



Data Normalization, Discretization and Reduction Techniques

Prepared By
-Anooja Joy

Tasks in data preprocessing

1. **Data cleaning:** fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies.
2. **Data integration:** using multiple databases, data cubes, or files.
3. **Data transformation:** normalization and aggregation.
4. **Data reduction:** reducing the volume but producing the same or similar analytical results.

Data cleaning

1. Fill in missing values (attribute or class value):

- **Ignore the tuple:** usually done when class label is missing.
- **Use the attribute mean** (or majority nominal value) to fill in the missing value.
- **Use the attribute mean** (or majority nominal value) for all samples belonging to the same class.
- **Predict the missing value by using a learning algorithm:** consider the attribute with the missing value as a dependent (class) variable and run a learning algorithm (usually Bayes or decision tree) to predict the missing value.

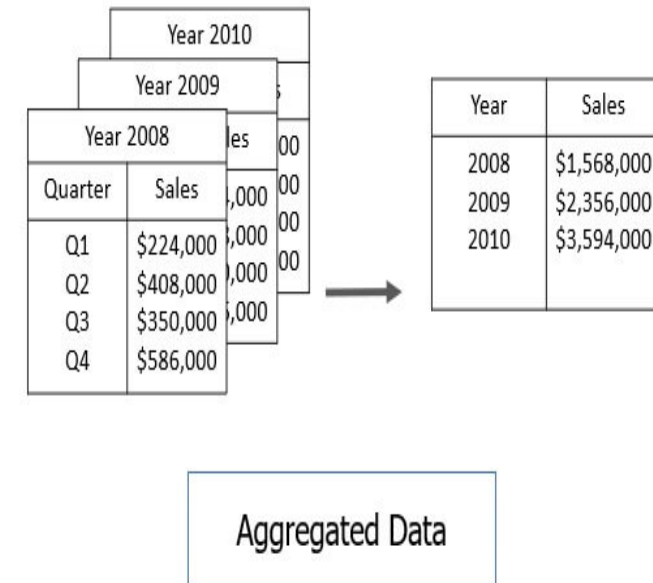
2. Identify outliers and smooth out noisy data:

- **Binning**
 - Sort the attribute values and partition them into bins
 - Then smooth by bin means, bin median, or bin boundaries.
- **Clustering:** group values in clusters and then detect and remove outliers (automatic or manual)
- **Regression:** smooth by fitting the data into regression functions.

3. Correct inconsistent data: use domain knowledge or expert decision.

DATA TRANSFORMATION

- Data must be transformed(or consolidated) to make patterns more understandable for mining.
- Data transformation is a function that maps the entire set of values of a given attribute to a new set of replacement values such that each old value can be identified with one of the new values
- The different data transformation Methods are:
 1. **Smoothing:** Remove noise from data using binning, regression, and clustering.
 2. **Attribute/feature construction:** New attributes constructed from the given ones
 3. **Aggregation:** Summarization, data cube construction. transforms a large set of data to a smaller volume by implementing aggregation operation on the data set.
 4. **Normalization:** Scaled to fall within a smaller, specified range Eg: min-max normalization, z-score normalization, normalization by decimal scaling
 5. **Discretization:** the raw values of a numeric attribute are replaced by interval label or conceptual label. Concept hierarchy climbing
 6. **Concept hierarchy generation for nominal data:** Here, attributes can be generalized to higher level concepts



Data Normalization

- Measurement unit used can affect the data analysis. Hence data are scaled to fall within a smaller range like 0.0 to 1.0. Such transformation or mapping the data to a smaller or common range will help all attributes to gain equal weight. This is known as Normalization.
- **Methods:** min-max normalization, z-score normalization, and normalization by decimal scaling.
- The normalization parameters such as mean, standard deviation, the maximum absolute value must be preserved in order to normalize the future data uniformly.



Age	Income
44	72000
27	48000
30	54000
38	61000
40	50000
35	58000
27	52000
48	79000
50	83000
37	67000



Age	Income
0.739	0.685
0	0
0.130	0.171
0.478	0.371
0.565	0.057
0.347	0.285
0	0.114
0.913	0.885
1	1
0.434	0.542

Min-Max Scaling

Why Normalization?

- If not normalized, one feature might completely dominate the others.
- make every datapoint have the same scale so each feature is equally important.
- It will help to speed up the learning phase while dealing with attributes on a different scale
- Avoid dependence on the choice of measurement units
- Comparison made easily
- The application of data mining algorithms becomes easier, effective and efficient
- Once the data is normalized, the extraction of data from databases becomes a lot faster.
- More specific data analyzing methods can be applied to normalized data.
- It prevent attributes with initially large ranges (e.g., income) from outweighing attributes with initially smaller ranges (e.g., binary attributes)

person_name	Salary	Year_of_experience	Expected Position Level
Aman	100000	10	2
Abhinav	78000	7	4
Ashutosh	32000	5	8
Dishi	55000	6	7
Abhishek	92000	8	3
Avantika	120000	15	1
Ayushi	65750	7	5

The attributes: year_of_experience on different scale hence attribute can take high priority over attribute year_of_experience the model.

Min-Max normalization

- Min-max normalization performs a linear transformation on the original data in range $[0, 1]$ or $[-1, 1]$. Selecting the target range depends on the nature of data.
- If *minA* and *maxA* are the **minimum** and **maximum** values of an **attribute A**, Min-max normalization maps a value, *vi* of A to *vi'* in the range [*new-minA*, *new-maxA*] by computing:

$$v' = \frac{v - \min_A}{\max_A - \min_A} (\text{new_max}_A - \text{new_min}_A) + \text{new_min}_A$$

- Min-max normalization preserves the relationships among the original data values.
- It encounter an “out-of-bounds” error if a future input case for normalization falls outside of the original data range for A.

FEATURES

- It does not center the mean at 0.
- It makes the variance vary across variables.
- It may not maintain the shape of the original distribution.
- The minimum and maximum values are in the range of $[0, 1]$.
- This method is very sensitive to outliers.

Min-Max normalization Example

1. Let income range be from \$12,000 to \$98,000. Map income to the range [0.0, 1.0](ie normalized to). By min-max normalization, a value of \$73,600 for income is transformed to

$$\frac{73,600 - 12,000}{98,000 - 12,000} (1.0 - 0) + 0 = 0.716$$

2. Given Data

Employee Name	Years of Experience
A	8
B	20
C	10
D	15

z-score normalization/ Feature Standardisation

- This method normalizes the value for attribute A using the **mean** and **standard deviation**. The formula for the same is:

$$v'_i = \frac{v_i - \bar{A}}{\sigma_A}$$

- \bar{A} and σ_A are the **mean** and **standard deviation** for attribute A respectively.
- Data can include **multiple dimensions**.
- **Feature standardization** makes the values of each feature in the data have **zero-mean** (when subtracting the mean in the numerator) and **unit-variance** but **not normal distribution it can be still skewed**.
- This method is widely used for normalization in many machine learning algorithms (e.g., support vector machines, logistic regression, and arti

FEATURES

- It scales the variance at 1.
- It centers the mean at 0.
- It preserves the shape of the original distribution.
- It preserves outliers if they exist.
- Minimum and maximum values vary.

Age	Income
44	72000
27	48000
30	54000
38	61000
40	50000
35	58000
27	52000
48	79000
50	83000
37	67000



Age	Income
0.827	0.818
-1.370	-1.228
-0.982	-0.716
0.051	-0.119
0.310	-1.057
-0.336	-0.375
-1.370	-0.887
1.344	1.415
1.602	1.757
-0.077	0.392

Standardization

z-score normalization Example

1. Given mean and standard deviation for attribute A as \$54,000 and \$16,000 respectively. Normalize the value \$73,600 using z-score normalization.

$$\frac{73600 - 54000}{16000} = 1.225$$

COMPARISON

Min-max normalization	Z-score normalization
Not very well efficient in handling the outliers	Handles the outliers in a good way.
Min-max Guarantees that all the features will have the exact same scale.	Helpful in the normalization of the data but not with the exact same scale.

Decimal scaling

- This method normalizes the value of attribute A by moving the decimal point in the value. This movement of a decimal point depends on the **maximum absolute value of A**.

The formula for the decimal scaling is:

$$v'_i = \frac{v_i}{10^j}$$

- J is the The smallest integer j such that $\text{Max}(|v_i/10^j|) < 1$
- Number of digits in data value with largest absolute value.

Decimal scaling Example

1. The observed values for attribute A lie in the range from -986 to 917 and the maximum absolute value for attribute A is 986. Here, to normalize each value of attribute A using decimal scaling, we have to divide each value of attribute A by 1000 i.e. $j=3$. So, the value -986 would get normalized to -0.986 and 917 would get normalized to 0.917.
2. *The Data is: -10, 201, 301, -401, 501, 601, 701. Normalize the given data*
3. *Glven Data:*

Employee Name	Salary
A	10,000
B	25,000
C	8,000
D	15,000

EXERCISE

- Use these methods to normalize the following group of data: 200, 300, 400, 600, 1000
 - (a) min-max normalization by setting min $D = 0$ and max $D = 1$
 - (b) z-score normalization
 - (c) z-score normalization using the mean absolute deviation instead of standard deviation
 - (d) normalization by decimal scaling

Data discretization

- **Data discretization** is defined as a process of **converting continuous data attribute values** into a **finite set of intervals with minimal loss of information** and associating with each interval some specific data value. (e.g., 0–10, 11–20, etc.) or conceptual labels (e.g., youth, adult, senior)
- The goal of discretization is to reduce the number of values a continuous variable assumes by grouping them into a number, **b, of intervals or bins**.
- Interval labels can then be used to replace actual data values. **Discretization** reduce data size.
- **Why Discretization?**
 - Improves the **quality of discovered knowledge**
 - Easy maintainability of the data
 - There is a necessity to use discretized data by **many DM algorithms which can only deal with discrete attributes**.
 - **Reduces the running time** of various data mining tasks such as association rule discovery, classification, and prediction
 - Prepare for further analysis, e.g., classification
 - Discretization is considered a **data reduction mechanism** because it **diminishes data from a large domain of numeric values to a subset of categorical values**.
- **Nature of good discretization:**
 - minimize information loss.
- **how to select the number of intervals or bins**
- **how to decide on their width**

Discretization Process

- Assuming a **dataset S** consisting of N examples, M attributes, and c class labels, a discretization scheme.
- DA would exist on the continuous attribute $A \in M$, which partitions this attribute into k discrete and disjoint intervals, where d_0 and d_{kA} are, respectively, the minimum and maximal value, and represents the set of cut points of A in ascending order. $\{|d_0 d_1|, |d_1 d_2|, |d_3 d_4|, \dots, |d_{ka-1} d_{kA}|\}$

STEPS OF DISCRETIZATION

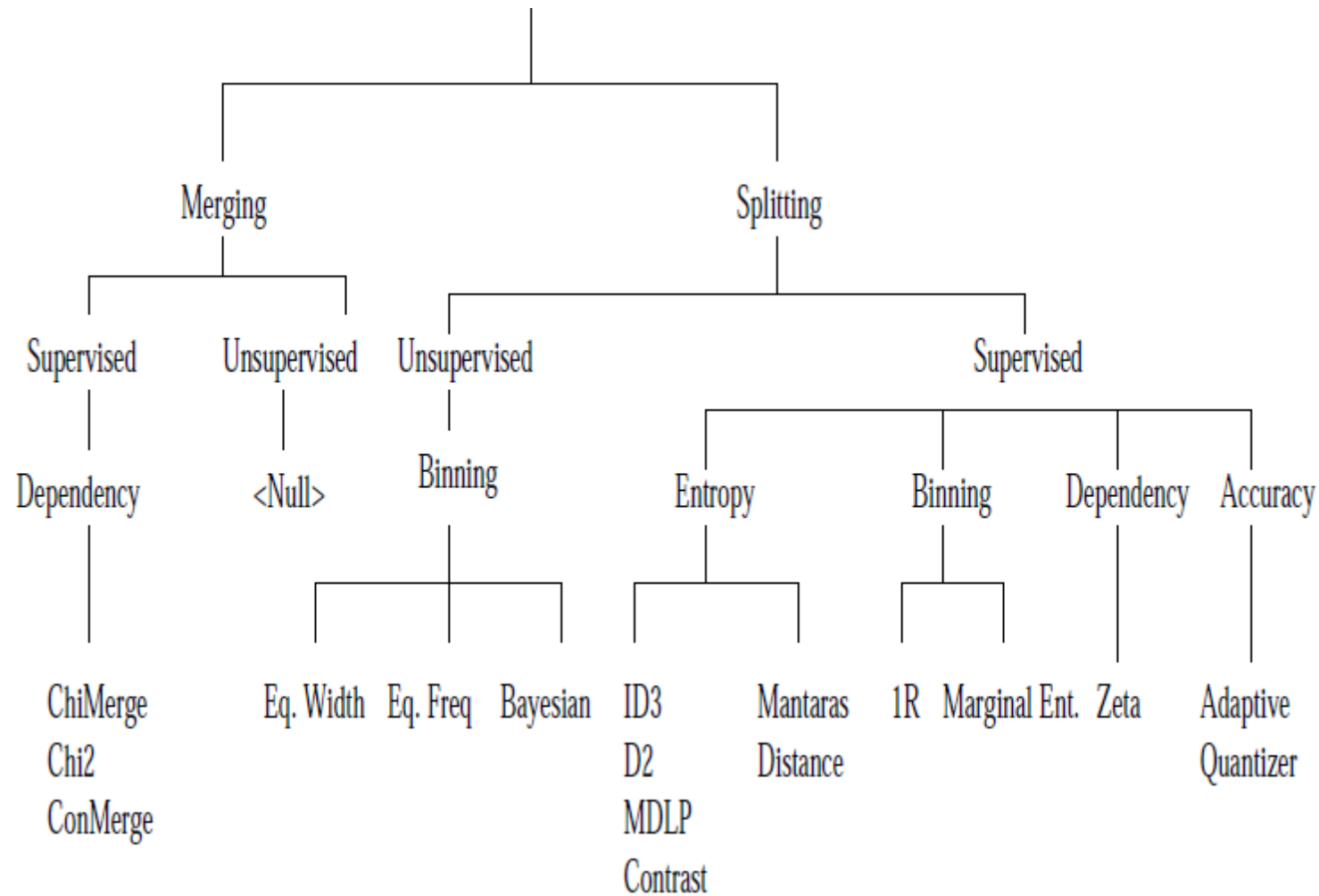
- Sorting the continuous values of the feature to be discretized,
- Evaluating a cut point for splitting or adjacent intervals for merging,
- Splitting or merging intervals of continuous values according to some defined criterion.
- Stopping at some point.

Table: Before discretization

Age	10,11,13,14,17,19,30, 31, 32, 38, 40, 42,70 , 72, 73, 75
-----	--

Table: After discretization

Attribute	Age	Age	Age
	10,11,13,14,17, 19,	30, 31, 32, 38, 40, 42	70 , 72, 73, 75
After Discretization	Young	Mature	Old



DATA DISCRETIZATION TYPES

Data discretization Types

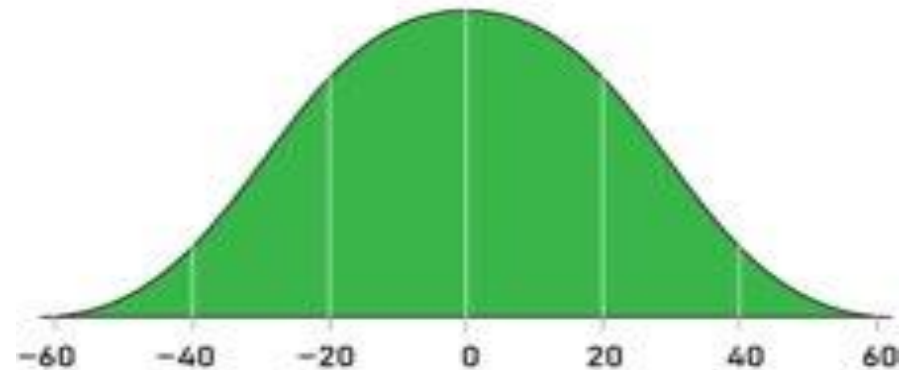
- Discretization can be performed with or without taking class information, if available, into account. These are the supervised and unsupervised ways
- **Unsupervised discretization methods** are **not provided with class label informatio**. Eg: **Equal-width discretization, Equal-frequency discretization, K-means discretization** **supervised discretization methods** are **supplied with a class label for each data item value**. Eg: **Entropy based, Decision Trees**
- Both unsupervised and supervised discretization methods can be further subdivided into **top-down(split)** and **bottom-up(merge)** methods.
- A **top-down method** starts with a single interval that includes all data attribute values and then generates a set of intervals by splitting the initial interval into two or more intervals. Splitting Algorithm consists of 4 steps. 1) Sort the feature values. 2) Search for a suitable cut point 3) split the range of continuous values according to cut point 4) Stop when criteria is satisfied
- A **bottom-up method** initially considers each data point as a separate interval. It then selects one or more adjacent data points merging them into a new interval. If the process starts by considering all of the continuous values as potential split-points, removes some by merging neighborhood values to form intervals, then it is called bottom-up discretization or merging.
- 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 collecting and replacing low-level concepts with higher-level concepts. Data mining on a reduced data set means fewer input/output operations and is more efficient than mining on a larger data set.
- discretization techniques and concept hierarchies are typically applied before data mining, rather than during mining.

Data discretization Types

Equal-width Discretization

- The most simple form of discretization that divides the range of possible values into **N** bins of the **same width**.
 - The width of intervals is determined by the following formula:
 - *where **N** is the number of bins or intervals, this parameter is something to determine experimentally—there's no rule of thumb here.*
- Example:** if the variable interval is [100, 200], and we want to create 5 bins, that means $200 - 100 / 5 = 20$, so each bin's width is 20, and the intervals will be [100, 120], [120, 140], ..., [180, 200]
- Equal-width discretization does not improve the values spread.
 - This method handles outliers.

$$width = \frac{maxvalue - minvalue}{N}$$

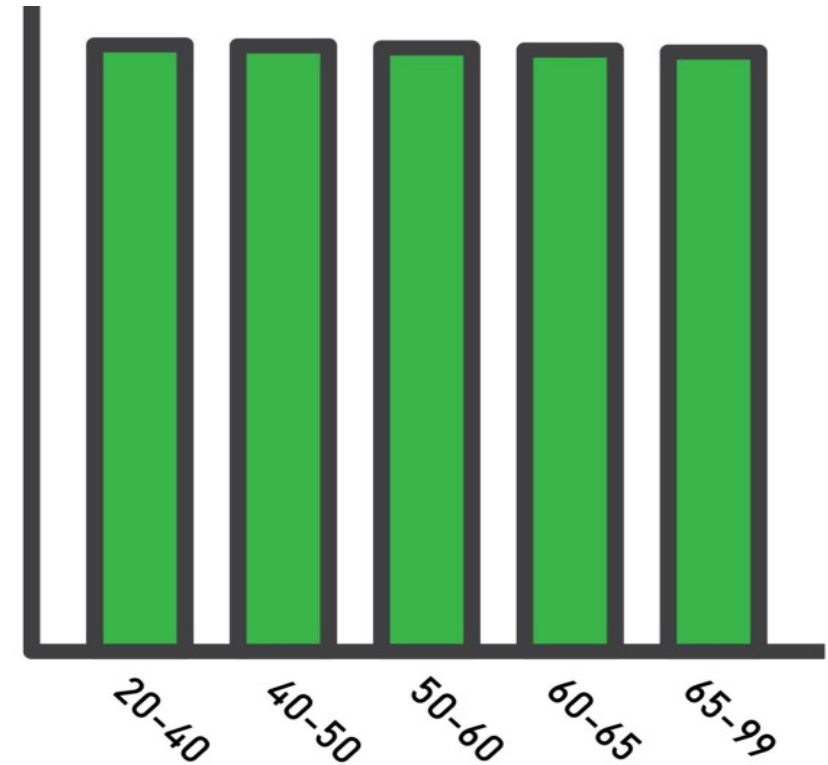


Equal width Discretization

Data discretization Types

Equal-Frequency Discretization

- Equal-frequency discretization divides the scope of possible values of the variable into **N** bins, where each bin holds the **same number** (or approximately the same number) of observations..
- the interval boundaries correspond to the quantiles.
- This method improves the value spread.
- Equal-frequency handles outliers.



Equal-Frequency Discretization

Data Discretization Typical Methods

- All the methods can be applied recursively.

1. Binning

- Binning is a **top-down splitting technique** based on a specified number of bins. Binning is an **unsupervised discretization technique**. Main challenge in discretization is to choose the number of intervals or bins and how to decide on their width.

2. Histogram Analysis

- It is a **Top-down split**. Since histogram analysis does not use class information so it is an **unsupervised discretization technique**. Histograms partition the values for an attribute into disjoint ranges called buckets.

3. Cluster Analysis

- Cluster analysis is a **unsupervised method** which can be **top-down split or bottom-up merge**. A clustering algorithm can be applied to discrete a numerical attribute of A by partitioning the values of A into clusters or groups. Each initial cluster or partition may be further decomposed into several subcultures, forming a lower level of the hierarchy. Detect and remove outliers

4. Decision-tree analysis (supervised, top-down split)

5. Correlation (e.g. chi merge) analysis (unsupervised, bottom-up merge)

1.Simple Discretization: Binning

- Binning methods smooth a sorted data value by consulting its “neighborhood,” that is, the values around it. The sorted values are distributed into a number of “buckets,” or bins. Because binning methods consult the neighborhood of values, they perform local smoothing
- Attribute values can be discretized by applying **equal-width** or **equal-frequency binning**, and then replacing each bin value by the bin mean or median, as in smoothing by bin means or smoothing by bin medians, respectively.
- Binning does not use class information and is therefore an unsupervised discretization technique. It is sensitive to the user-specified number of bins, as well as the presence of outliers.

Binning Methods for Data Smoothing

- Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
1. Partition into **equal-frequency (equi-depth)** bins: of size 4
 - Bin 1: 4, 8, 9, 15
 - Bin 2: 21, 21, 24, 25
 - Bin 3: 26, 28, 29, 34
 2. Smoothing by **bin means**: each value in a bin is replaced by the mean value of the bin.
 - Bin 1: 9, 9, 9, 9
 - Bin 2: 23, 23, 23, 23
 - Bin 3: 29, 29, 29, 29
 3. Smoothing by **bin boundaries**: The minimum and maximum values in a given bin are identified as the bin boundaries. Each bin value is then replaced by the closest boundary value. In general, the larger the width, the greater the effect of the smoothing.
 - Bin 1: 4, 4, 4, 15
 - Bin 2: 21, 21, 25, 25
 - Bin 3: 26, 26, 26, 34
 4. Smoothing by **bin medians** : each bin value is replaced by the bin median.

Entropy Based Binning

- Given Data. Discretize the temperature variable using entropy-based binning algorithm

O-Ring Failure	Temperature
Y	53
Y	56
Y	57
N	63
N	66
N	67
N	67
N	67
N	68
N	69
N	70
Y	70
Y	70
Y	70
N	72
N	73
N	75
Y	75
N	76
N	76
N	78
N	79
N	80
N	81

ENTROPY BASED BINNING

- Calculate "Entropy" for the target.

O-Ring Failure	
Y	N
7	17

- $E(\text{Failure}) = E(7, 17) = E(0.29, .71) = -0.29 \times \log_2(0.29) - 0.71 \times \log_2(0.71)$
 $= \mathbf{0.871}$

- Calculate "Entropy" for the target given a bin

O-Ring Failure			
Y	N		
Temperature	≤ 60	3	0
	> 60	4	17

- $E(\text{Failure, Temperature}) = P(\leq 60) \times E(3, 0) + P(> 60) \times E(4, 17) = 3/24 \times 0 + 21/24 \times 0.7 = \mathbf{0.615}$

- Calculate "Information Gain" given a bin.

- Information Gain (Failure, Temperature) $= 0.871 - 0.615 = \mathbf{0.256}$

FINAL

O-Ring Failure	Temperature
Y	53
Y	56
Y	57
N	63
N	66
N	67
N	67
N	67
N	68
N	69
N	70
Y	70
Y	70
Y	70
N	72
N	73
N	75
Y	75
N	76
N	76
N	78
N	79
N	80
N	81

Gain = 0.256

Temperature ≤ 60
 > 60

O-Ring Failure	
Y	N
3	0
4	17

Gain = 0.101

Temperature ≤ 70
 > 70

O-Ring Failure	
Y	N
6	8
1	9

Gain = 0.148

Temperature ≤ 75
 > 75

O-Ring Failure	
Y	N
7	11
0	6

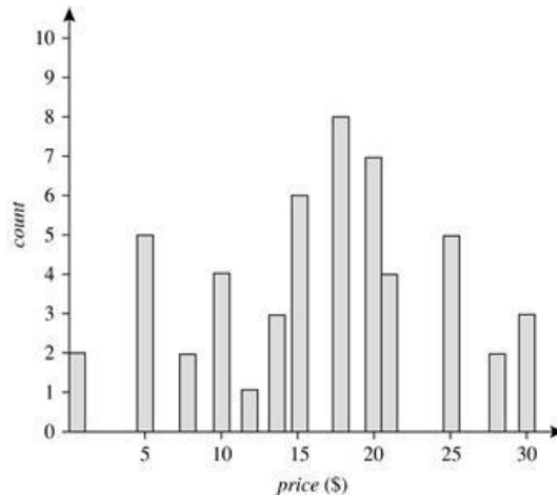
The information gains for all three bins show that the best interval for "Temperature" is (≤ 60 , > 60) because it returns the highest gain.

2. Histogram

- Histogram analysis is an unsupervised discretization technique because it does not use class information.
- A histogram partitions the values of an attribute, A , into disjoint ranges called buckets or bins.
- If each bucket represents only a single attribute–value/frequency pair, the buckets are called singleton buckets. Singleton buckets are useful for storing high-frequency outliers.
- Histograms are effective at approximating sparse data, dense data, as well as highly skewed and uniform data.
- The histograms described before for single attributes can be extended for multiple attributes. Multidimensional histograms can capture dependencies between attributes. These histograms have been found effective in approximating data with up to five attributes.

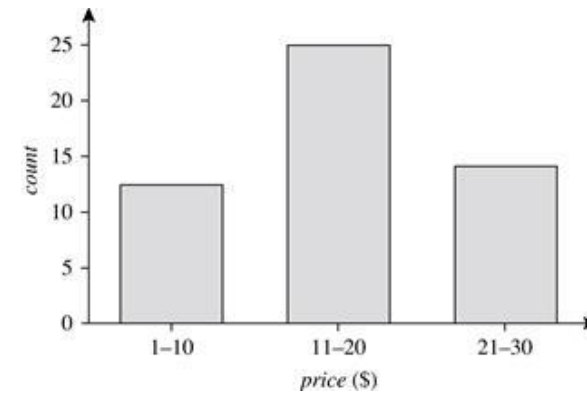
Histograms Example

- The following data are a list of AllElectronics prices for commonly sold items (rounded to the nearest dollar). The numbers have been sorted: 1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30.



A histogram for price using **singleton buckets**—each bucket represents one price–value/frequency pair.

FIGURE



An equal-width histogram for price, where values are aggregated so that each bucket has a **uniform width of \$10**.

Histogram Types

- Various partitioning rules can be used to define histograms
- **Equal-width(or distance):** In an equal-width histogram, the width of each bucket range is uniform.
 - Divides the range into **N intervals of equal size**: uniform grid
 - if A and B are the lowest and highest values of the attribute, the width of intervals will be: **$W = (B - A)/N$** . And the interval boundaries are: **$A + w, A + 2w, \dots, A + (k-1)w$**
 - The most straightforward, but outliers may dominate presentation
 - Skewed data is not handled well
 - Advantages:
 - natural order of the attribute-values is preserved.
 - Light on storage requirements
 - Disadvantages:
 - High variance
 - Difficult to estimate errors
- **Equal-frequency (or equal-depth):** In an equal-frequency histogram, the buckets are created so that, roughly, the frequency of each bucket is constant (i.e., each bucket contains roughly the same number of contiguous data samples).
 - Divides the range into N intervals, each containing approximately same number of samples
 - Good data scaling
 - Managing categorical attributes can be tricky
 - Disadvantages:
 - Variance within a bucket may be still very high
 - Storage requirement same as equi-width, but more complex to maintain

Example

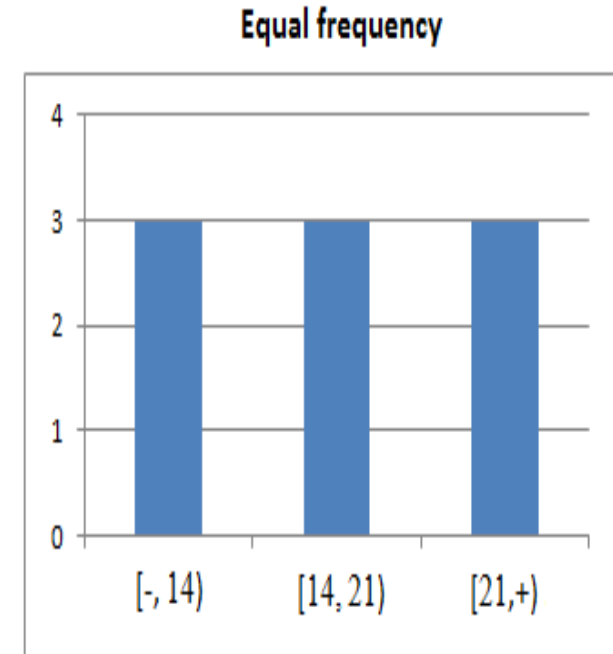
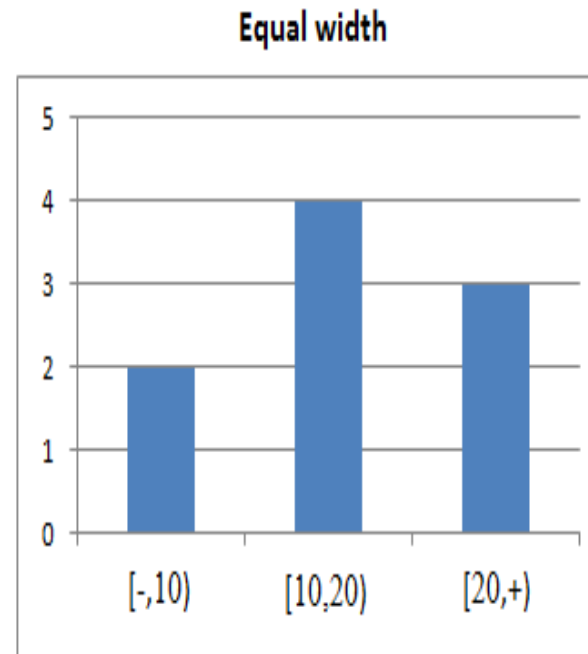
- **Data** : 0, 4, 12, 16, 16, 18, 24, 26, 28

- **Equal width**

- Bin 1: 0, 4 [-,10)
- Bin 2: 12, 16, 16, 18 [10,20)
- Bin 3: 24, 26, 28 [20,+)

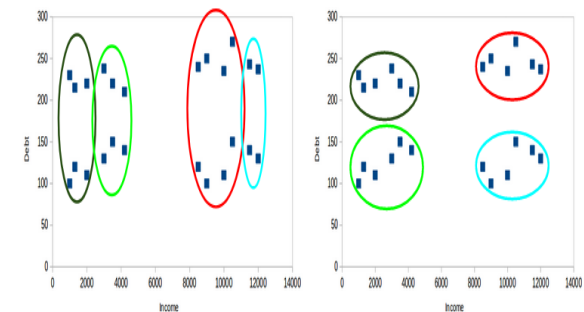
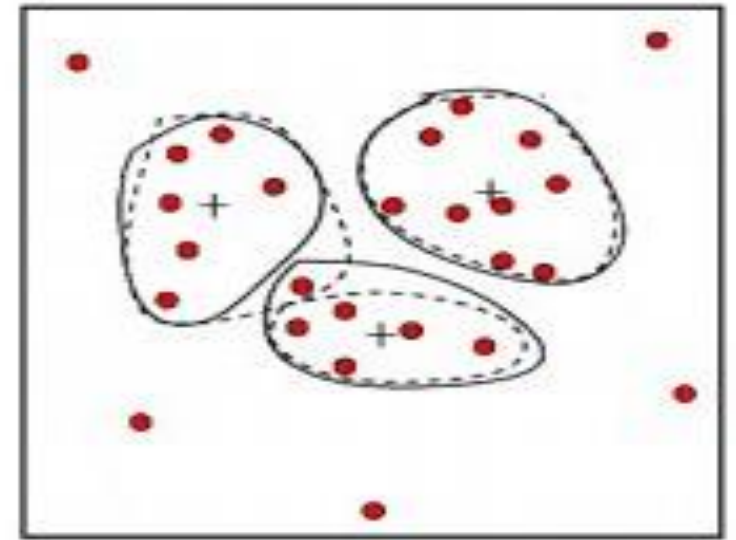
- **Equal frequency**

- Bin 1: 0, 4, 12 [-, 14)
- Bin 2: 16, 16, 18 [14, 21)
- Bin 3: 24, 26, 28 [21,+)



3. Discretization using data clustering techniques

- Cluster analysis is a popular data discretization method.
- A clustering algorithm can be applied to discretize a numeric attribute, A , by partitioning the values of A into clusters or groups based on similarity, and store cluster representation (e.g., centroid and diameter) only
- Partition data set into clusters
- There are many choices of clustering definitions and clustering algorithms. Eg: K-Means and K-Medoid algorithm
- properties of clusters
 - **All the data points in a cluster should be similar to each other.**
 - **The data points from different clusters should be as different as possible**

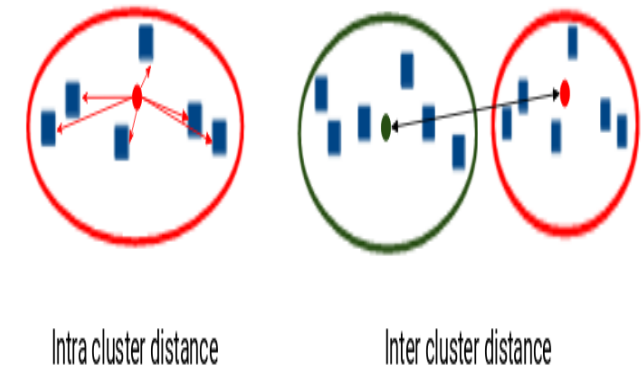


Case - I

Case - II

Different Evaluation Metrics for Clustering

- **Property 1-inertia** calculates the sum of distances of all the points within a cluster from the centroid of that cluster. This distance within the clusters is known as **intracluster distance**. So, inertia gives us the sum of intracluster distances. *the lesser the inertia value, the better our clusters are.*
- **Property 2-Dunn index.** The distance between the centroids of two different clusters is known as **inter-cluster distance**. *Dunn index is the ratio of the minimum of inter-cluster distances and maximum of intracluster distances.* maximize the Dunn index. The more the value of the Dunn index, the better will be the clusters



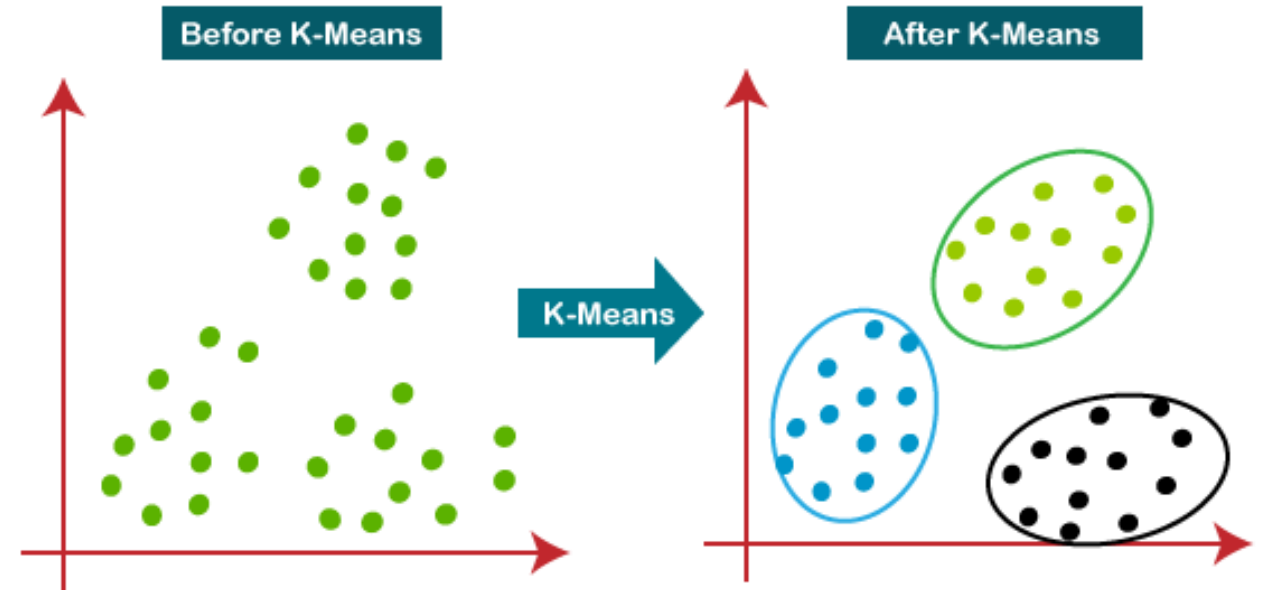
$$\text{Dunn Index} = \frac{\min(\text{Inter cluster distance})}{\max(\text{Intra cluster distance})}$$

Clusters are far apart

$$\text{Dunn Index} = \frac{\min(\text{Inter cluster distance})}{\max(\text{Intra cluster distance})}$$

Clusters are compact

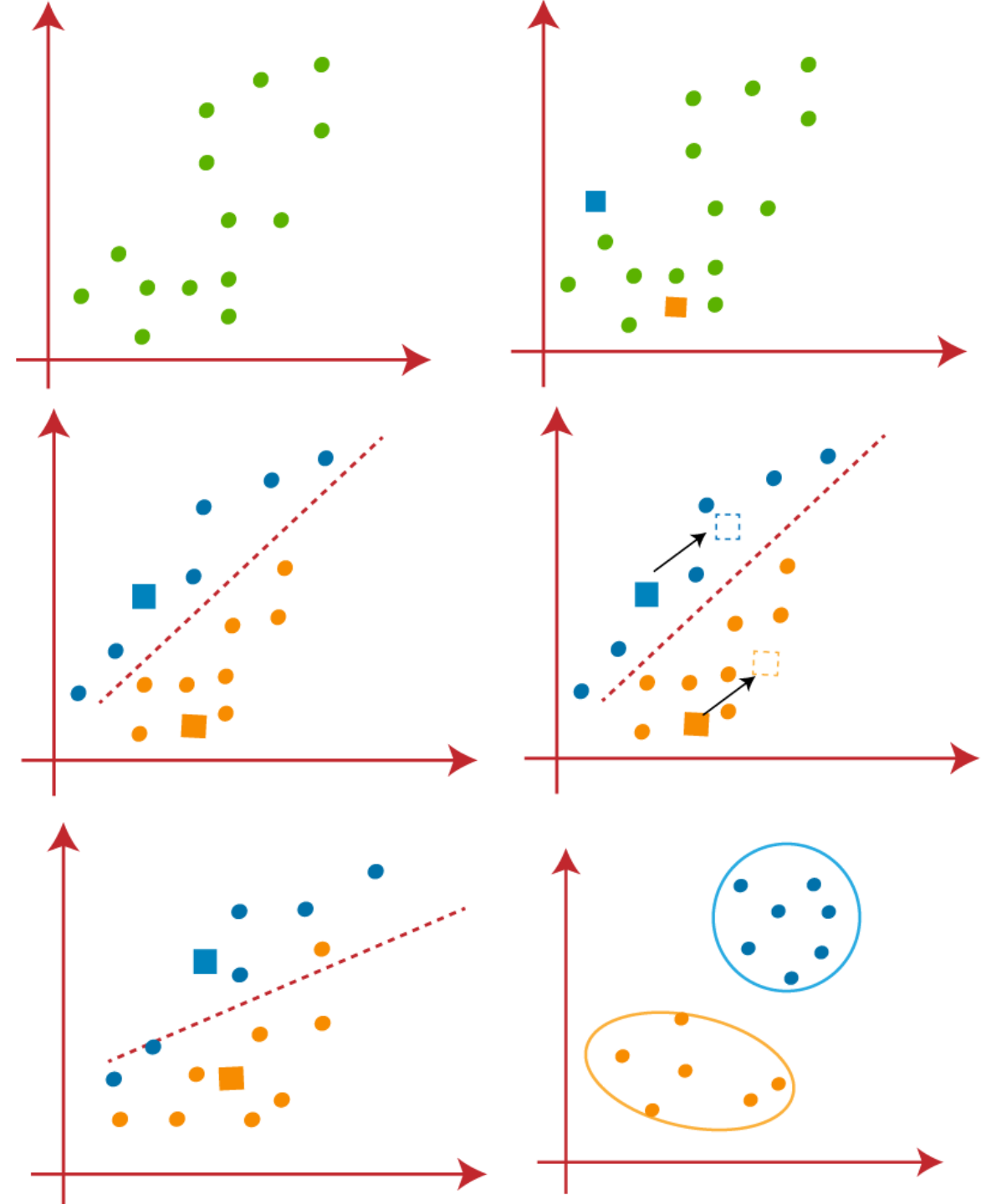
3.1 K-Means Clustering Algorithm



- A **cluster** refers to a collection of data points aggregated together because of certain similarities.
- K-mean algorithm is one of the **centroid based technique**. it is also referred to as **Lloyd's algorithm**.
- K means algorithm takes the **unlabeled dataset as input**, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters without the need for any training.
- K refers to the number of pre-defined clusters that need to be created in the process, as if $K=2$, there will be two clusters, and for $K=3$, there will be three clusters.
- Each cluster is associated with a centroid and algorithm aims to minimize the sum of distances between the data point and their corresponding cluster centroids.

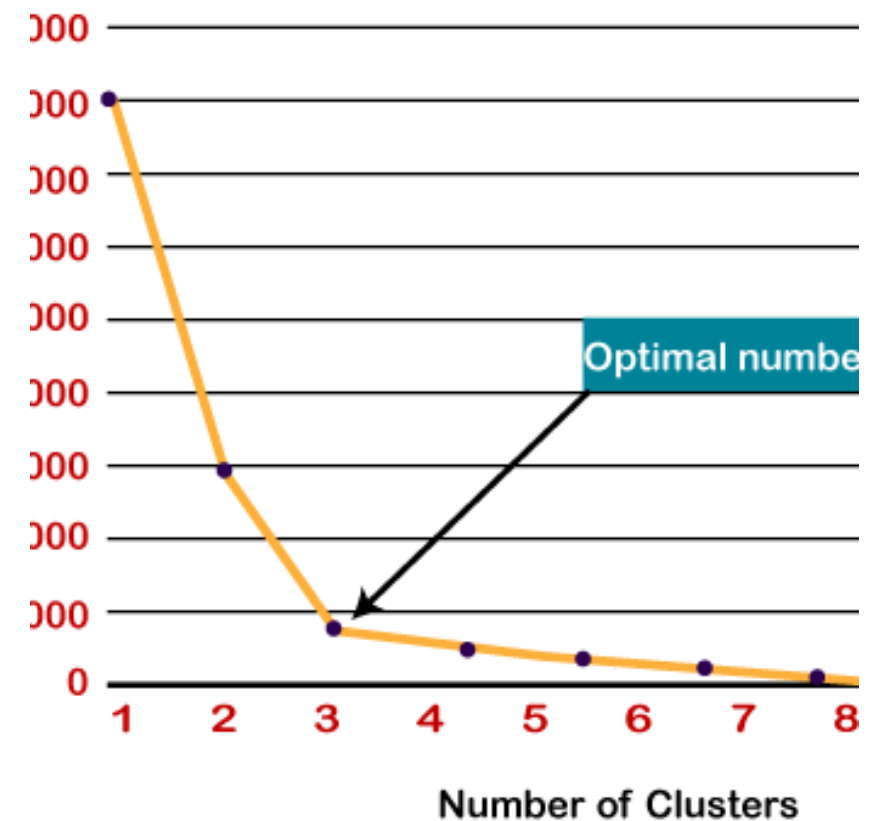
K-Means Algorithm Steps

1. Select the number K to decide the number of clusters. K can be determined using some techniques like Elbow Method using WCSS (Within Cluster Sum of Squares), Silhouette Method.
2. Compute random seed points as the centroids or K points of the clusters of the current partitioning (A centroid is the imaginary or real location representing the center of the cluster or mean point of the cluster.)
3. Assign each object to the cluster with the nearest seed point based on euclidean distance. Using a different distance function other than (squared) Euclidean distance may prevent the algorithm from converging. Partition objects into k nonempty subsets by seeing the closest centroids for each data points.
4. Calculate the variance and place a new centroid of each cluster based on mean value of the cluster. Better choice is to place them as much as possible far away from each other
5. Go back to Step 3, stop when the assignment does



choose the value of "K number of clusters" in K-means Clustering

- $WCSS = \sum_{P_i \text{ in Cluster1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster3}} \text{distance}(P_i, C_3)^2$
- $\sum_{P_i \text{ in Cluster1}} \text{distance}(P_i, C_1)^2$: It is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other two terms.
- To find the optimal value of clusters, the elbow method follows the below steps:
- It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).
- For each value of K, calculates the WCSS value.
- Plots a curve between calculated WCSS values and the number of clusters K.
- The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.



Stopping Criteria for K-Means Clustering

- There are essentially three stopping criteria that can be adopted to stop the K-means algorithm:
 1. Centroids of newly formed clusters do not change
 2. Points remain in the same cluster
 3. Maximum number of iterations are reached

EXAMPLE

- Apply k-means clustering algorithm and euclidean distance to cluster the following 1 dimensional data set {2, 4, 10, 12, 3, 20, 30, 11, 25} into 2 clusters with initial means as $M_1 = 4$, $M_2 = 11$

EXAMPLE

- Apply k-means clustering algorithm and euclidean distance to cluster the following 1 dimensional data set {2, 4, 10, 12, 3, 20, 30, 11, 25} into 2 clusters

Iteration 1

- M1, M2 are the two randomly selected centroids/means where M1= 4, M2=11
- Calculate the Euclidean distance as $D=[x,a]=\sqrt{(x-a)^2}$
- D1 is the distance from M1 and D2 is the distance from M2
- initial clusters are C1= {2, 4, 3} C2= {10, 12, 20, 30, 11, 25}

Datapoint	D1	D2	Cluster
2	2	9	C1
4	0	7	C1
10	6	1	C2
12	8	1	C2
3	1	8	C1
20	16	9	C2
30	26	19	C2
11	7	0	C2
25	21	14	C2

EXAMPLE

Iteration 2

- Calculate new mean of datapoints in C1 and C2. $M1 = (2+3+4)/3 = 3$ and $M2 = (10+12+20+30+11+25)/6 = 18$. New Clusters $C1 = \{2, 3, 4, 10\}$ and $C2 = \{12, 20, 30, 11, 25\}$

Iteration 3

- Calculate new mean of datapoints in C1 and C2.
- $M1 = (2+3+4+10)/4 = 4.75$ and $M2 = (12+20+30+11+25)/5 = 19.6$
- New Clusters $C1 = \{2, 3, 4, 10, 12, 11\}$ and $C2 = \{20, 30, 25\}$

Iteration 4

- Calculate new mean of datapoints in C1 and C2.
- $M1 = (2+3+4+10+12+11)/6 = 7$ and $M2 = (20+30+25)/3 = 25$
- New Clusters $C1 = \{2, 3, 4, 10, 12, 11\}$ and $C2 = \{20, 30, 25\}$

Datapoint	D1	D2	Cluster
2	1	16	C1
4	1	14	C1
3	0	15	C1
10	7	8	C1
12	9	6	C2
20	17	2	C2
30	27	12	C2
11	8	7	C2
25	22	7	C2

Datapoint	D1	D2	Cluster
2	2.75	17.6	C1
4	0.75	15.6	C1
3	1.75	16.6	C1
10	5.25	9.6	C1
12	7.25	7.6	C1
20	15.25	0.4	C2
30	25.25	10.4	C2
11	6.25	8.6	C1
25	20.25	5.4	C2

Datapoint	D1	D2	Cluster
2	5	23	C1
4	3	21	C1
3	4	22	C1
10	3	15	C1
12	5	13	C1
11	4	14	C1
20	13	5	C2
30	23	5	C2
25	18	0	C2

CHALLENGE

- Use the k-means algorithm and Euclidean distance to cluster the following 8 examples into 3 clusters: $A1=(2,10)$, $A2=(2,5)$, $A3=(8,4)$, $A4=(5,8)$, $A5=(7,5)$, $A6=(6,4)$, $A7=(1,2)$, $A8=(4,9)$. Run the k-means algorithm for 3 epoch/iteration. Let $seed1=A1=(2,10)$, $seed2=A4=(5,8)$, $seed3=A7=(1,2)$

SOLUTION

- Let seed1=A1=(2,10), seed2=A4=(5,8), seed3=A7=(1,2)

Data Point	Value	D1(2,10)	D2(5,8)	D3(1,2)	Cluster
A1	(2,10)	0	$\sqrt{13}$	$\sqrt{65}$	C1
A2	(2,5)	$\sqrt{25}$	$\sqrt{18}$	$\sqrt{10}$	C3
A3	(8,4)	$\sqrt{36}$	$\sqrt{25}$	$\sqrt{53}$	C2
A4	(5,8)	$\sqrt{13}$	0	$\sqrt{52}$	C2
A5	(7,5)	$\sqrt{50}$	$\sqrt{13}$	$\sqrt{45}$	C2
A6	(6,4)	$\sqrt{52}$	$\sqrt{17}$	$\sqrt{29}$	C2
A7	(1,2)	$\sqrt{65}$	$\sqrt{52}$	0	C3
A8	(4,9)	$\sqrt{5}$	$\sqrt{2}$	$\sqrt{58}$	C2

- new clusters: 1: {A1}, 2: {A3, A4, A5, A6, A8}, 3: {A2, A7}
- Centroid 1=(2,10) Centroid 2 = $((8+5+7+6+4)/5, (4+8+5+4+9)/5) = (6, 6)$
Centroid 3 = $((2+1)/2, (5+2)/2) = (1.5, 3.5)$

SOLUTION

- Iteration 2

Data Point	Value	D1 (2,10)	D2(6, 6)	D3(1.5, 3.5)	Cluster
A1	(2,10)	0	$\sqrt{32}$	$\sqrt{65}$	C1
A2	(2,5)	$\sqrt{25}$	$\sqrt{18}$	$\sqrt{10}$	C3
A3	(8,4)	$\sqrt{36}$	$\sqrt{25}$	$\sqrt{53}$	C2
A4	(5,8)	$\sqrt{13}$	0	$\sqrt{52}$	C2
A5	(7,5)	$\sqrt{50}$	$\sqrt{13}$	$\sqrt{45}$	C2
A6	(6,4)	$\sqrt{52}$	$\sqrt{17}$	$\sqrt{29}$	C2
A7	(1,2)	$\sqrt{65}$	$\sqrt{52}$	0	C3
A8	(4,9)	$\sqrt{5}$	$\sqrt{2}$	$\sqrt{58}$	C2

- New Clusters : 1: {A1, A8}, 2: {A3, A4, A5, A6}, 3: {A2, A7}
- centroids $C1=(3, 9.5)$, $C2=(6.5, 5.25)$ and $C3=(1.5, 3.5)$. After the 3rd epoch, the results would be: 1: {A1, A4, A8}, 2: {A3, A5, A6}, 3: {A2, A7} with centers $C1=(3.66, 9)$, $C2=(7, 4.33)$ and $C3=(1.5, 3.5)$.

SOLUTION

- Iteration 3

Data Point	Value	D1	D2	D3	Cluster
A1	(2,10)	0	$\sqrt{13}$	$\sqrt{65}$	C1
A2	(2,5)	$\sqrt{25}$	$\sqrt{18}$	$\sqrt{10}$	C3
A3	(8,4)	$\sqrt{36}$	$\sqrt{25}$	$\sqrt{53}$	C2
A4	(5,8)	$\sqrt{13}$	0	$\sqrt{52}$	C2
A5	(7,5)	$\sqrt{50}$	$\sqrt{13}$	$\sqrt{45}$	C2
A6	(6,4)	$\sqrt{52}$	$\sqrt{17}$	$\sqrt{29}$	C2
A7	(1,2)	$\sqrt{65}$	$\sqrt{52}$	0	C3
A8	(4,9)	$\sqrt{5}$	$\sqrt{2}$	$\sqrt{58}$	C2

- New Clusters : 1 1: {A1, A4, A8}, 2: {A3, A5, A6}, 3: {A2, A7}
- centroids C1=(3.66, 9), C2=(7, 4.33) and C3=(1.5, 3.5).

Advantages and Disadvantages of K-Means Clustering

ADVANTAGES

- It is very simple to implement.
- It is scalable to a huge data set and also faster to large datasets.
- it adapts the new examples very frequently.
- Generalization of clusters for different shapes and sizes.

DISADVANTAGES

- Slow for large number of samples: As this algorithm access each point of the dataset, it becomes slow when the sample size grows.
- The k-means algorithm is sensitive to outliers ! Since an object with an extremely large value may substantially distort the distribution of the data
- Value of K need to be specified beforehand. Choosing the k values manually is a tough job.
- As the number of dimensions increases its scalability decreases.
- It is not suitable to identify clusters with non-convex shapes.

3. 2 K Medoid

- A **medoid** can be defined as the point in the cluster, whose dissimilarities with all the other points in the cluster is minimum. That is, it is a most centrally located point in the cluster.
- **PAM -Partitioning Around Medoids**(Kaufmann & Rousseeuw 1987)
- Splits the data set of n objects into k clusters, where the number k of clusters assumed known *a priori* (which implies that the programmer must specify)
- Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
- *PAM* works effectively for small data sets, but does not scale well for large data sets.
- The dissimilarity of the medoid(C_i) and object(P_i) is calculated by using $E = |P_i - C_i|$
- The runtime complexity of the original PAM algorithm per iteration of is $O(k(n-k)^2)$

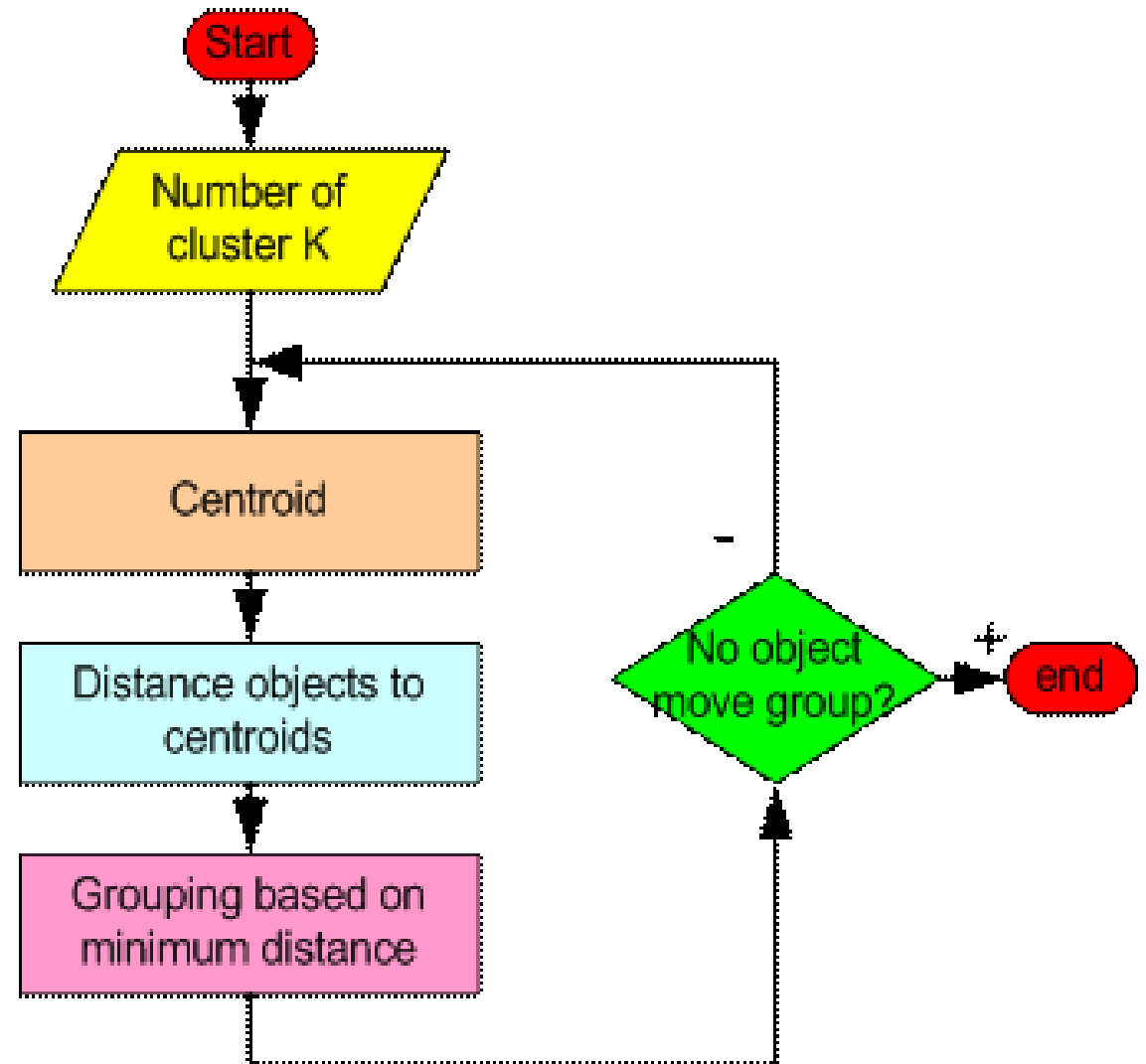
ALGORITHM

Build phase:

1. Select k objects to become the medoids, or in case these objects were provided use them as the medoids;
2. Calculate the dissimilarity matrix if it was not provided;
3. Assign every object to its closest medoid;

Swap phase:

4. For each cluster compute the distance between the non-medoid data point o and medoid m .
 5. Consider the swap of m and o , and compute the cost change. See if it decreases the average dissimilarity coefficient; if it does, select the entity that decreases this coefficient
 6. If the cost change is the current best, remember this m and o combination
- Medoid process is continued until no any medoid move



EXAMPLE

- Cluster the given data set of ten objects into two clusters i.e. $k=2$. Let the initial medoids be $(3,4)$ and $(7,4)$

X	Y
2	6
3	4
3	8
4	7
6	2
6	4
7	3
7	4
8	5
7	6

EXAMPLE

- Cluster the given data set of ten objects into two clusters i.e. $k=2$. Let the initial medoids be (3,4) and (7,4)

x	y	D1(3,4)	D2(7,4)	Cluster
2	6	3	7	C1
3	4	0	4	C1
3	8	4	8	C1
4	7	4	6	C1
6	2	5	3	C2
6	4	3	1	C2
7	3	5	1	C2
7	4	4	0	C2
8	5	6	2	C2
7	6	6	2	C2

Cluster1 = {(3,4) (2,6) (3,8) (4,7)

Cluster2 = {(7,4) (6,2) (6,4) (7,3) 1]

Cost (Distance)
= 3+0+4+4+3+1+1+0+2+2
= 20

EXAMPLE

- Let the new medoid O' be $(7,3)$. So new medoids are $(3,4)$ and $(7,3)$

x	y	D1(3,4)	D2(7,3)	Cluster
2	6	3	8	C1
3	4	0	5	C1
3	8	4	9	C1
4	7	4	7	C1
6	2	5	2	C2
6	4	3	2	C2
7	3	5	0	C2
7	4	4	1	C2
8	5	6	3	C2
7	6	6	4	C2

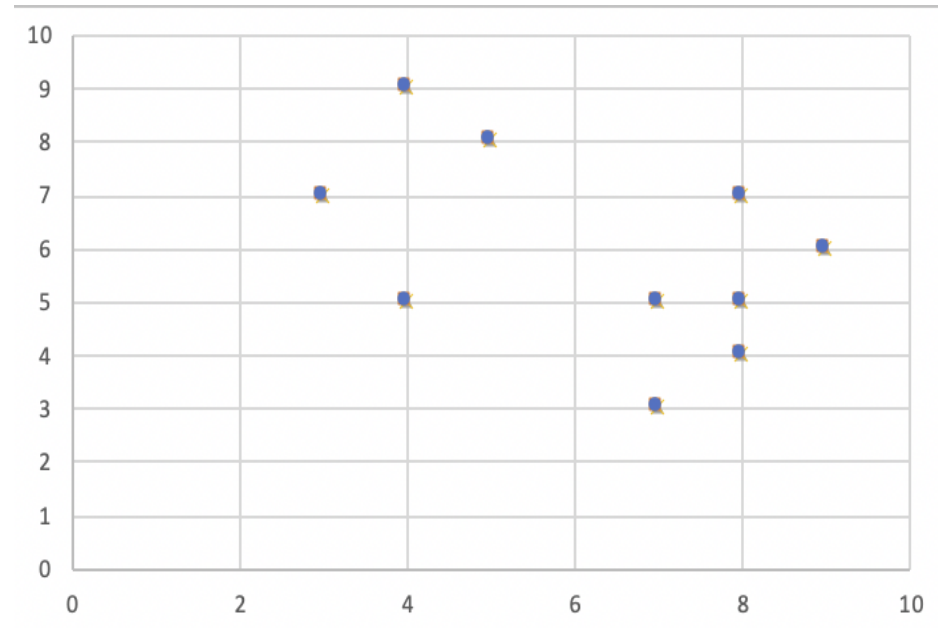
Cluster1 = $\{(3,4) (2,6) (3,8) (4,7)$

Cluster2 = $\{(7,4) (6,2) (6,4) (7,3) 1\}$

Cost (Distance)
 $= 3 + 0 + 4 + 4 + 2 + 2 + 0 + 1 + 3 + 4$
 $= 22$

EXAMPLE

	X	Y
0	8	7
1	3	7
2	4	9
3	9	6
4	8	5
5	5	8
6	7	3
7	8	4
8	7	5
9	4	5



Let the randomly selected 2 medoids, so select $k = 2$ and let **C1** -(4, 5) and **C2** -(8, 5) are the two medoids.

EXAMPLE

	X	Y	Dissimilarity from C1	Dissimilarity from C2
0	8	7	6	2
1	3	7	3	7
2	4	9	4	8
3	9	6	6	2
4	8	5	-	-
5	5	8	4	6
6	7	3	5	3
7	8	4	5	1
8	7	5	3	1
9	4	5	-	-

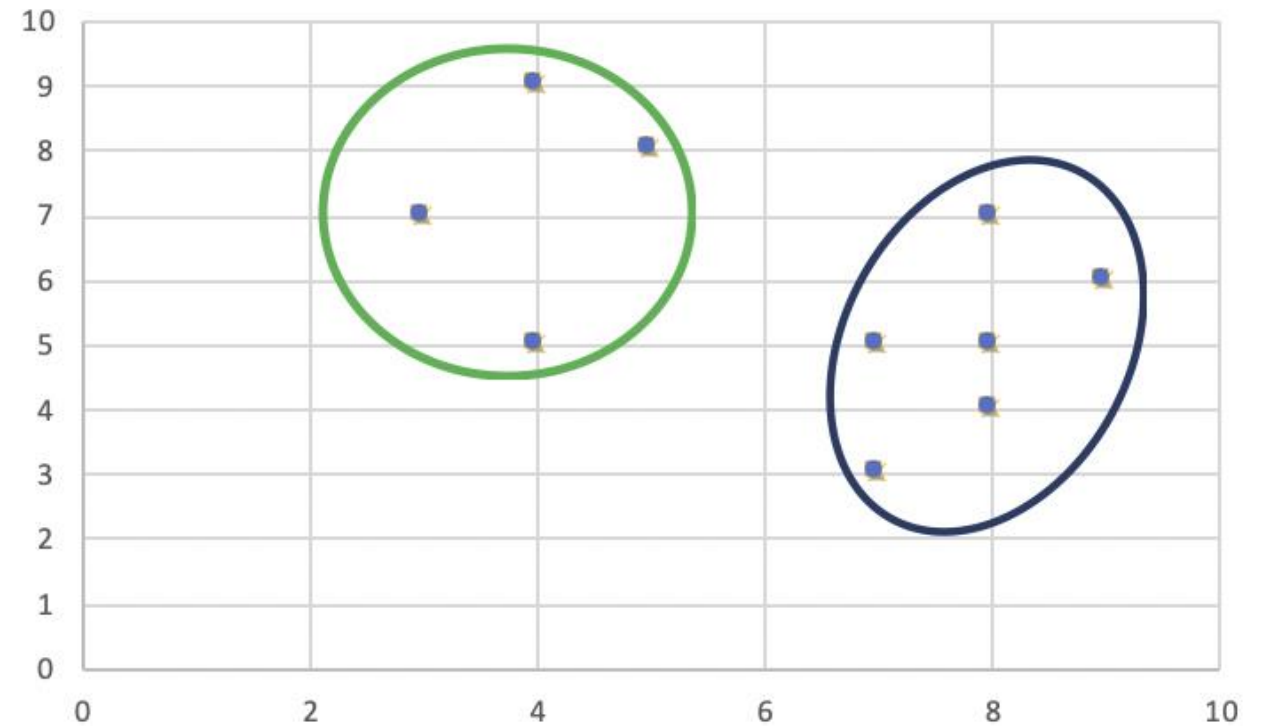
The points 1, 2, 5 go to cluster C1 and 0, 3, 6, 7, 8 go to cluster C2. The Cost = $(3 + 4 + 4) + (2 + 2 + 3 + 1 + 1) = 20$

	X	Y	Dissimilarity from C1	Dissimilarity from C2
0	8	7	6	3
1	3	7	3	8
2	4	9	4	9
3	9	6	6	3
4	8	5	4	1
5	5	8	4	7
6	7	3	5	2
7	8	4	-	-
8	7	5	3	2
9	4	5	-	-

Let the randomly selected point be (8, 4). The dissimilarity of each non-medoid point with the medoids – C1 (4, 5) and C2 (8, 4) is calculated and tabulated.

EXAMPLE

- Each point is assigned to that cluster whose dissimilarity is less. So, the points 1, 2, 5 go to cluster C1 and 0, 3, 6, 7, 8 go to cluster C2.
The New cost = $(3 + 4 + 4) + (2 + 2 + 1 + 3 + 3) = 22$
Swap Cost = New Cost - Previous Cost = $22 - 20$ and $2 > 0$
- As the swap cost is not less than zero, we undo the swap. Hence (3, 4) and (7, 4) are the final medoids.



Advantages and Disadvantages of K-Medoid Clustering

Advantages:


- It is simple to understand and easy to implement.
- K-Medoid Algorithm is fast and converges in a fixed number of steps.
- PAM is less sensitive to outliers than other partitioning algorithms.

Disadvantages:

- The main disadvantage of K-Medoid algorithms is that it is not suitable for clustering non-spherical (arbitrary shaped) groups of objects. This is because it relies on minimizing the distances between the non-medoid objects and the medoid (the cluster centre) – briefly, it uses compactness as clustering criteria instead of connectivity.
- It may obtain different results for different runs on the same dataset because the first k medoids are chosen randomly.



Challenge



**Given Datapoints: {2,4,10,12,3,20,30,11,25}
and Cluster number : 2. Apply K-Means and
K-Medoid Algorithm**

k-mean

- Datapoints: {2,4,10,12,3,20,30,11,25}
- Cluster number : 2. Initial centroids 2 and 3
- **Step 1:** Value of clusters K1{ 2 } K2{ 3 6 8 12 15 18 22 25 }
- Centroid m1 and m2 : mean values
m1=2.0 m2=13.625 . Distance:
- **Step 2:** Value of clusters K1 {2 3 6} K2 {8 12 15 18 22 25} Value of m m1=3.66 m2=16.66
- **Step 3:** Value of clusters K1 {2 3 6 8 } K2 {12 15 18 22 25} Value of m m1=4.75 m2=18.4
- **Step 4:** Value of clusters K1 {2 3 6 8 } K2 {12 15 18 22 25} Value of m m1=4.75 m2=18.4
- **Step 5:** The final clusters by K-means are as follows: K1 {2 3 6 8 } K2 {12 15 18 22 25}

k-medoid

- Given data points: {2, 4, 10, 12, 3, 20, 30, 11, 25}
- Number of clusters k=2 . Initial medoids 2 and 3
- **Step: 1:** {2} {3 6 8 12 15 18 22 25} C: 2&3 : S=85
- **Step: 2:** {2 3 } {6 8 12 15 18 22 25} C:2 &6 : S=65
- **Step: 3:** {2 3} {8 6 12 15 18 22 25} C:2&8 : S=55
- **Step: 4:** {2 3 6} {12 8 15 18 22 25} C:2&12 : S= 41
- **Step: 5:** {2 3 6 8} {15 12 18 22 25} C:2 &18 : S=31
- **Step: 6:** {2 3 6 8} {18 12 15 22 25 } C:2 & 15: S=34.
Stop Iteration as cost increases
- **Step: 7:** The final clusters by K-Medoids are as follows: {2 3 6 8} {18 12 15 22 25}

k-mean

1. In K-means, the center of a cluster is not necessarily one of the input data points . It is the average/mean between the points in the cluster/data to compute centroid.
2. K-means generally requires Euclidean distance for efficient solutions.
3. Tries to minimize total squared distance or error
4. Less robust to noise and outliers

k-medoid

1. In K-medoids chooses actual data points as centers (medoids or exemplars), which are selected randomly and thereby allows for greater interpretability of the cluster centers. remaining all data objects are placed in a cluster having medoid nearest (or most similar) to that data object
2. K-medoids can be used with arbitrary dissimilarity measures, because k-medoids minimizes a sum of pairwise dissimilarities instead of a sum of squared Euclidean distances
3. Tries to minimize the absolute distance between the points and the selected centroid(sum of dissimilarities between points labeled to be in a cluster and a point designated as the center of that cluster)
4. more robust to noise and outliers

4. Discretization using Correlation analysis

- **Correlation analysis (e.g., Chi-merge: χ^2 -based discretization)**
 - **Supervised:** use class information
 - **Bottom-up merge:** find the best neighboring intervals (those having similar distributions of classes, i.e., low χ^2 values) to merge
 - Also known as **Chi Merge algorithm** performs recursively by finding best neighboring intervals that have similar distribution of classes and merge them together, until a predefined stopping condition
 - Steps:
 - Each distinct value of a numeric attribute A is considered to be one interval. Chi-2 tests are performed for every pair of adjacent intervals.
 - Adjacent intervals with the **least Chi-2 values** are merged together, because low Chi-2 values for a pair indicate similar class distributions.
 - This merging process proceeds recursively until a predefined stopping criterion is met.

χ^2 -based

- χ^2 (chi-square) test

Pearson χ^2 statistic

$$\chi^2 = \sum \frac{(\text{Observed} - \text{Expected})^2}{\text{Expected}}$$

- The larger the χ^2 value, the more likely the variables are related
- Expected frequency is calculated as $e_{ij} = \text{count}(A=a_i) * \text{count}(B=b_j) / n$. where n is the number of data tuples, $\text{count}(A=a_i)$ is the number of tuples having value a_i for A , and $\text{count}(B=b_j)$ is the number of tuples having value b_j for B
- The cells that contribute the most to the χ^2 value are those whose actual count is very different from the expected count.
- χ^2 statistic tests the hypothesis that A and B are independent, that is, there is no correlation between them.
- The test is based on a significance level, with $(r-1)*(c-1)$ degrees of freedom.
- For **nominal data** use χ^2 (chi-square) test. For **numeric attributes** use the **correlation coefficient and covariance**.



EXAMPLE

- Suppose that a group of 1500 people was surveyed. The gender of each person was noted. Each person was polled as to whether his or her preferred type of reading material was fiction or nonfiction. we have two attributes, ***gender*** and ***preferred reading***. The observed frequency (or count) of each possible joint event is summarized in the contingency table. Identify whether ***gender*** and ***preferred reading*** is correlated.

	MALE	FEMALE	TOTAL
FICTION	250	200	450
NON-FICTION	50	1000	1050
TOTAL	300	1200	1500

EXAMPLE

- expected frequency for the cell (male, fiction) is

$$e_{11} = \frac{\text{count}(\text{male}) \times \text{count}(\text{fiction})}{n} = \frac{300 \times 450}{1500} = 90,$$

- $e_{12}=360$ $e_{21}=210$ $e_{22}=840$

$$\begin{aligned}\chi^2 &= \frac{(250 - 90)^2}{90} + \frac{(50 - 210)^2}{210} + \frac{(200 - 360)^2}{360} + \frac{(1000 - 840)^2}{840} \\ &= 284.44 + 121.90 + 71.11 + 30.48 = 507.93.\end{aligned}$$

- Degrees of freedom for given table = $(2-1) \times (2-1) = 1$.
- For 1 degree of freedom, the χ^2 value needed to reject the hypothesis at the 0.001 significance level is 10.828 (taken from the table of upper percentage points of the χ^2 distribution).
- Since our computed value is above this, we can reject the hypothesis that gender and preferred reading are independent and conclude that the two attributes are (strongly) correlated for the given group of people

Chi-Square table

	P										
DF	0.995	0.975	0.2	0.1	0.05	0.025	0.02	0.01	0.005	0.002	0.001
1	.0004	.00016	1.642	2.706	3.841	5.024	5.412	6.635	7.879	9.55	10.828
2	0.01	0.0506	3.219	4.605	5.991	7.378	7.824	9.21	10.597	12.429	13.816
3	0.0717	0.216	4.642	6.251	7.815	9.348	9.837	11.345	12.838	14.796	16.266
4	0.207	0.484	5.989	7.779	9.488	11.143	11.668	13.277	14.86	16.924	18.467
5	0.412	0.831	7.289	9.236	11.07	12.833	13.388	15.086	16.75	18.907	20.515
6	0.676	1.237	8.558	10.645	12.592	14.449	15.033	16.812	18.548	20.791	22.458
7	0.989	1.69	9.803	12.017	14.067	16.013	16.622	18.475	20.278	22.601	24.322
8	1.344	2.18	11.03	13.362	15.507	17.535	18.168	20.09	21.955	24.352	26.124
9	1.735	2.7	12.242	14.684	16.919	19.023	19.679	21.666	23.589	26.056	27.877
10	2.156	3.247	13.442	15.987	18.307	20.483	21.161	23.209	25.188	27.722	29.588
11	2.603	3.816	14.631	17.275	19.675	21.92	22.618	24.725	26.757	29.354	31.264
12	3.074	4.404	15.812	18.549	21.026	23.337	24.054	26.217	28.3	30.957	32.909
13	3.565	5.009	16.985	19.812	22.362	24.736	25.472	27.688	29.819	32.535	34.528
14	4.075	5.629	18.151	21.064	23.685	26.119	26.873	29.141	31.319	34.091	36.123
15	4.601	6.262	19.311	22.307	24.996	27.488	28.259	30.578	32.801	35.628	37.697
16	5.142	6.908	20.465	23.542	26.296	28.845	29.633	32	34.267	37.146	39.252
17	5.697	7.564	21.615	24.769	27.587	30.191	30.995	33.409	35.718	38.648	40.79
18	6.265	8.231	22.76	25.989	28.869	31.526	32.346	34.805	37.156	40.136	42.312
19	6.844	8.907	23.9	27.204	30.144	32.852	33.687	36.191	38.582	41.61	43.82
20	7.434	9.591	25.038	28.412	31.41	34.17	35.02	37.566	39.997	43.072	45.315

Chi Merge Discretization Process

1. Initially, each distinct value of a numerical attribute is considered to be one interval.
2. χ^2 tests are performed for every pair of adjacent intervals.
3. Adjacent intervals with the least χ^2 values are merged together, because low χ^2 values for a pair indicates similar class distributions. This merging process proceeds recursively until a predefined stopping criterion is met.

Chi Merge Example 1

<i>X</i>	<i>Y</i>	<i>CLASS</i>
<i>1</i>	<i>2</i>	<i>A</i>
<i>3</i>	<i>4</i>	<i>B</i>
<i>5</i>	<i>6</i>	<i>A</i>
<i>7</i>	<i>8</i>	<i>B</i>
<i>9</i>	<i>10</i>	<i>A</i>
<i>11</i>	<i>12</i>	<i>B</i>
<i>13</i>	<i>14</i>	<i>A</i>

A process that transforms quantitative data into qualitative data.

EXAMPLE

- **Step — 1**
- Split the datasets into 2 datasets and find values separately.
- Data Set 1 → X, Class
- Data Set 2 → Y, Class

DataSet			
DataSet 1		DataSet 2	
X	CLASS	Y	CLASS
1	A	2	A
3	B	4	B
5	A	6	A
7	B	8	B
9	A	10	A
11	B	12	B
13	A	14	A

- **Step 2 → Find ChiSquare for X attribute**

Let's take DataSet 1 and find the interval which is obtained by summing the adjacent rows and dividing by 2.

X	CLASS	Interval
1	A	$= (1 + 3) / 2 = 2$ 0 is Min and 2 is Max $= [0, 2]$
3	B	[2, 4]
5	A	[4, 6]
7	B	[6, 8]
9	A	[8, 10]
11	B	[10, 12]

○ 2.1 Contingency Tables for X attribute

Contingency Tables for X attribute			
Contingency Tables for the intervals [0, 2] and [2, 4]			
	Class A	Class B	Sums
Interval [0, 2]	1	0	1
Interval [2, 4]	0	1	1
Sums	1	1	2
<p>Expected Value → (Row Sum * Column Sum) / Total Sum</p> <p> $E_{11} \rightarrow (1 * 1) / 2 = 0.5$ $E_{12} \rightarrow (1 * 1) / 2 = 0.5$ $E_{21} \rightarrow (1 * 1) / 2 = 0.5$ $E_{22} \rightarrow (1 * 1) / 2 = 0.5$ </p> <p> $\chi^2 \rightarrow [(1 - 0.5)^2 / 0.5 + (0 - 0.5)^2 / 0.5 + (0 - 0.5)^2 / 0.5 + (1 - 0.5)^2 / 0.5]$ $\chi^2 = 2$ </p>			
Contingency Tables for the intervals [2, 4] and [4, 6]			
	Class A	Class B	Sums
Interval [2, 4]	0	1	1
Interval [4, 6]	1	0	1
Sums	1	1	2
<p>Expected Value → (Row Sum * Column Sum) / Total Sum</p> <p> $E_{11} \rightarrow (1 * 1) / 2 = 0.5$ $E_{12} \rightarrow (1 * 1) / 2 = 0.5$ $E_{21} \rightarrow (1 * 1) / 2 = 0.5$ $E_{22} \rightarrow (1 * 1) / 2 = 0.5$ </p>			

$\chi^2 \rightarrow [(0 - 0.5)^2 / 0.5 + (1 - 0.5)^2 / 0.5 + (1 - 0.5)^2 / 0.5 + (0 - 0.5)^2 / 0.5]$ $\chi^2 = 2$			
Contingency Tables for the intervals [4, 6] and [6, 8]			
	Class A	Class B	Sums
Interval [4, 6]	1	0	1
Interval [6, 8]	0	1	1
Sums	1	1	2
<p>Expected Value → (Row Sum * Column Sum) / Total Sum</p> <p> $E_{11} \rightarrow (1 * 1) / 2 = 0.5$ $E_{12} \rightarrow (1 * 1) / 2 = 0.5$ $E_{21} \rightarrow (1 * 1) / 2 = 0.5$ $E_{22} \rightarrow (1 * 1) / 2 = 0.5$ </p> <p> $\chi^2 \rightarrow [(1 - 0.5)^2 / 0.5 + (0 - 0.5)^2 / 0.5 + (0 - 0.5)^2 / 0.5 + (1 - 0.5)^2 / 0.5]$ $\chi^2 = 2$ </p>			
Contingency Tables for the intervals [6, 8] and [8, 10]			
	Class A	Class B	Sums
Interval [6, 8]	0	1	1
Interval [8, 10]	1	0	1
Sums	1	1	2
<p>Expected Value → (Row Sum * Column Sum) / Total Sum</p> <p> $E_{11} \rightarrow (1 * 1) / 2 = 0.5$ $E_{12} \rightarrow (1 * 1) / 2 = 0.5$ $E_{21} \rightarrow (1 * 1) / 2 = 0.5$ $E_{22} \rightarrow (1 * 1) / 2 = 0.5$ </p>			

$$X^2 \rightarrow [(0 - 0.5)^2 / 0.5 + (1 - 0.5)^2 / 0.5 + (1 - 0.5)^2 / 0.5 + (0 - 0.5)^2 / 0.5]$$

$$X^2 = 2$$

Contingency Tables for the intervals [8, 10] and [10, 12]

	Class A	Class B	Sums
Interval [8, 10]	1	0	1
Interval [10, 12]	0	1	1
Sums	1	1	2

Expected Value \rightarrow (Row Sum * Column Sum) / Total Sum

$$E_{11} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{12} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{21} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{22} \rightarrow (1 * 1) / 2 = 0.5$$

$$X^2 \rightarrow [(1 - 0.5)^2 / 0.5 + (0 - 0.5)^2 / 0.5 + (0 - 0.5)^2 / 0.5 + (1 - 0.5)^2 / 0.5]$$

$$X^2 = 2$$

Contingency Tables for the intervals [10, 12] and [12, 14]

	Class A	Class B	Sums
Interval [10, 12]	0	1	1
Interval [12, 14]	1	0	1
Sums	1	1	2

Expected Value \rightarrow (Row Sum * Column Sum) / Total Sum

$$E_{11} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{12} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{21} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{22} \rightarrow (1 * 1) / 2 = 0.5$$

$$X^2 \rightarrow [(0 - 0.5)^2 / 0.5 + (1 - 0.5)^2 / 0.5 + (1 - 0.5)^2 / 0.5 + (0 - 0.5)^2 / 0.5]$$

$$X^2 = 2$$

We cannot merge further because all the X^2 values are the same (2).

X	CLASS	Interval	X^2
1	A	= (1 + 3) / 2 = 2 0 is Min and 2 is Max = [0, 2]	2
3	B	[2, 4]	2
5	A	[4, 6]	2
7	B	[6, 8]	2
9	A	[8, 10]	2
11	B	[10, 12]	2
13	A	[12, 14]	2

d=degrees of freedom = 1, Threshold (for $\alpha=10\%$)=2.706

All intervals are in same value, so stopping with this merging

$X^2 = 2$. Significant differences \rightarrow No merging

FINAL RESULT : 7 intervals [0, 2], [2, 4], [4, 6], [6, 8], [8, 10], [10, 12], [12, 14]

- **Step 3 → Find ChiSquare for Y attribute**

Let's take DataSet 2 and find the interval which is obtained by summing the adjacent rows and dividing by 2.

Y	CLASS	Intervals
2	A	$= (2 + 4) / 2 = 3$ 1 is Min and 3 is Max $= [1, 3]$
4	B	$[3, 5]$
6	A	$[5, 7]$
8	B	$[7, 9]$
10	A	$[9, 11]$
12	B	$[11, 13]$
14	A	$[13, 15]$

Find contingency table for Y attribute

Contingency Tables for Y attribute			
Contingency Tables for the intervals [1, 3] and [3, 5]			
	Class A	Class B	Sums
Interval [1, 3]	1	0	1
Interval [3, 5]	0	1	1
Sums	1	1	2

Expected Value → $(\text{Row Sum} * \text{Column Sum}) / \text{Total Sum}$

$$E_{11} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{12} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{21} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{22} \rightarrow (1 * 1) / 2 = 0.5$$

$$X^2 \rightarrow [(1 - 0.5)^2 / 0.5 + (0 - 0.5)^2 / 0.5 + (0 - 0.5)^2 / 0.5 + (1 - 0.5)^2 / 0.5]$$

$$X^2 = 2$$

Contingency Tables for the intervals [3, 5] and [5, 7]

	Class A	Class B	Sums
Interval [3, 5]	0	1	1
Interval [5, 7]	1	0	1
Sums	1	1	2

Expected Value → $(\text{Row Sum} * \text{Column Sum}) / \text{Total Sum}$

$$E_{11} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{12} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{21} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{22} \rightarrow (1 * 1) / 2 = 0.5$$

$$X^2 \rightarrow [(0 - 0.5)^2 / 0.5 + (1 - 0.5)^2 / 0.5 + (1 - 0.5)^2 / 0.5 + (0 - 0.5)^2 / 0.5]$$

$$X^2 = 2$$

Contingency Tables for the intervals [5, 7] and [7, 9]

	Class A	Class B	Sums
Interval [5, 7]	1	0	1
Interval [7, 9]	0	1	1
Sums	1	1	2

Expected Value \rightarrow (Row Sum * Column Sum) / Total Sum

$$E_{11} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{12} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{21} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{22} \rightarrow (1 * 1) / 2 = 0.5$$

$$X^2 \rightarrow [(1 - 0.5)^2 / 0.5 + (0 - 0.5)^2 / 0.5 + (0 - 0.5)^2 / 0.5 + (1 - 0.5)^2 / 0.5]$$

$$X^2 = 2$$

Contingency Tables for the intervals [7, 9] and [9, 11]

	Class A	Class B	Sums
Interval [7, 9]	0	1	1
Interval [9, 11]	1	0	1
Sums	1	1	2

Expected Value \rightarrow (Row Sum * Column Sum) / Total Sum

$$E_{11} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{12} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{21} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{22} \rightarrow (1 * 1) / 2 = 0.5$$

$$X^2 \rightarrow [(0 - 0.5)^2 / 0.5 + (1 - 0.5)^2 / 0.5 + (1 - 0.5)^2 / 0.5 + (0 - 0.5)^2 / 0.5]$$

$$X^2 = 2$$

Contingency Tables for the intervals [9, 11] and [11, 13]

	Class A	Class B	Sums
Interval [9, 11]	1	0	1
Interval [11, 13]	0	1	1
Sums	1	1	2

Expected Value \rightarrow (Row Sum * Column Sum) / Total Sum

$$E_{11} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{12} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{21} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{22} \rightarrow (1 * 1) / 2 = 0.5$$

$$X^2 \rightarrow [(1 - 0.5)^2 / 0.5 + (0 - 0.5)^2 / 0.5 + (0 - 0.5)^2 / 0.5 + (1 - 0.5)^2 / 0.5]$$

$$X^2 = 2$$

Contingency Tables for the intervals [11, 13] and [13, 15]

	Class A	Class B	Sums
Interval [11, 13]	0	1	1
Interval [13, 15]	1	0	1
Sums	1	1	2

Expected Value \rightarrow (Row Sum * Column Sum) / Total Sum

$$E_{11} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{12} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{21} \rightarrow (1 * 1) / 2 = 0.5$$

$$E_{22} \rightarrow (1 * 1) / 2 = 0.5$$

$$X^2 \rightarrow [(0 - 0.5)^2 / 0.5 + (1 - 0.5)^2 / 0.5 + (1 - 0.5)^2 / 0.5 + (0 - 0.5)^2 / 0.5]$$

$$X^2 = 2$$

We cannot merge further because all the X^2 values are the same (2).

Y	CLASS	Interval	X^2
2	A	$= (2 + 4) / 2 = 3$ 1 is Min and 3 is Max $= [1, 3]$	2
4	B	$[3, 5]$	2
6	A	$[5, 7]$	2
8	B	$[7, 9]$	2
10	A	$[9, 11]$	2
12	B	$[11, 13]$	2
14	A	$[13, 15]$	2

Probability less than the critical value					
ν	0.90	0.95	0.975	0.99	0.999
1	2.706	3.841	5.024	6.635	10.828
2	4.605	5.991	7.378	9.210	13.816

Conclusion:

From the above probability distribution table, concluding that

For d =degrees of freedom = 1, Threshold (for α =10%)=2.706

The dataset intervals are in same value, so stopping with this merging

$X^2 = 2$. Significant differences \rightarrow No merging

FINAL RESULT : 7 intervals $\rightarrow [1, 3], [3, 5], [5, 7], [7, 9], [9, 11], [11, 13], [13, 15]$

Attribute Values	Intervals	X^2
$X \rightarrow 1, 3, 5, 7, 9, 11, 13$	$[0, 2], [2, 4], [4, 6], [6, 8], [8, 10], [10, 12], [12, 14]$	2
$Y \rightarrow 2, 4, 6, 8, 10, 12, 14$	$[1, 3], [3, 5], [5, 7], [7, 9], [9, 11], [11, 13], [13, 15]$	2

ChiMerge

Discretization

Example 2

- Statistical approach to Data Discretization
- Applies the Chi Square method to determine the probability of similarity of data between two intervals.

Sample	F	K
1	1	1
2	3	2
3	7	1
4	8	1
5	9	1
6	11	2
7	23	2
8	37	1
9	39	2
10	45	1
11	46	1
12	59	1

Sample	F	K
1	1	1
2	3	2
3	7	1
4	8	1
5	9	1
6	11	2
7	23	2
8	37	1
9	39	2
10	45	1
11	46	1
12	59	1

Intervals

{0,2}

{2,5}

{5,7.5}

{7.5,8.5}

{8.5,10}

{10,17}

{17,30}

{30,38}

{38,42}

{42,45.5}

{45.5,52}

{52,60}

ChiMerge Discretization Example

- Find chi square for F attributes. Sort and order the attributes that you want to group (in this example attribute F).
- Find the intervals by considering adjacent rows and dividing it by 2.
- Start with having every unique value in the attribute be in its own interval.

ChiMerge Discretization

Example

- Begin calculating the Chi Square test on every interval

Sample	F	K
1	1	1
2	3	2
3	7	1
4	8	1
5	9	1
6	11	2
7	23	2
8	37	1
9	39	2
10	45	1
11	46	1
12	59	1

Sample	K=1	K=2	
2	0	1	1
3	1	0	1
total	1	1	2

Sample	K=1	K=2	
3	1	0	1
4	1	0	1
total	2	0	2

ChiMerge Discretization Example

Sample	K=1	K=2	
2	0	1	1
3	1	0	1
total	1	1	2

$$E_{11} = (1/2)*1 = .05$$

$$E_{12} = (1/2)*1 = .05$$

$$E_{21} = (1/2)*1 = .05$$

$$E_{22} = (1/2)*1 = .05$$

$$X^2 = (0-.5)^2/.5 + (0-.5)^2/.5 + (0-.5)^2/.5 + (0-.5)^2/.5 = 2$$

Sample	K=1	K=2	
3	1	0	1
4	1	0	1
total	2	0	2

$$E_{11} = (1/2)*2 = 1$$

$$E_{12} = (0/2)*2 = 0$$

$$E_{21} = (1/2)*2 = 1$$

$$E_{22} = (0/2)*2 = 0$$

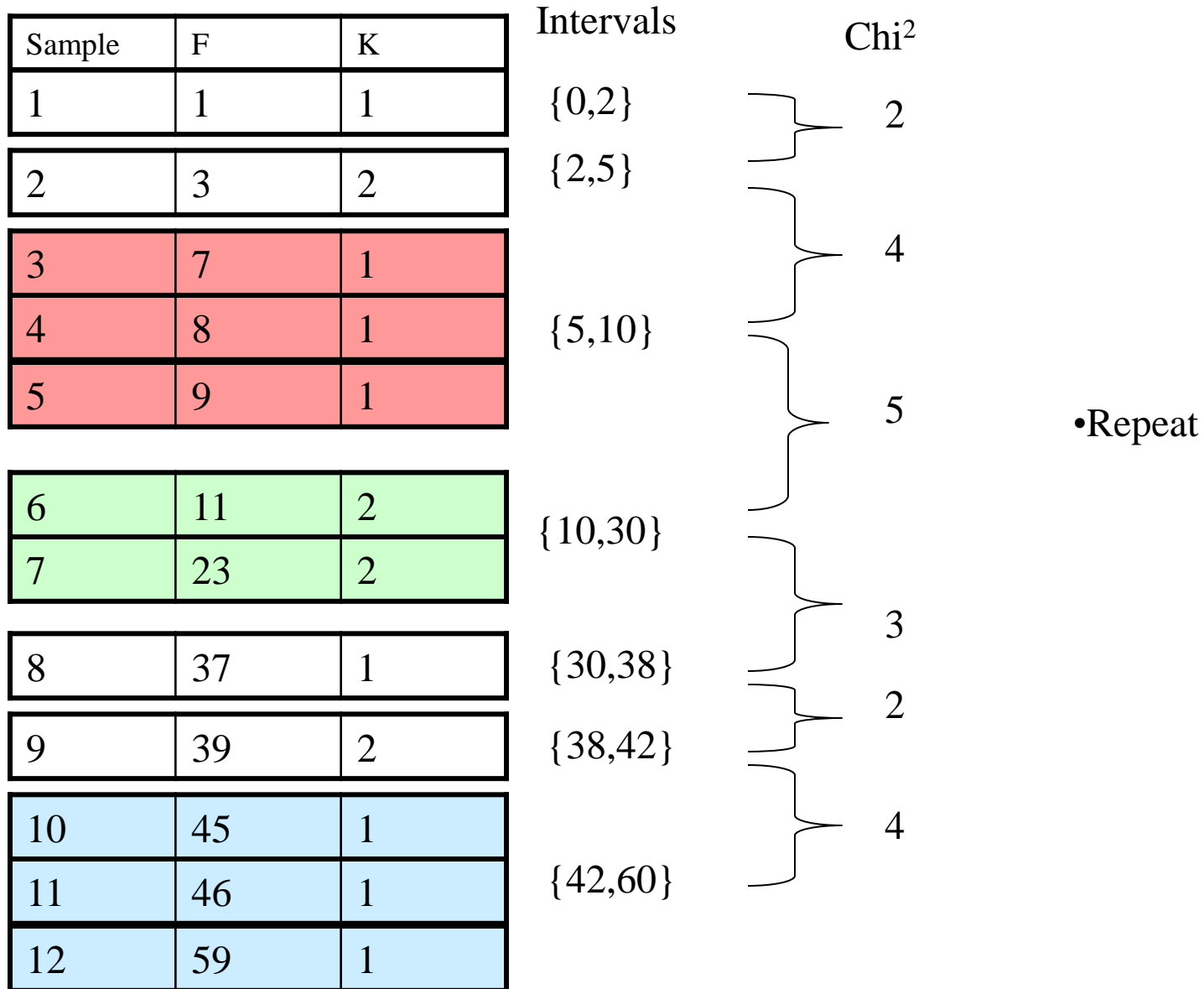
$$X^2 = (1-1)^2/1 + (0-0)^2/0 + (1-1)^2/1 + (0-0)^2/0 = 0$$

Threshold .1 with df=1 from Chi square distribution chart merge if $X^2 < 2.7024$

ChiMerge Discretization Example

Sample	F	K	Intervals	Chi ²	
1	1	1	{0,2}	2	•Calculate all the Chi Square value for all intervals
2	3	2	{2,5}	2	
3	7	1	{5,7.5}	0	
4	8	1	{7.5,8.5}	0	
5	9	1	{8.5,10}	2	•Merge the intervals with the smallest Chi values
6	11	2	{10,17}	0	
7	23	2	{17,30}	2	
8	37	1	{30,38}	2	
9	39	2	{38,42}	2	
10	45	1	{42,45.5}	0	
11	46	1	{45.5,52}	0	
12	59	1	{52,60}	0	

ChiMerge Discretization Example



ChiMerge Discretization Example

Sample	F	K
1	1	1
2	3	2

3	7	1
4	8	1
5	9	1

6	11	2
7	23	2

8	37	1
9	39	2

10	45	1
11	46	1
12	59	1

Intervals

{0,5}

{5,10}

{10,30}

{30,42}

{42,60}

Chi²

1.875

5

1.33

1.875

•Again

ChiMerge Discretization Example

Sample	F	K	Intervals	Chi ²
1	1	1	{0,5}	1.875
2	3	2		
3	7	1	{5,10}	3.93
4	8	1		
5	9	1		
6	11	2	{10,30}	3.93
7	23	2		
8	37	1		
9	39	2		
10	45	1	{42,60}	
11	46	1		
12	59	1		

•Until

ChiMerge Discretization Example

Sample	F	K
1	1	1
2	3	2
3	7	1
4	8	1
5	9	1

Intervals χ^2

{0,10}

2.72

6	11	2
7	23	2
8	37	1
9	39	2

{10,30}

3.93

10	45	1
11	46	1
12	59	1

{42,60}

•There are no more intervals that can satisfy the Chi Square test.