., instead of eliminating features, we try to find the best features which improve the performance of the model.

Steps:

- Start with a single feature. Essentially, we train the model n number of times using each feature separately. The feature giving the best performance is selected as the starting variable
- Repeat this process and add one variable at a time. The variable that produces the highest increase in performance is retained.
- Repeat this process until no significant improvement is seen in the model's performance

NOTE: Both Backward and Forward Feature Selection are time consuming and computationally expensive. They are practically only used on datasets that have a small number of input variables.

6. Factor Analysis

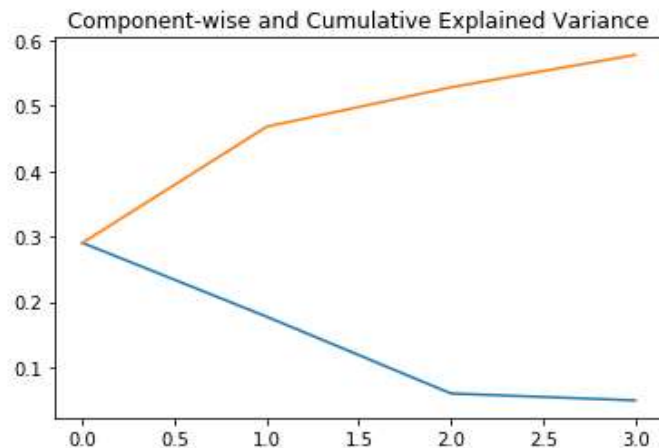
In Factor Analysis technique, variables are grouped by their correlations (say, for ex., Income and Education), i.e., all variables in a particular group will have a high correlation among themselves, but a low correlation with variables of other group(s). Here, each group is known as a **factor**. These factors are small in numbers as compared to the original dimensions of the data.

[Read more as a Multivariate Analysis Technique](#)

7. Principal Component Analysis

PCA is a technique which helps us in **extracting** a new set of variables from an existing large set of variables. These newly extracted variables are called Principal Components. Some of the key points about PCA are as follows:

- A Principal Component (PC) is a linear combination of the original variables
- PCs are extracted in such a way that the 1st PC explains maximum variance in the dataset
- 2nd PC tries to explain the remaining variance in the dataset and is uncorrelated to the 1st PC
- 3rd PC tries to explain the variance which is not explained by the first two PCs and so on...

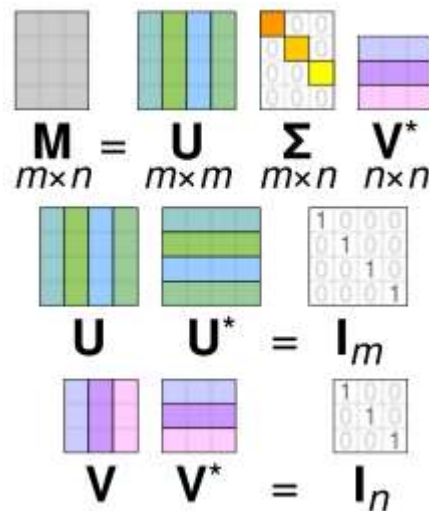


Read more as a [Multivariate Analysis Technique](#)

8. Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is essentially used to remove redundant features from the dataset.

SVD decomposes the original variables into three constituent matrices, using the concept of Eigen values and Eigenvectors.

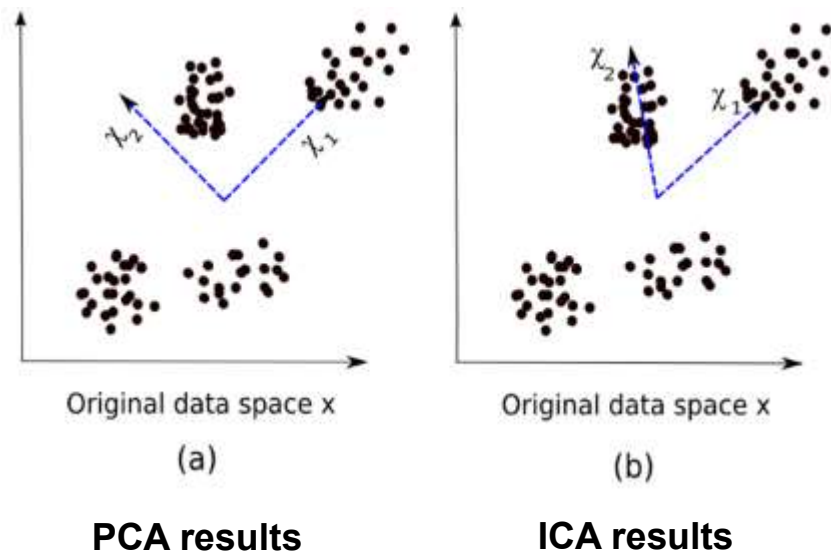


The diagram illustrates the SVD decomposition of matrix M into matrices U , Σ , and V^* . The decomposition is shown as $M = U \Sigma V^*$ with dimensions $m \times n = m \times m \times m \times n \times n \times n$. Below this, the matrices U and U^* are shown as $U U^* = I_m$, and V and V^* are shown as $V V^* = I_n$. The matrices are represented by colored blocks: U is a 4x4 grid of green and blue blocks, U^* is a 4x4 grid of green and blue blocks, V is a 4x4 grid of purple and pink blocks, and V^* is a 4x4 grid of purple and pink blocks. The identity matrices I_m and I_n are shown as 4x4 grids of 1s and 0s.

Read more as a Linear Algebra Technique

9. Independent Component Analysis (ICA)

Independent Component Analysis (ICA) is based on information-theory. The major difference between PCA and ICA is that PCA looks for uncorrelated factors while ICA looks for independent factors.

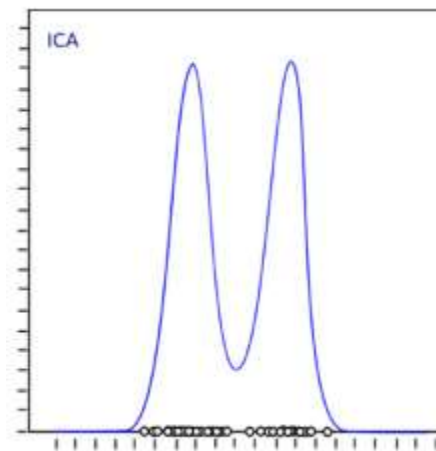
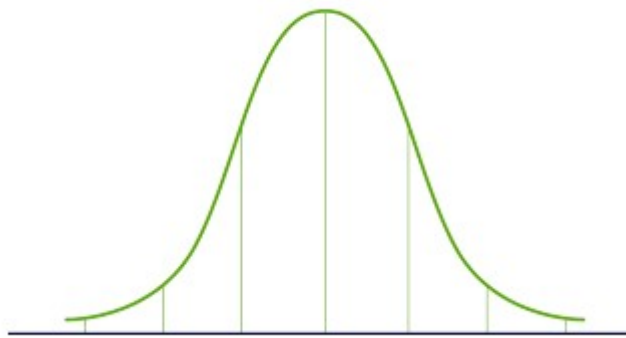


Note: If two variables are uncorrelated (no linear relation), they may not be independent (non-linear relation). However, if they are independent, then they are uncorrelated too.

9. Independent Component Analysis (ICA)

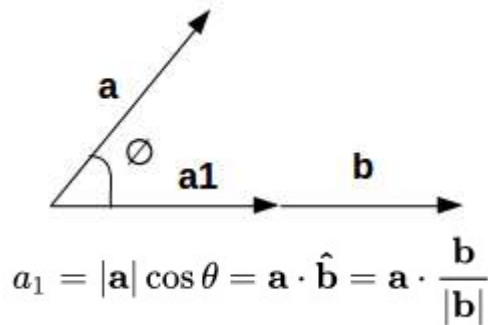
This algorithm assumes that the given variables are linear mixtures of some unknown latent variables. It also assumes that these latent variables are **mutually independent**, i.e., they are not dependent on other variables and hence, they are called the independent components of the observed data.

Most common method to measure independence of components is Non-Gaussianity. As per the central limit theorem, distribution of the sum of independent components tends to be normally distributed (Gaussian). So we can look for the transformations that maximize the kurtosis of each component of the independent components. Maximizing the kurtosis will make the distribution non-gaussian and hence we will get independent components.



Methods based on Projections

Projection of **a** onto **b**:



By projecting one vector onto the other, dimensionality can be reduced.

In projection techniques (**Local/Global**), multi-dimensional data is represented by projecting its points onto a lower-dimensional space.

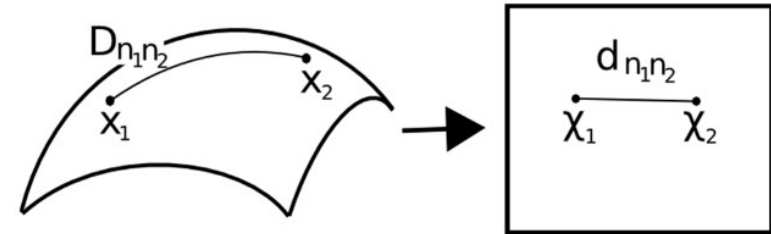
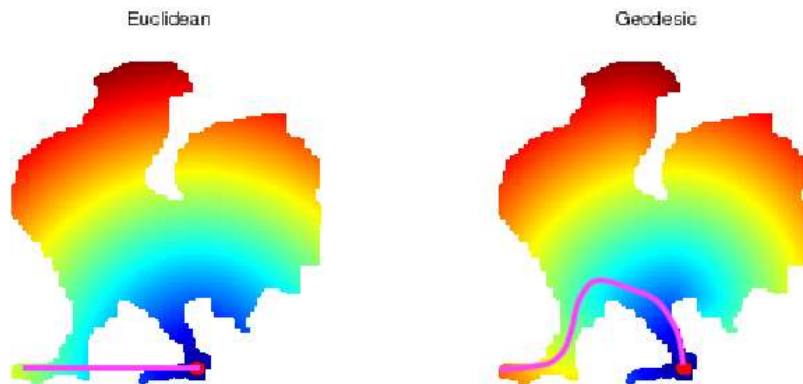
For an n -dimensional curve, small flat pieces are manifolds and a combination of these manifolds will give us the original n -dimensional curve (original data).

Local Approach: They map nearby points on the manifold to nearby points in the low dimensional representation.

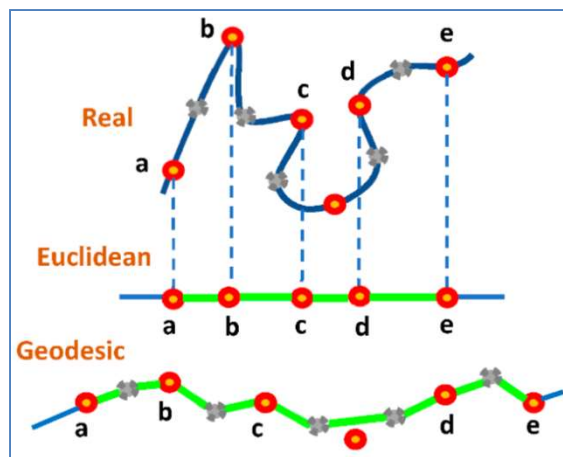
Global Approach: They attempt to preserve geometry at all scales, i.e. mapping nearby points on manifold to nearby points in low dimensional representation as well as far away points to far away points.

10. Isometric Feature Mapping (ISOMAP)

Geodesic distance is the distance between two points following the path available/possible between the two points whereas **Euclidean distance** doesn't have a path constraint to follow. It is the length of a straight line from point 'a' to 'b'.



$D_{n_1 n_2}$ = Geodesic distance between x_1 and x_2
 $d_{n_1 n_2}$ = Euclidean distance between x_1 and x_2



Dimension Reduction - Two families :

Linear projection of data (PCA: Euclidean distance).

Non-linear projection of data (ISOMAP: Geodesic distance) - preserves much more information than linear representation

10. Isometric Feature Mapping (ISOMAP)

ISOMAP is a manifold learning algorithm that tries to preserve the geodesic distance between samples while reducing the dimension, i.e., to recover full low-dimensional representation of a non-linear differentiable (smooth) manifold. [[Tenebaum et al., A global geometric framework for nonlinear dimensionality reduction. Science **2000**, 290, 2319–2323](#)]

Assumption: for any pair of points on manifold, the geodesic distance (shortest distance on a curved surface) between the two points is equal to the Euclidean distance (shortest distance on a straight line).

10. Isometric Feature Mapping (ISOMAP)

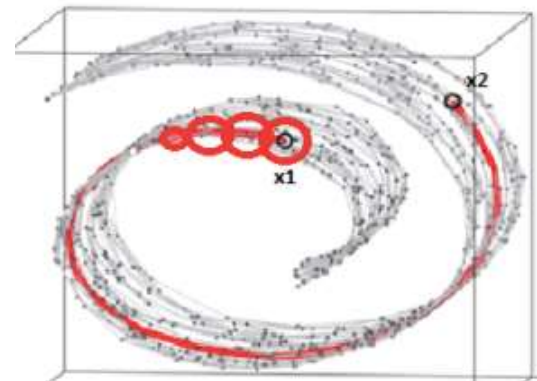
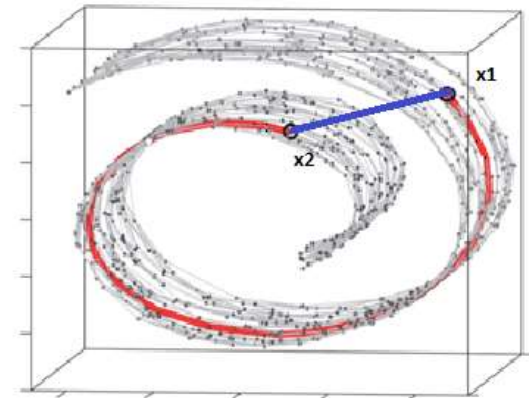
Ex.: data forms a spiral shape (nonlinear). The red line determines the geodesic distance between x_1 & x_2 while the blue line represents Euclidean distance.

1. **Run k-NN** (using Euclidean distance) to form a neighborhood graph/adjacency matrix.

Figure out 'n' (3 to 4) neighbors for all points & form a 'neighborhood' graph/adjacency matrix to measure the geodesic distance between any two points.

2. **Calculate geodesic distance** using the graph and any **shortest path algorithm**

The shortest distance will become the sum of weights of all the points lying on the red spirals between x_1 & x_2 , in the shortest path, hence giving geodesic distance (approx) between x_1 & x_2 .



10. Isometric Feature Mapping (ISOMAP)

3. Form a **dissimilarity matrix** using the above-calculated geodesic distance between points and **Square** it.

4. **Eigen decomposition & choosing 'k' eigenvectors.** This is something similar to what we do in PCA after calculating the correlation matrix.

Algorithm 1: ISOMAP

Step 1. construct the matrix of squared pairwise similarities \mathbf{D} , $D_{ij} = \left\| y_i - y_j \right\|^2$, the distance matrix, measured on temporal dimension. The i, j are temporal indexes.

Step 2. build the weighted graph based on the \mathbf{D} according to how many neighbors of each point

Step 3. estimate the geodesic distances \mathbf{D}_G by finding the shortest paths on the weighted graph (Dijkstra's algorithm [20])

Step 4. define $\mathbf{B} = -1/2 \mathbf{J} \mathbf{D}_G \mathbf{J}^T$, where $\mathbf{J} = \mathbf{I} - 1/N$, \mathbf{I} is the identity matrix, and N is the number of data points

Step 5. solve eigen problem $\mathbf{B}\mathbf{P} = \mathbf{P}\Sigma$

Step 6. computing the leading principal vectors by $\mathbf{X} = \mathbf{P}\Sigma^{1/2}$

11. t- Distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE can easily search for patterns (better than PCA) in a non-linear way with the help of mainly two types of approaches:

t-SNE is one of the few algorithms which is capable of retaining both local and global structure of the data at the same time. It calculates the probability of similarity of points in high dimensional space as well as in low dimensional space.

11. t- Distributed Stochastic Neighbor Embedding (t-SNE)

High-dimensional Euclidean distances between data points x_i and x_j are converted into conditional probabilities that represent similarities:

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)}$$

where, $\|x_i - x_j\|$ represents the Euclidean distance between x_i and x_j , and σ_i is the variance of data points in high dimensional space.

For the **low-dimensional** data points y_i and y_j corresponding to the high-dimensional data points x_i and x_j , it is possible to compute a similar conditional probability using:

$$q_{j|i} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2)}$$

where, $\|y_i - y_j\|$ represents the Euclidean distance between y_i and y_j .

After calculating both the probabilities, it minimizes the difference between both the probabilities.

12. UMAP

Uniform Manifold Approximation and Projection (UMAP) can preserve as much of the local, and more of the global data structure as compared to t-SNE, with a shorter runtime. It can handle large datasets and high dimensional data without too much difficulty. The correlation between the components obtained from UMAP is **quite less as compared** to the correlation between the components obtained from t-SNE. The visualization power is also quite high.

Method:

Step-1. It first calculates the distance between the points in high dimensional space, projects them onto the low dimensional space, and calculates the distance between points in this low dimensional space using k-NN.

Step-2. It then optimizes the results (minimizes the difference between these distances) using Scaled Conjugate Gradient (SCG) algorithm.

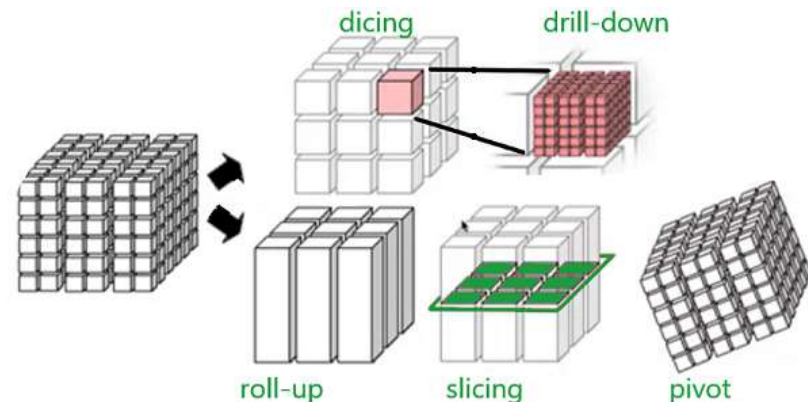
Data Cube Aggregation

Aggregation in data mining is the process of finding, collecting, and presenting the data in a summarized format to perform statistical analysis of business problems.

Data Cube Aggregation is a multi-dimensional aggregation that represents the original data set by aggregating at multiple layers of a data cube, resulting in data reduction.

Problem: “Find the total sales in 2010, broken down by *item*, *region*, and *month*, with subtotals for each dimension.” To answer this query, a traditional data cube is constructed that aggregates the total sales at the following eight different granularity levels: $\{(item, region, month), (item, region), (item, month), \dots\}$

The Data cube **pictorially shows how different attributes of data are arranged in the data model**. Below is the diagram of a general data cube. The example above is a 3D cube having attributes like *region* (A,B,C,D), *item type* (home, entertainment, computer, phone, security), *month* (Jan, Feb, ...,Dec)



Data Compression

Data compression is **a reduction in the number of bits needed to represent data**. Compressing data can save storage capacity, speed up file transfer, and decrease costs for storage hardware and network bandwidth. For example, **in a 2:1 compression ratio, a 20 MB file takes up 10 MB of space**.

Two types of data compression techniques — **Lossy & Lossless**. **[ASSIGNMENT]**

In Lossy compression, the insignificant/redundant piece of data is removed quickly to reduce the size of (video, image & even audio) but retain the information.

In Lossless compression, data is not removed, but transformed through encoding for size reduction of (text, numbers).