# PHBS Course : Machine Learnings and Algorithms
# Data Preprocessing

Zhen Zhang



## Outlines





## Data Types

- Tabular data : matrices, vectors, objects, relations, etc.
  - Data objects : also called samples, examples, instances, data points, objects, tuples, vectors
  - Attributes : each row of a table, also called dimensions, features, variables
- Graphical data : networks, graphs, etc.
- Multi-media data : texts, images, videos, audios, etc.

| 行星 | 周期（年） | 平均距离 | 周期$^2$/距离$^3$ |
|---|---|---|---|
| 水星 | 0.241 | 0.39 | 0.98 |
| 金星 | 0.615 | 0.72 | 1.01 |
| 地球 | 1.00 | 1.00 | 1.00 |
| 火星 | 1.88 | 1.52 | 1.01 |
| 木星 | 11.8 | 5.20 | 0.99 |
| 土星 | 29.5 | 9.54 | 1.00 |
| 天王星 | 84.0 | 19.18 | 1.00 |
| 海王星 | 165 | 30.06 | 1.00 |



## Types of Attributes

- Discrete : $\mathbf{x} \in$ some countable sets, e.g., $\mathbb{N}$
  - Nominal : Countries={China, US, UK, France, Germany}, Universities={Peking U, Tsinghua U, SUSTech, Shenzhen U, HIT}, not comparable
  - Boolean : 0 or 1, male or female, spam or non-spam, etc.
  - Ordinal : Heights={tall, short}, Scores={A+, A, A-, B+, B, B-, C, C-, D, F}, can be compared, but cannot operated arithmetically
- Continuous : $\mathbf{x} \in$ some subset in $\mathbb{R}^n$
  - Numerical : Income, exact marks, weights, etc., can be operated arithmetically



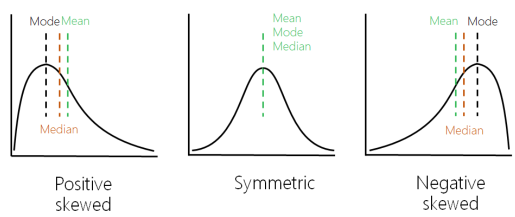
## Basic Statistics

- Mean : $\mathrm{E}X = \min_c \mathrm{E}(X-c)^2 \approx \frac{1}{n}\sum_{i=1}^n x_i$
- Median :
$$\min_c \mathrm{E}|X-c| = \begin{cases} x_{(\frac{n+1}{2})} & \text{if } n \text{ is odd} \\ (x_{(\frac{n}{2})} + x_{(\frac{n}{2}+1)})/2 & \text{otherwise} \end{cases}$$
- Maximum : $\max_i x_i$ ; Minimum : $\min_i x_i$
- Quantile : a generalization of median, $k$-th $q$-quantile $x_q$ : $P[X < x_q] \leqslant k/q$ ; interquartile range (IQR)=$Q_3(75\%) - Q_1(25\%)$
- Variance : $\mathrm{Var}(X) = \mathrm{E}[X - \mathrm{E}X]^2 \approx \frac{1}{n}\sum_{i=1}^n (x_i - \bar{x})^2$ ; Standard deviation : $\sqrt{\mathrm{Var}(X)}$
- Mode : $\min_c \mathrm{E}|X-c|^0$ = the most frequently occurring value (define $0^0 = 0$)



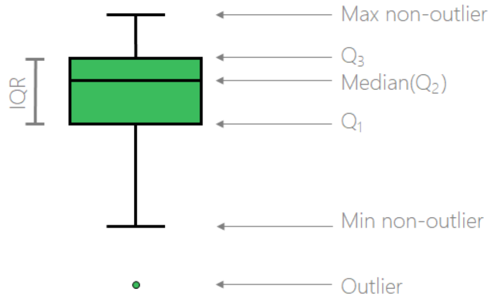
## Central Tendency

For one-peak skewed density distribution, empirical formula : $Mean - Mode = 3 \times (Mean - Median)$



## Box Plot

Measure the dispersion of data






## Metrics

- Proximity :
  - Similarity : range is $[0, 1]$
  - Dissimilarity : range is $[0, \infty]$, sometimes distance
- For nominal data, $d(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_k I(x_{i,k} \neq x_{j,k})}{p}$ ; or one-hot encoding into Boolean data
- For Boolean data, symmetric distance $d(\mathbf{x}_i, \mathbf{x}_j) = \frac{r+s}{q+r+s+t}$ or Rand index $Sim_{Rand}(\mathbf{x}_i, \mathbf{x}_j) = \frac{q+t}{q+r+s+t}$ ; non-symmetric distance $d(\mathbf{x}_i, \mathbf{x}_j) = \frac{r+s}{q+r+s}$ or Jaccard index $Sim_{Jaccard}(\mathbf{x}_i, \mathbf{x}_j) = \frac{q}{q+r+s}$

| | | Sample $j$ | | |
|---|---|---|---|---|
| | | 1 | 0 | sum |
| Sample $i$ | 1 | $q$ | $r$ | $q+r$ |
| | 0 | $s$ | $t$ | $s+t$ |
| | sum | $q+s$ | $r+t$ | $p$ |



## Metrics : Distance

- Example : Let $H = F = 1$ and $L = S = 0$, $d(LandRover, Jeep) = \frac{1+0}{4+1+0} = 0.20$, $d(LandRover, TOYOTA) = \frac{3+1}{1+3+1} = 0.80$, $d(Jeep, TOYOTA) = \frac{3+2}{1+3+2} = 0.83$
- Minkowski distance : $d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt[h]{\sum_{k=1}^{p} |x_{ik} - x_{jk}|^h}$ is $L_h$-norm
  - Positive definiteness $d(\mathbf{x}_i, \mathbf{x}_j) \geqslant 0$ and "$=$" if and only if $i = j$ ;
  - Symmetry $d(\mathbf{x}_i, \mathbf{x}_j) = d(\mathbf{x}_j, \mathbf{x}_i)$ ;
  - Triangle inequality $d(\mathbf{x}_i, \mathbf{x}_j) \leqslant d(\mathbf{x}_i, \mathbf{x}_k) + d(\mathbf{x}_k, \mathbf{x}_j)$

| | Weight | Price | Acceleration | MPG | Quality | Sales Volume |
|---|---|---|---|---|---|---|
| Land Rover | $H$ | $H$ | $F$ | $H$ | $H$ | $L$ |
| Jeep | $H$ | $H$ | $S$ | $H$ | $H$ | $L$ |
| TOYOTA | $L$ | $L$ | $F$ | $L$ | $H$ | $H$ |

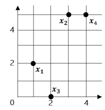
| | | Jeep | |
|---|---|---|---|
| | | 1 | 0 |
| Land Rover | 1 | $q=4$ | $r=1$ |
| | 0 | $s=0$ | $t=1$ |



## Metrics : Distance (Cont')

- Manhattan distance : $h = 1$, and $d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{p} |x_{ik} - x_{jk}|$
- Euclidean distance : $h = 2$, and $d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{p} |x_{ik} - x_{jk}|^2}$
- Supremum distance : $h = \infty$, and $d(\mathbf{x}_i, \mathbf{x}_j) = \max_{k=1}^{p} |x_{ik} - x_{jk}|$

| Point | Attr 1 | Attr 2 |
|---|---|---|
| $x_1$ | 1 | 2 |
| $x_2$ | 3 | 5 |
| $x_3$ | 2 | 0 |
| $x_4$ | 4 | 5 |




| $L_1$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|---|
| $x_1$ | 0 | | | |
| $x_2$ | 5 | 0 | | |
| $x_3$ | 3 | 6 | 0 | |
| $x_4$ | 6 | 1 | 7 | 0 |

(a) Manhattan

| $L_2$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|---|
| $x_1$ | 0 | | | |
| $x_2$ | 3.61 | 0 | | |
| $x_3$ | 2.24 | 5.1 | 0 | |
| $x_4$ | 4.24 | 1 | 5.39 | 0 |

(b) Euclidean

| $L_\infty$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|---|
| $x_1$ | 0 | | | |
| $x_2$ | 3 | 0 | | |
| $x_3$ | 2 | 5 | 0 | |
| $x_4$ | 3 | 1 | 5 | 0 |

(c) Supremum



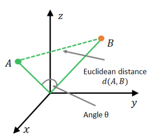
## Metrics : Cosine Similarity

- Definition : $\cos(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{k=1}^{p} x_{ik} x_{jk}}{\sqrt{\sum_{k=1}^{p} x_{ik}^2} \sqrt{\sum_{k=1}^{p} x_{jk}^2}} = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}$
- Example : $\cos(\mathbf{x}_1, \mathbf{x}_2) = 0.94$

| Instance | Team | Coach | Hockey | Baseball | Soccer | penalty | Score | Win | Loss | Season |
|---|---|---|---|---|---|---|---|---|---|---|
| Instance 1 | 5 | 0 | 3 | 0 | 2 | 0 | 0 | 2 | 0 | 0 |
| Instance 2 | 3 | 0 | 2 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |

Euclidean vs. Cosine :

- Euclidean : measures the distance in absolute value, many applications
- Cosine : insensitive to absolute value, e.g., analyze users' interests based on movie ratings






## Metrics : Other Distances

- For ordinal data, mapping the data to numerical data : $X = \{x_{(1)}, x_{(2)}, \ldots, x_{(n)}\}$, $x_{(i)} \mapsto \frac{i-1}{n-1} \in [0, 1]$
- For mixed type, use weighted distance with prescribed weights :

$$d(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{g=1}^{G} w_{ij}^{(g)} d_{ij}^{(g)}}{\sum_{g=1}^{G} w_{ij}^{(g)}}$$

Put the attributes of the same type into groups, for each data type $g$, use the corresponding distance $d_{ij}^{(g)}$

## Outlines

Basic Concepts

Data Preprocessing

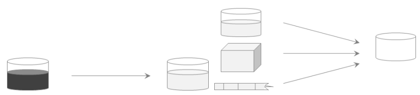


## Why Data Preprocessing ?

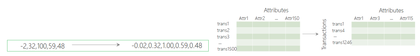
- Missing values
- Noisy with outliers
- Inconsistent representations
- Redundancy
- Errors may come during data input, data gathering, and data transferring
- Errors occur in about 5% of the data



## Four Types of Data Preprocessing



(a) Data cleaning (b) Data integration

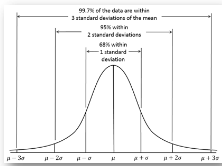


(c) Data conversion (d) Data reduction



## Data Scaling

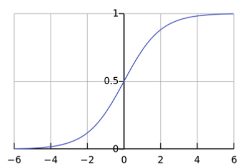
- Why scaling :
  - For better performance : e.g., RBF in SVM and penalty in Lasso/ridge regression assume the zero mean and unit variance
  - Normalize different dimensions : many algorithms are sensitive to the variables with large variances, e.g., height (1.75m) and weight (70kg) in distance calculation
- Z-score scaling : $x_i^* = \frac{x_i - \hat{\mu}}{\hat{\sigma}}$, $\hat{\mu}$ : sample mean, $\hat{\sigma}$ : sample variance, applicable if max and min are unknown and the data distributes well






## Data Scaling (Cont')

- 0-1 scaling : $x_i^* = \frac{x_i - \min_k x_k}{\max_k x_k - \min_k x_k} \in [0, 1]$, applicable for bounded data sets, need to recompute the max and min when new data arrive
- Decimal scaling : $x_i^* = \frac{x_i}{10^k}$, applicable for data varying across many magnitudes
- Logistic scaling : sigmoid transform $x_i^* = \frac{1}{1+e^{-x_i}}$, applicable for data concentrating nearby origin






## Data Discretization

- Why discretization :
  - Improve the robustness : removing the outliers by putting them into certain intervals
  - For better interpretation
  - Reduce the storage and computational power
- Unsupervised discretization : equal-distance discretization, equal-frequency discretization, clustering-based discretization, $3\sigma$-based discretization
- Supervised discretization : information gain based discretization, $\chi^2$-based discretization

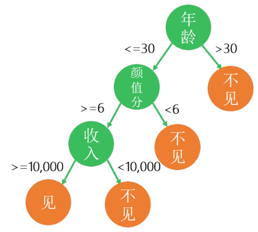
## Unsupervised Discretization

- Equal-distance discretization : split the range to $n$ intervals (bins) with the same length, group the data into each bin, sensitive to outliers
- Equal-frenquency discretization : group the data into $n$ subset so that each subset has the same number of points, tend to separate samples with similar values and produce uniform distribution
- Clustering-based discretization : do hierarchical clustering and form a hierarchical structure (e.g., using $K$-Means), and put the samples in the same branch into the same interval (a natural example is family tree)
- $3\sigma$-based discretization : put the samples into 8 intervals, need to take logarithm first



## Supervised Discretization - Information Gain

- Top-down splitting, similar to create a decision tree
- Do a decision tree classification using information gain, find a proper splitting point for each continuous variable such that the information gain increases the most
- The final leaf nodes summarize the discrete intervals






## Supervised Discretization - ChiMerge

- Bottom-up : similar to hierarchical clustering
- $\hat{\chi}^2$ statistics proposed by Karl Pearson, is used to test whether the observations dramatically deviate from theoretical distribution : $\hat{\chi}^2 = \sum_{i=1}^{k} \frac{(A_i - \mathbb{E}A_i)^2}{\mathbb{E}A_i} = \sum_{i=1}^{k} \frac{(A_i - np_i)^2}{np_i}$, where $n_i$ is the number of samples in the $i$-th interval $A_i = [a_{i-1}, a_i]$ (frequency of observations), $\bigcup_{i=1}^{k} A_i$ covers the range of the variable, and $\mathbb{E}A_i = p_i$ is its expectation computed from the theoretical distribution ; it can be shown that $\hat{\chi}^2 \to \chi^2_{k-1}$
- ChiMerge : Given a threshold level $t$,
  1. Treat each value of the continuous variable as an interval and sort them in increasing order ;
  2. For each pair of adjacent intervals, compute its $\hat{\chi}^2$ statistics, if $\hat{\chi}^2 < t$, merge them into a new interval ;
  3. Repeat the above steps until no adjacent intervals can be merged.
- Two shortcomings : $t$ is hard to set appropriately ; too long loop for large sample set, computationally intensive



## ChiMerge : Iris Data Example

- $\hat{\chi}^2 = \sum_{i=1}^{m} \sum_{j=1}^{k} \frac{(A_{ij} - E_{ij})^2}{E_{ij}}$, where
  $m = 2$ (two adjacent intervals)
  $k$ is the number of classes
  $A_{ij}$ is the number of samples in $i$-th interval and in class $k$
  $R_i = \sum_{j=1}^{k} A_{ij}$ is the total number of samples in $i$-th interval
  $C_j = \sum_{i=1}^{m} A_{ij}$ is the total number of samples in class $j$
  $N = \sum_{i=1}^{m} \sum_{j=1}^{k} A_{ij}$ is the total number of samples
  $E_{ij} = R_i \cdot \frac{C_j}{N}$
- $\chi^2$ of 4.3 and 4.4 : $C_1 = 4$, $C_2 = 0$, $C_3 = 0$, $N = 4$, $A_{11} = 1$, $A_{12} = A_{13} = 0$, $A_{21} = 3$, $A_{22} = A_{23} = 0$, $R_1 = 1$, $R_2 = 3$, $E_{11} = 1$, $E_{12} = E_{13} = 0$, $E_{21} = 3$, $E_{22} = E_{23} = 0$, $\hat{\chi}^2 = 0$.

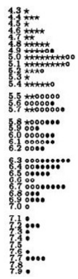


FIGURE: Sepal lengths of 3 types of iris



## ChiMerge Results

Left : significance level is 0.5 and the threshold for $\chi^2$ is 1.4 ;
Right : significance level is 0.9 and the threshold for $\chi^2$ is 4.6 ;
The final results keep the intervals with $\chi^2$ larger than the thresholds

| Int | Class frequency | | | $\chi^2$ |
|---|---|---|---|---|
| 4.3 | 16 | 0 | 0 | 4.1 |
| 4.9 | 4 | 1 | 1 | 2.4 |
| 5.0 | 25 | 5 | 0 | 8.6 |
| 5.5 | 2 | 5 | 0 | 2.9 |
| 5.6 | 0 | 5 | 1 | 1.7 |
| 5.7 | 2 | 5 | 1 | 1.8 |
| 5.8 | 1 | 3 | 3 | 2.2 |
| 5.9 | 0 | 12 | 7 | 4.8 |
| 6.3 | 0 | 6 | 15 | 4.1 |
| 6.6 | 0 | 2 | 0 | 3.2 |
| 6.7 | 0 | 5 | 10 | 1.5 |
| 7.0 | 0 | 1 | 0 | 3.6 |
| 7.1 | 0 | 0 | 12 | |

| Int | Class frequency | | | $\chi^2$ |
|---|---|---|---|---|
| 4.3 | 45 | 6 | 1 | 30.9 |
| 5.5 | 4 | 15 | 2 | 6.7 |
| 5.8 | 1 | 15 | 10 | 4.9 |
| 6.3 | 0 | 14 | 25 | 5.9 |
| 7.1 | 0 | 0 | 12 | |

Figure 2: ChiMerge discretizations for *sepal-length* at the .50 and .90 significance levels ($\chi^2 = 1.4$ and 4.6)



## Data Redundancy

- When strong correlations exist among different attributes, then we say that the some attributes can be derived from the others (Recall linear dependency for vectors)
- E.g., two attributes "Age" and "Birthday", then "Age" can be calculated from "Birthday"
- Determine the data redundancy by correlation analysis
- For continuous variables $A$ and $B$, compute the correlation coefficient $\rho_{A,B} = \frac{\sum_{i=1}^{k}(a_i - \bar{A})(b_i - \bar{B})}{k\hat{\sigma}_A\hat{\sigma}_B} \in [-1, 1]$ :
  1. If $r > 0$, $A$ and $B$ are positively correlated ;
  2. If $r < 0$, $A$ and $B$ are negatively correlated ;
  3. If $r = 0$, $A$ and $B$ are uncorrelated.

  Note that the correlation between $A$ and $B$ does not imply the causal inference.
- For discrete variables $A$ and $B$, compute the $\chi^2$ statistics : large $\hat{\chi}^2$ value implies small correlation

## Missing Data

- Where missing data come from ?
  - Missing Completely At Random (MCAR) : the occurrence of missing data is a random event
  - Missing At Random (MAR) : depending on some control variables, e.g., the age $> 20$ is not acceptable in an investigation for teenager and thus is replaced by MAR
  - Missing Not At Random (MNAR) : missing data for bad performed employees after they are fired



## Simple Methods

- Deleting samples : for small size of samples with missing values
- Deleting variables : for series missing values in variables

| gradyear | gender | age | friends |
|---|---|---|---|
| 2006 | M | 18.98 | 7 |
| 2006 | F | 18.801 | 0 |
| 2006 | M | 18.335 | 69 |
| 2006 | F | 18.875 | 0 |
| 2006 | NA | 18.995 | 10 |
| 2006 | F | | 142 |
| 2006 | F | 18.93 | 72 |
| 2006 | M | 18.322 | 17 |
| 2006 | F | 19.055 | 52 |
| 2006 | F | 18.708 | 39 |
| 2006 | F | 18.543 | 8 |
| 2006 | F | 19.463 | 21 |
| 2006 | F | 18.097 | 87 |
| 2006 | NA | | 0 |
| 2006 | F | 18.398 | 0 |
| 2006 | NA | | 0 |
| 2006 | NA | | 135 |
| 2006 | F | 18.987 | 26 |
| 2006 | F | 17.158 | 27 |
| 2006 | F | 18.497 | 123 |
| 2006 | F | 18.738 | 35 |



## Filling Methods

- Filling with zero
- Filling with means for numerical type, and with modes for non-numerical type, applicable for MCAR ; drawback : concentrating in the mean and underestimating the variance ; solution : filling in different groups
- Filling with similar variables : auto-correlation is introduced
- Filling with past data
- Filling by K-Means : Compute the pairwise distances of the data using good variables (no missing values), then fill the missing values with the mean of the first K most similar good data, auto-correlation is introduced
- Filling with Expectation-Maximization (EM) : introduce hidden variables and use MLE to estimate the parameters (missing values)



## Filling Methods (Cont')

- Random filling :
  - Bayesian Bootstrap : for discrete data with range $\{x_i\}_{i=1}^k$, randomly sample $k-1$ numbers from $U(0,1)$ as $\{a_{(i)}\}_{i=0}^k$ with $a_{(0)} = 0$ and $a_{(k)} = 1$ ; then randomly sample from $\{x_i\}_{i=1}^k$ with probability distribution $\{a_{(i)} - a_{(i-1)}\}_{i=1}^k$ accordingly to fill in the missing values
  - Approximate Bayesian Bootstrap：Sample with replacement from $\{x_i\}_{i=1}^k$ to form new data set $X^* = \{x_i^*\}_{i=1}^{k^*}$ ; then randomly sample $n$ values from $X^*$ to fill in the missing values, allowing for repeatedly filling missing values
- Model based methods : treat missing variable as $y$, other variables as $\mathbf{x}$ ; take the data without missing values as our training set to train a classification or regression model ; take the data with missing values as our test set to predict the miss values



## Filling by Interpolation

- For the data of numeric type, each attribute (column vector) can be viewed as the function values $z_i = f(x_i)$ at the points $x_i$, where $x_i$ is a reference attribute (the reference attribute usually has no missing values, it can be chosen as the index)
- We can interpolate a function $f$ using the existing values $(x_i, z_i)$, and then fill in the missing values $z_k$ with $f(x_k)$
- Linear interpolation : treat $z = f(x)$ as linear function between the neighboring points $x_{k-1}$ and $x_{k+1}$ of $x_k$
- Lagrange interpolation : interpolate the $m+1$ existing values $\{(x_{i_j}, z_{i_j})\}_{j=1}^{m+1}$ by a degree $m$ polynomial $L_m(x)$

```
gen_data.interpolate()
```

| | feature1 | feature2 | feature3 |
|---|---|---|---|
| 1 | 1.728534 | -0.371519 | 1.451700 |
| 2 | 0.795975 | -1.067026 | -1.861944 |
| 3 | -0.030449 | -0.050409 | 1.299994 |
| 4 | -0.856872 | 0.966208 | 0.987861 |

Missing value

Missing value



## Special Values and Dummy Variables

- In Python, "np.nan" means missing values (Not a Number, missing float value)
- "None" is a Python object, used to represent missing values of the object type
- Dummy variables : e.g., missing values in gender ("Male" or "Female"), then define a third value "unknown" for the missing values

```
import pandas as pd
import numpy as np

teenager_sns = pd.read_csv('teenager_sns.csv')

print teenager_sns['gender'].value_counts()

teenager_sns['gender'] = teenager_sns['gender'].replace(np.NaN, 'unknown')

print ""
print "哑变量方法处理后: \n"
print teenager_sns['gender'].value_counts()
```

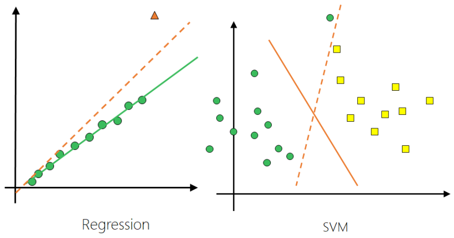
```
F    22054
M     5222
Name: gender, dtype: int64

哑变量方法处理后:

F          22054
M           5222
unknown     2724
Name: gender, dtype: int64
```

## Outliers

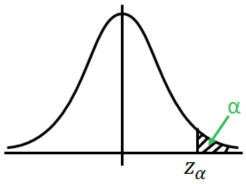
- Outliers : the data points seem to come from different distribution, or noisy data
- Outlier detection : unsupervised, e.g., Credit cheating detection, medical analysis, and information security, etc.

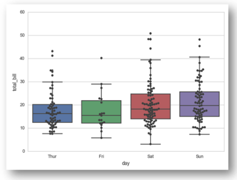





## Outliers Detection - Statistics Based Methods

- The samples outside the upper and lower $\alpha$-quantile for some small $\alpha$ (usually 1%)
- Observe from box plot
- $3\sigma$-rule in 1D : the sample $x$ with $x^*_{Z-score} > 3$ is an outlier

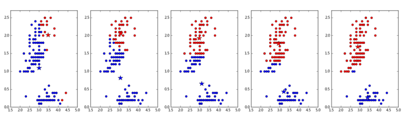


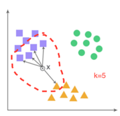





## Outliers Detection - Distance Based Methods

- $K$-means : run $K$-means clustering first, and then select the farthest $m$ points from their centers as outliers
- $K$NN : run $K$NN first, and then select the points that are far from their $K$ nearest neighbors (distance $> C$) as outliers



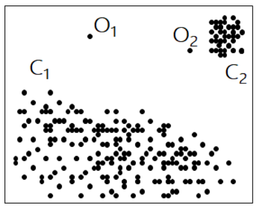




## Outliers Detection - Local Outlier Factor

Local Outlier Factor (LOF) is a density based method :

1. We could compute the density at each position $\mathbf{x}$, e.g., $p(\mathbf{x})$ (how to define the density if we only have data samples) ;
2. We could compare the density of each point $\mathbf{x}$ with the density of its neighbors, i.e., compare $p(\mathbf{x})$ with $p(\mathbf{x}_k)$ where $\mathbf{x}_k$ is close to $\mathbf{x}$ (in a neighborhood of $\mathbf{x}$, but how to define the neighborhood)






## Computing Density by Distance

Some definitions :

- $d(A, B)$ : distance between A and B ;
- $d_k(A)$ : $k$-distance of $A$, or the distance between $A$ and the $k$-th nearest point from $A$
- $N_k(A)$ : $k$-distance neighborhood of $A$, or the points within $d_k(A)$ from $A$ ;
- $rd_k(B, A)$ : $k$-reach-distance from A to B, the repulsive distance from $A$ to $B$ as if $A$ has a hard-core with radius $d_k(A)$, $rd_k(B, A) = \max\{d_k(A), d(A, B)\}$ ; note that $rd_k(A, B) \neq rd_k(B, A)$, which implies that $k$-reach-distance is not symmetric.

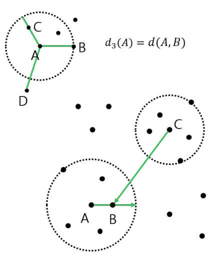


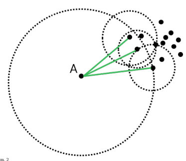

FIGURE: $rd_5(B, A) = d_5(A)$ and $rd_5(B, C) = d(B, C)$

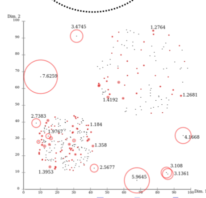


## Local Outlier Factor

Some definitions :

- $lrd_k(A)$ : local reachability density is inversely proportional to the average distancep, $lrd_k(A) = 1/\left(\frac{\sum_{O\in N_k(A)} rd_k(A,O)}{|N_k(A)|}\right)$ ; intuitively, if for most $O \in N_k(A)$, more than $k$ points are closer to $O$ than $A$ is, then the denominator is much larger than $d_k(A)$ and $lrd_k(A)$ is small ; e.g., $k = 3$ in the figure
- $LOF_k(A)$ : local outlier factor, $LOF_k(A) = \frac{\sum_{O\in N_k(A)} \frac{lrd_k(O)}{lrd_k(A)}}{|N_k(A)|}$ ;
- $LOF_k(A) \ll 1$, the density of $A$ is locally higher, dense point ; $LOF_k(A) \gg 1$, the density of $A$ is locally lower, probably outlier





# Further topics

- Other methods for outlier detection :
  - Isolation Forest : small path length (normally distributed) in a random forest (`sklearn.ensemble.Isolation`)
  - One-class support vector machine : classification as 1 (normal) vs. -1 (outlier) (`sklearn.svm.OneClassSVM`)
  - Robust covariance : based on Gaussian assumption, $3\sigma$ rule in high dimensions (`sklearn.covariance.EllipticEnvelope`)
- Outlier processing :
  - Delete outliers (treat them as missing values)
  - Robust regression : e.g., Theil-Sen regression, select the median of all possible slopes in two-point linear regression