# CSCD 429: Data Mining

## —Data Preprocessing—

Dr. Dan Li

# Assignments

- Reading Assignment
  - Chapter 3 from the book
- Homework 1
  - Exercises from Chapters 2 & 3
- Lab 3
  - Using RapidMiner for data preprocessing

# Chapter 3: Data Preprocessing

- Data Preprocessing: An Overview
  - Why Preprocess the Data?
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation
- Summary

# Data Quality: Why Preprocess the Data?

- GIGO - good data is a prerequisite for producing effective models of any type
- Data need to be formatted for a given software tool
- Data need to fit in a given method
- Data in the real world is dirty
  - incomplete: lacking attribute values
  - noisy: containing errors or outliers
  - inconsistent: containing discrepancies in codes or names

# Major Tasks in Data Preprocessing

- **Data cleaning**
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- **Data integration**
  - Integration from multiple sources
- **Data reduction**
  - Dimensionality reduction
  - Numerosity reduction
  - Data compression
- **Data transformation**
  - Normalization
  - Data discretization
  - Concept hierarchy generation

# Chapter 3: Data Preprocessing

- Data Preprocessing: An Overview
  - Data Quality
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation
- Summary

# Data Cleaning

- Data in the Real World Is Dirty: Lots of potentially incorrect data, e.g., instrument faulty, human or computer error, transmission error
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., *Occupation*=“ ” (missing data)
  - noisy: containing noise, errors, or outliers
    - e.g., *Salary*=“−10” (an error)
  - inconsistent: containing discrepancies in codes or names, e.g.,
    - *Age*=“42”, *Birthday*=“03/07/2010”
    - Was rating “1, 2, 3”, now rating “A, B, C”
    - discrepancy between duplicate records
  - Intentional (e.g., *disguised missing* data)
    - Jan. 1 as everyone’s birthday?

# Incomplete (Missing) Data

- Data is not always available
  - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to
  - equipment malfunction
  - data not entered due to misunderstanding
  - certain data may not be considered important at the time of entry

# How to Handle Missing Data?

- Ignore the tuples: usually done when class label is missing (when doing classification)
  - Not effective when the % of missing values per attribute varies considerably
- Ignore the attributes with missing values
  - May leave out important attributes
- Fill in the missing value manually
  - tedious + infeasible

# How to Handle Missing Data?

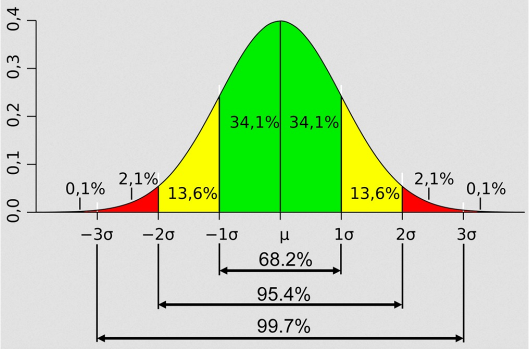
- Fill in it automatically with
  - a global constant : e.g., “unknown”, a new class?!
  - the attribute mean, median or mode
  - the attribute mean/median/mode for all samples belonging to the same class: smarter
  - the most probable value, i.e., imputation
    - Use existing attributes to fill in missing attributes, e.g. KNN
    - Identify relationship among variables, e.g. regression
- Rule of thumb: avoid adding bias!!
  - Error rate on training data shouldn’t be increased after filling in missing values

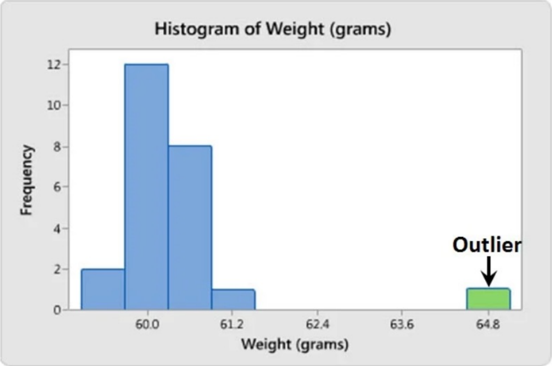
# Noisy Data

- Noise/Outliers: values thought to be out of range or different from most of other observations.
- Incorrect attribute values may be due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention

# Outlier Detection

- Univariate
  - Use mean and standard deviation
  - Use histogram analysis
  - Use boxplot





- Multivariate
  - Regression: smooth by fitting the data into regression functions
  - Clustering: detect and remove outliers

# Chapter 3: Data Preprocessing

- Data Preprocessing: An Overview
  - Data Quality
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation
- Summary

# Data Integration

- **Data integration**
  - Combines data from multiple sources into a coherent store
- Schema integration
  - Integrate metadata from different sources, e.g., A.cust-id ≡ B.cust-#
- Entity integration
  - For the same real world entity, attribute values from different sources are different, e.g., Bill Clinton = William Clinton
  - Possible reasons: different representations, different scales, e.g., metric vs. British units

# Handling Redundancy in Data Integration

- Redundant attributes occur often when integration of multiple databases
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality
- Redundant attributes may be able to be detected by covariance analysis and correlation analysis
  - Correlation analysis of categorical data
  - Covariance analysis of numeric data
  - Correlation analysis of numeric data

# Correlation Analysis (Nominal Data)

- **$X^2$ (chi-square) test**

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

- The larger the $X^2$ value, the more likely the variables are related
- Need to use $X^2$ distribution table to find the probability (that two variables are independent from each other)
- Correlation does not imply causality
  - # of hospitals and # of car-theft in a city are correlated
  - Both are causally linked to the third variable: population

# Chi-Square Calculation: An Example

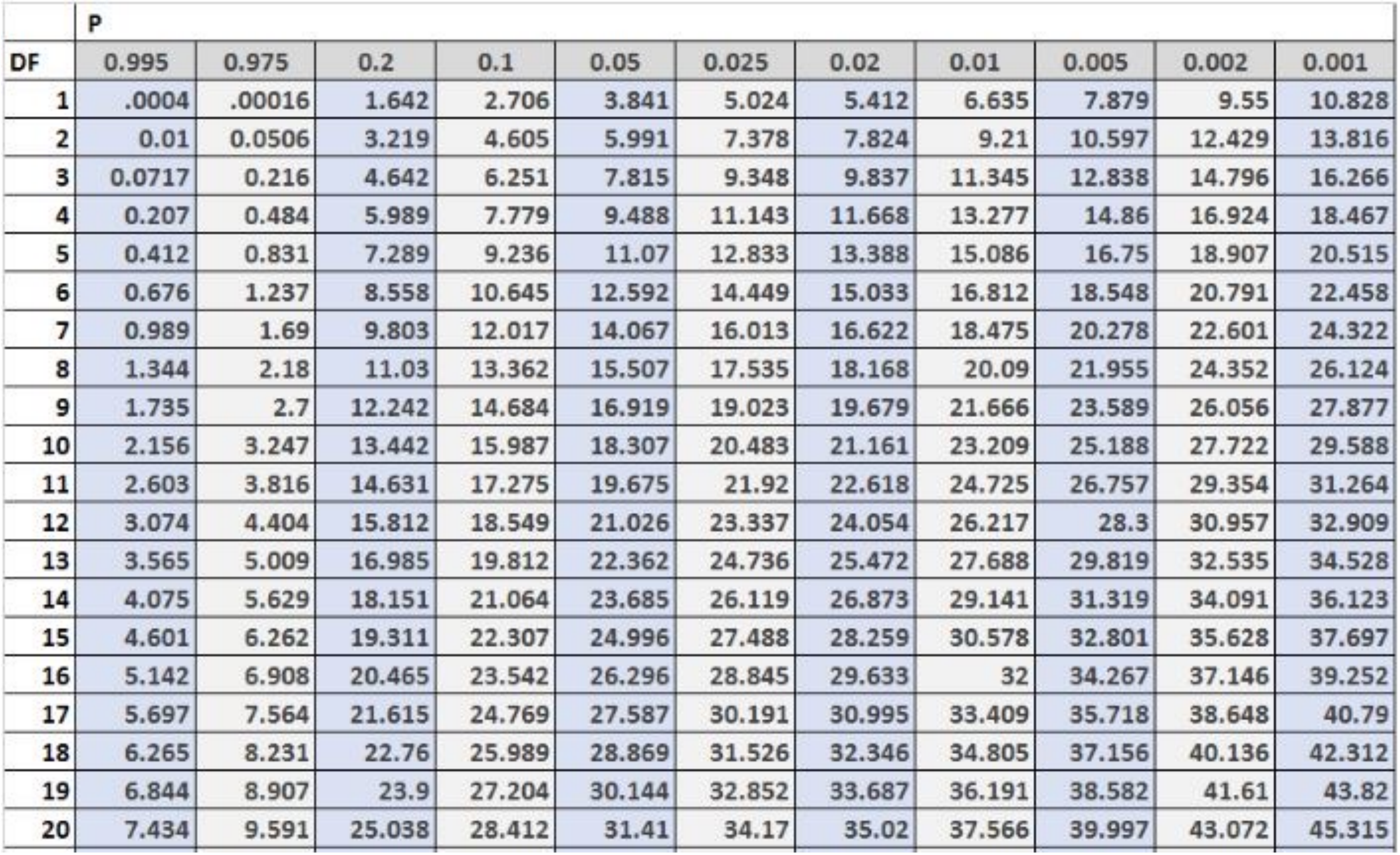
| | Play chess | Not play chess | Sum (row) |
|---|---|---|---|
| Like science fiction | 250(90) | 200(360) | 450 |
| Not like science fiction | 50(210) | 1000(840) | 1050 |
| Sum(col.) | 300 | 1200 | 1500 |

- Expected frequency $= \dfrac{count(A=a_i)*count(B=b_j)}{N}$
- X² (chi-square) calculation (numbers in parenthesis are expected counts calculated based on the data distribution in the two categories)

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840} = 507.93$$

- It shows that like_science_fiction and play_chess are correlated in the group

# $X^2$ Distribution 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 |

# $X^2$ Distribution Table

| | undergraduate | graduate |
|---|---|---|
| Face-to-face | 19 | 6 |
| Online | 10 | 15 |

Answer: 6.65

# Covariance (Numeric Data)

- Covariance is similar to correlation

$$Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}$$

where n is the number of tuples, $\overline{A}$ and $\overline{B}$ are the respective mean or **expected values** of A and B.

- **Positive covariance**: If $Cov_{A,B} > 0$, then A and B tend to move in the same direction.
- **Negative covariance**: If $Cov_{A,B} < 0$ then A and B tend to move in different directon.
- **Independence**: If $Cov_{A,B} = 0$ then A and B are independent.

# Co-Variance: An Example

$$Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}$$

- It can be simplified in computation as

$$Cov(A,B) = E(A\cdot B) - \bar{A}\bar{B}$$

- Suppose two stocks A and B have the following values in one week: (2, 5), (3, 8), (5, 10), (4, 11), (6, 14).
- Question: If the stocks are affected by the same industry trends, will their prices rise or fall together?
  - $\bar{A}$ = (2 + 3 + 5 + 4 + 6)/ 5 = 20/5 = 4
  - $\bar{B}$ = (5 + 8 + 10 + 11 + 14) /5 = 48/5 = 9.6
  - Cov(A,B) = (2×5+3×8+5×10+4×11+6×14)/5 − 4 × 9.6 = 4
- Thus, A and B go in the same direction since Cov(A, B) > 0.

# Correlation Analysis (Numeric Data)

- Correlation coefficient (also called Pearson’s product moment coefficient)

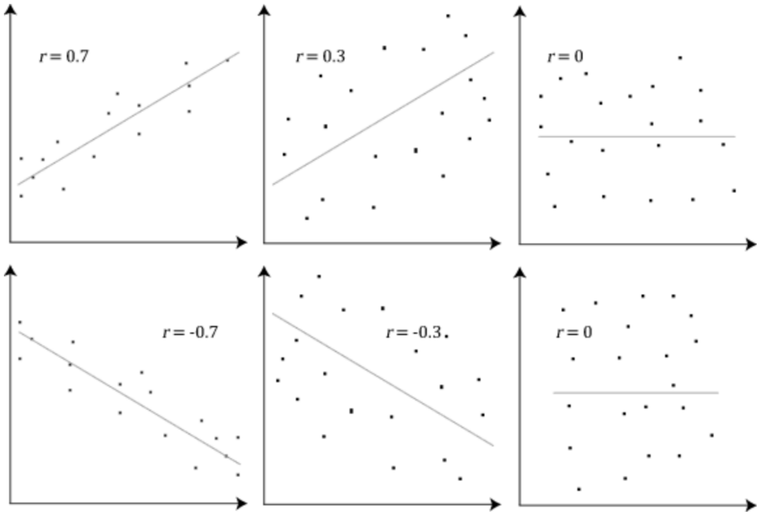
$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y}$$

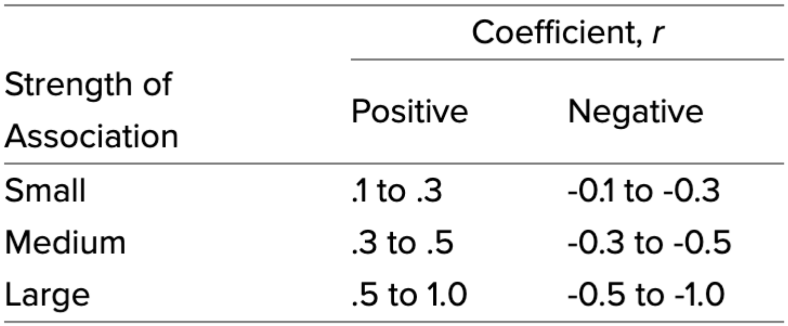
cov(X, Y) is covariance
$\sigma_X$ and $\sigma_Y$ are the respective standard deviation of X and Y

- **What does it tell us?**
  - The strength of a linear correlation between two variables
- **What values can the Pearson correlation coefficient take?**
  - It has a range of values from +1 to -1.
  - A value of 0 indicates that there is no correlation between the two variables.
  - A value greater than 0 indicates a positive correlation.
  - A value less than 0 indicates a negative correlation.

# Correlation Analysis (Numeric Data)

- **How can we determine the strength of correlation based on the Pearson correlation coefficient?**
  - The stronger the correlation of the two variables, the closer the Pearson correlation coefficient will be to either +1 or -1 depending on whether the relationship is positive or negative, respectively.
- **Are there guidelines to interpreting Pearson's correlation coefficient?**





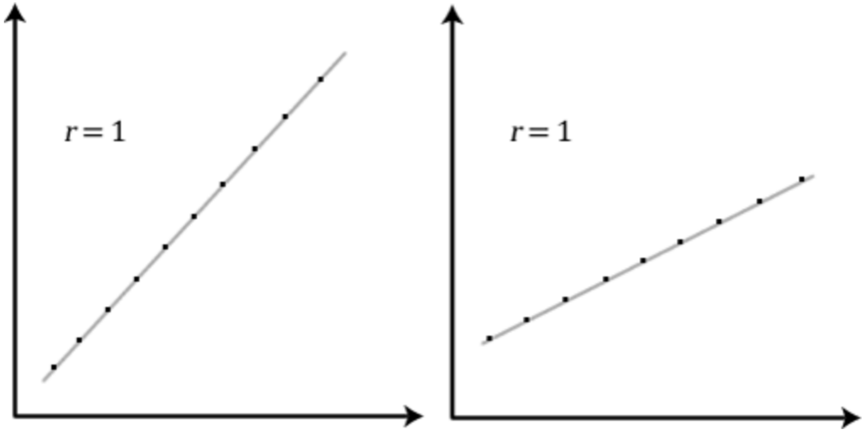
|  | Coefficient, *r* |  |
|---|---|---|
| Strength of Association | Positive | Negative |
| Small | .1 to .3 | -0.1 to -0.3 |
| Medium | .3 to .5 | -0.3 to -0.5 |
| Large | .5 to 1.0 | -0.5 to -1.0 |

# Correlation Analysis (Numeric Data)

- **Can we use any type of variable for Pearson's correlation coefficient?**
  - No, the two variables have to be measured on either an interval or ratio scale. However, both variables do not need to be measured on the same scale.
- **Do the two variables have to be measured in the same units?**
  - No, the two variables can be measured in entirely different units.
  - For example, you could correlate a person's age ( in years) with their weighs in kg.
  - Indeed, the calculations for Pearson's correlation coefficient were designed such that the units of measurement do not affect the calculation. This allows the correlation coefficient to be comparable and not influenced by the units of the variables used.

# Correlation Analysis (Numeric Data)

- **Does the Pearson correlation coefficient indicate the slope of the line?**
  - It is important to realize that the Pearson correlation coefficient, *r*, does not represent the slope of the line of best fit. Therefore, if you get a Pearson correlation coefficient of +1 this does not mean that for every unit increase in one variable there is a unit increase in another. It simply means that there is no variation between the data points and the line of best fit. This is illustrated below:



# Chapter 3: Data Preprocessing

- Data Preprocessing: An Overview
  - Data Quality
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation
- Summary

# Data Reduction Strategies

- **Data reduction**: Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- Why data reduction? — A database/data warehouse may store terabytes of data. Complex data analysis may take a very long time to run on the complete data set.
- Data reduction strategies
  - Dimensionality reduction, e.g., remove unimportant attributes
    - Wavelet transforms
    - Principal Components Analysis (PCA)
    - Feature subset selection
  - Numerosity reduction (some simply call it: Data Reduction)
    - Regression models
    - Histograms, clustering, sampling
    - Data cube aggregation
  - Data compression

# Data Reduction 1: Dimensionality Reduction

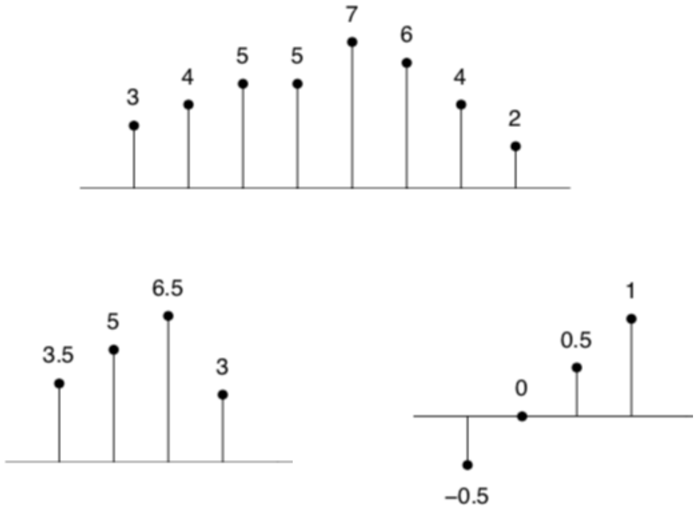
- **Curse of dimensionality**
  - When dimensionality increases, data becomes increasingly sparse
  - Density and distance between points, which is critical to clustering, outlier analysis, becomes less meaningful
  - The possible combinations of subspaces will grow exponentially
- **Dimensionality reduction**
  - Avoid the curse of dimensionality
  - Help eliminate irrelevant features and reduce noise
  - Reduce time and space required in data mining
  - Allow easier visualization
- **Dimensionality reduction techniques**
  - Wavelet transforms
  - Principal Component Analysis
  - Supervised and nonlinear techniques (e.g., feature selection)

# What is Wavelet Transform?

- Decomposes a signal into different frequency subbands
- Data are transformed to preserve relative distance between objects at different levels of resolution
- Used for image compression
- Method - Haar
  - Length, L, must be an integer power of 2 (padding with 0’s, when necessary)
  - Each transform has 2 functions: smoothing, difference
  - Applies to pairs of data, resulting in two set of data of length L/2
  - Applies two functions recursively, until reaches the desired length

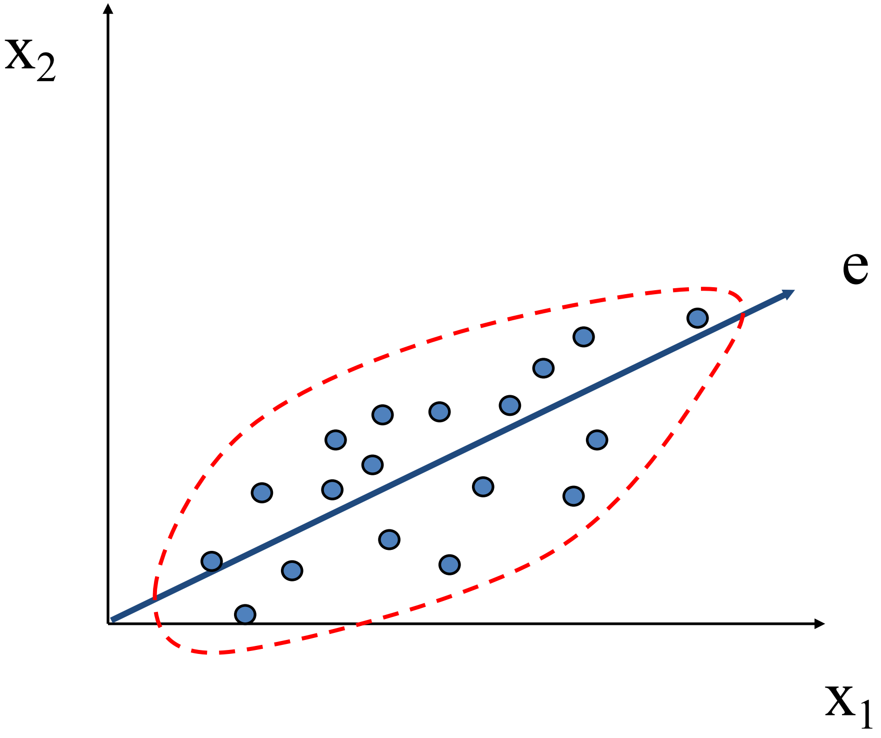
# Haar: An Example

- The basis of the Haar transform is the decomposition of a signal, say the eight-point signal x(n) into two four-point signals.
  - One being the average of pairs of signal values, c(n), i.e., smoothing
  - The other signal being their differences, d(n), i.e., difference
- Repeat this process, until c(n) and d(n) are in one dimension
  - Try it yourself!
- Is it a lossless or lossy transformation?



# Principal Component Analysis (PCA)

- Find a projection that captures the largest amount of variation in data
- The original data are projected onto a much smaller space, resulting in dimensionality reduction. We find the eigenvectors of the covariance matrix, and these eigenvectors define the new space



# PCA Example –STEP 1

- Subtract the mean

from each of the data dimensions. All the x values have $\overline{x}$ subtracted and y values have $\overline{y}$ subtracted from them. This produces a data set whose mean is zero.

Subtracting the mean makes variance and covariance calculation easier by simplifying their equations.

# PCA Example –STEP 1

DATA:

| x | y |
|---|---|
| 2.5 | 2.4 |
| 0.5 | 0.7 |
| 2.2 | 2.9 |
| 1.9 | 2.2 |
| 3.1 | 3.0 |
| 2.3 | 2.7 |
| 2 | 1.6 |
| 1 | 1.1 |
| 1.5 | 1.6 |
| 1.1 | 0.9 |

ZERO MEAN DATA:

| x | y |
|---|---|
| .69 | .49 |
| -1.31 | -1.21 |
| .39 | .99 |
| .09 | .29 |
| 1.29 | 1.09 |
| .49 | .79 |
| .19 | -.31 |
| -.81 | -.81 |
| -.31 | -.31 |
| -.71 | -1.01 |

# PCA Example –STEP 2

- Calculate the covariance matrix

$$\text{cov} = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

- since the non-diagonal elements in this covariance matrix are positive, we should expect that both the x and y variables move in the same direction.

# PCA Example –STEP 3

- Calculate the eigenvectors and eigenvalues of the covariance matrix

$$\text{eigenvalues} = \begin{pmatrix} 0.0490833989 \\ 1.28402771 \end{pmatrix}$$

$$\text{eigenvectors} = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

$$\begin{array}{cc} \uparrow & \uparrow \\ \text{Corresponds to eigenvalue 0.0490833989} & \text{Corresponds to eigenvalue 1.28402771} \end{array}$$

# PCA Example –STEP 3

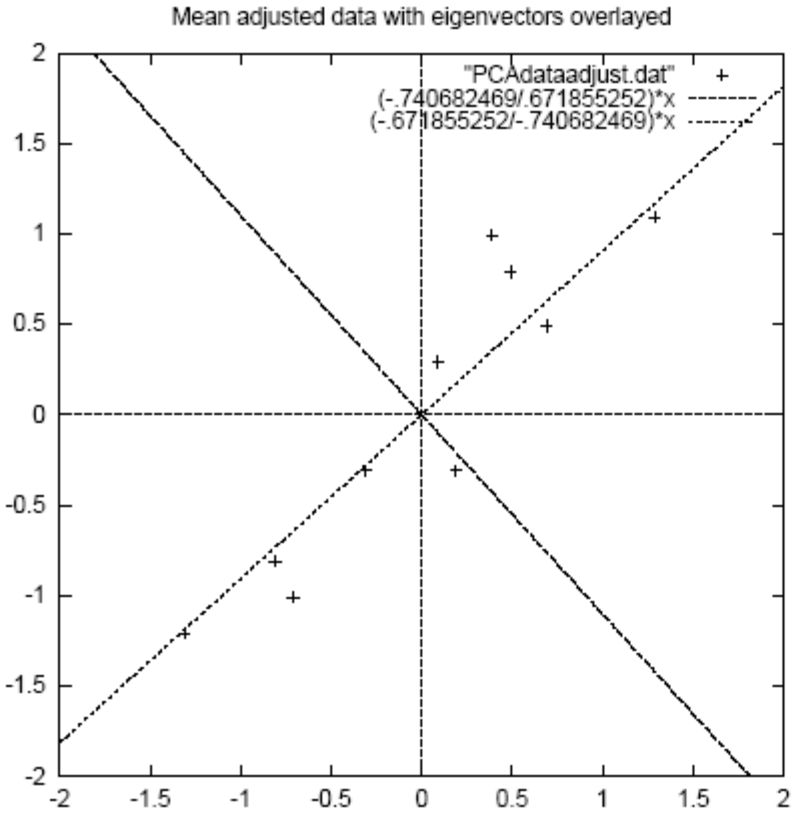


Figure 3.2: A plot of the normalised data (mean subtracted) with the eigenvectors of the covariance matrix overlayed on top.

- Eigenvectors are plotted as diagonal dotted lines on the plot.
- Note they are perpendicular to each other.
- Note one of the eigenvectors goes through the middle of the points, like drawing a line of best fit.
- The second eigenvector gives us the other, less important, pattern in the data, that all the points follow the main line, but are off to the side of the main line by some amount.

# PCA Example –STEP 4

- Reduce dimensionality and form *feature vector*

Once eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest. This gives you the components in order of significance.

The eigenvector with the *highest* eigenvalue is the *principle component* of the data set.

In our example, the eigenvector with the largest eigenvalue was the one that pointed down the middle of the data.

Now, you can decide to *ignore* the components of less significance.

# PCA Example –STEP 4

- Ordered eigenvalues from largest to smallest

  (eig$_1$ eig$_2$ eig$_3$ … eig$_n$)

  = (1.28402771, 0.0490833989) in our example

- Feature vectors: eigenvectors reordered accordingly

$$\begin{pmatrix} -.677873399 & -.735178656 \\ -.735178656 & .677873399 \end{pmatrix}$$

- We can choose to leave out the smaller, less significant component and only have a single column/eigenvector:

$$\begin{pmatrix} -\,.677873399 \\ -\,.735178656 \end{pmatrix}$$

# PCA Example –STEP 5

- Project on the new feature space to derive new data

**FinalData = RowFeatureVector x RowZeroMeanData**

RowFeatureVector is the matrix with the eigenvectors in the columns *transposed* so that the eigenvectors are now in the rows, with the most significant eigenvector at the top

RowZeroMeanData is the mean-adjusted data *transposed*, ie. the data items are in each column, with each row holding a separate dimension.

# PCA Example –STEP 5

FinalData transpose: dimensions along columns

| x | y |
|---|---|
| -.827970186 | -.175115307 |
| 1.77758033 | .142857227 |
| -.992197494 | .384374989 |
| -.274210416 | .130417207 |
| -1.67580142 | -.209498461 |
| -.912949103 | .175282444 |
| .0991094375 | -.349824698 |
| 1.14457216 | .0464172582 |
| .438046137 | .0177646297 |
| 1.22382056 | -.162675287 |

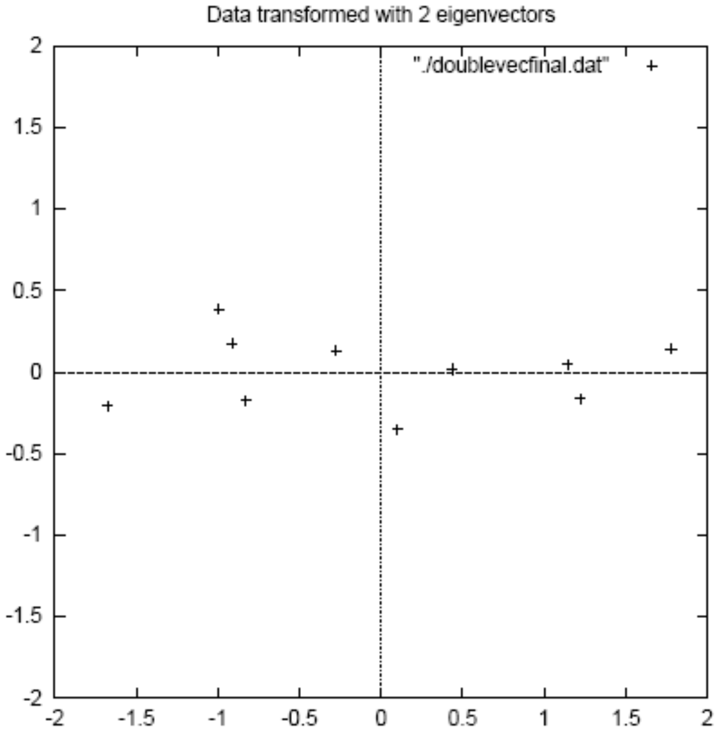


Figure 3.3: The table of data by applying the PCA analysis using both eigenvectors, and a plot of the new data points.

In this example, both feature vectors were kept, but data were projected into this new feature space.

# Attribute Subset Selection

- Another way to reduce dimensionality of data
- Redundant attributes
  - Duplicate much or all of the information contained in one or more other attributes
  - E.g., purchase price of a product and the amount of sales tax paid
- Irrelevant attributes
  - Contain no information that is useful for the data mining task at hand
  - E.g., students' ID is often irrelevant to the task of predicting students' GPA

# Heuristic Search in Attribute Selection

- There are $2^p$ possible attribute combinations of $p$ attributes
- Typical heuristic attribute selection methods:
  - Best single attribute under the attribute independence assumption: choose by significance tests
  - Best step-wise feature selection:
    - The best single-attribute is picked first
    - Then next best attribute, ...
  - Step-wise attribute elimination:
    - Repeatedly eliminate the worst attribute
  - Combined attribute selection and elimination

# Data Reduction 2: Numerosity Reduction

- Reduce data volume by choosing alternative, *smaller forms* of data representation
- **Parametric methods**
  - **Assume the data fits some model**, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
  - Linear regression
    - Data modeled to fit a straight line
    - Often uses the least-square method to fit the line
- **Non-parametric** methods
  - **Do not assume models**
  - Major method: sampling

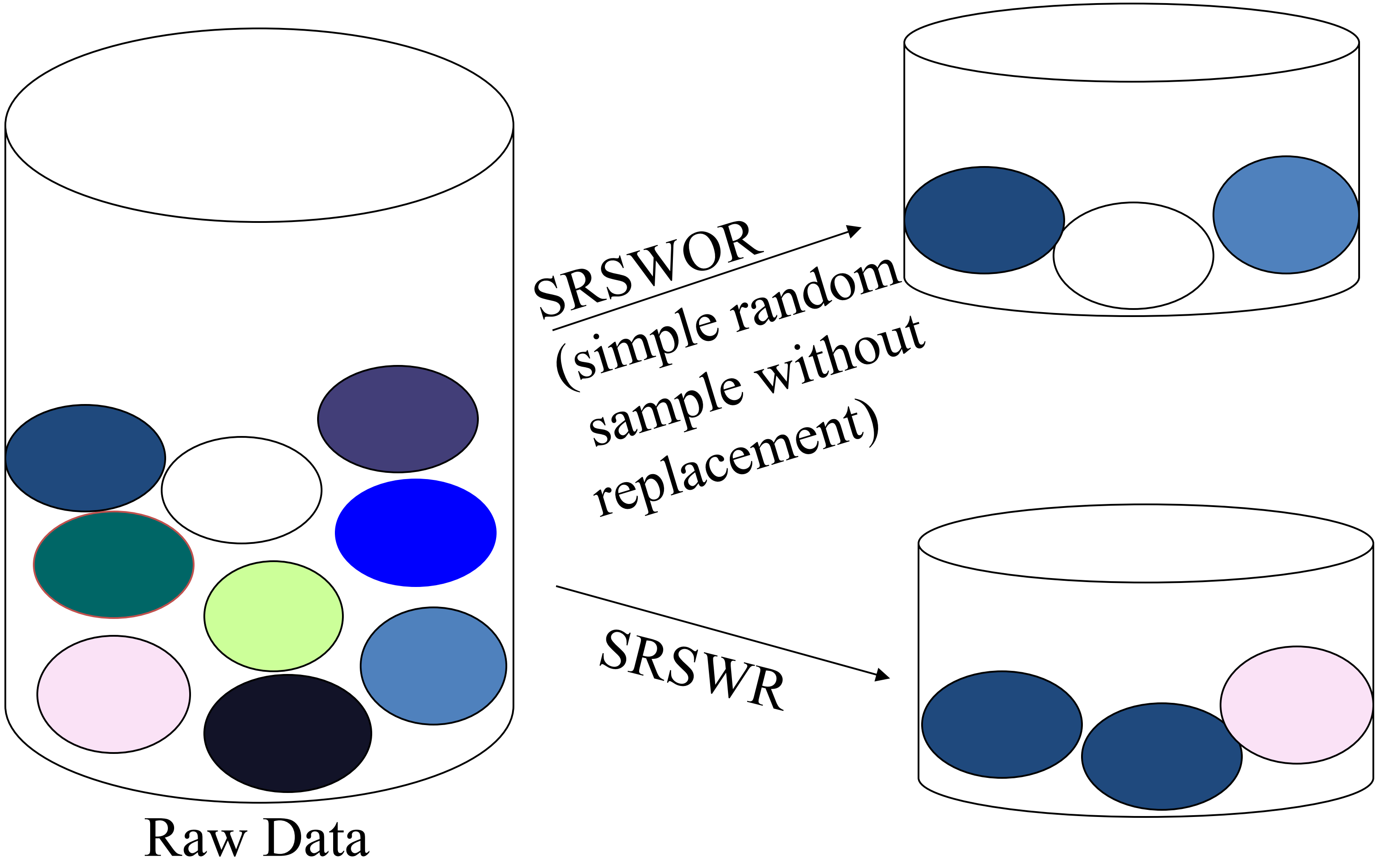
# Non-parametric Data Reduction: Sampling

- Sampling: obtaining a small sample *s* to represent the whole data set *N*
- Key principle: Choose a representative subset of the data
  - Simple random sampling may have very poor performance in the presence of skew
  - Develop adaptive sampling methods, e.g., stratified sampling

# Types of Sampling

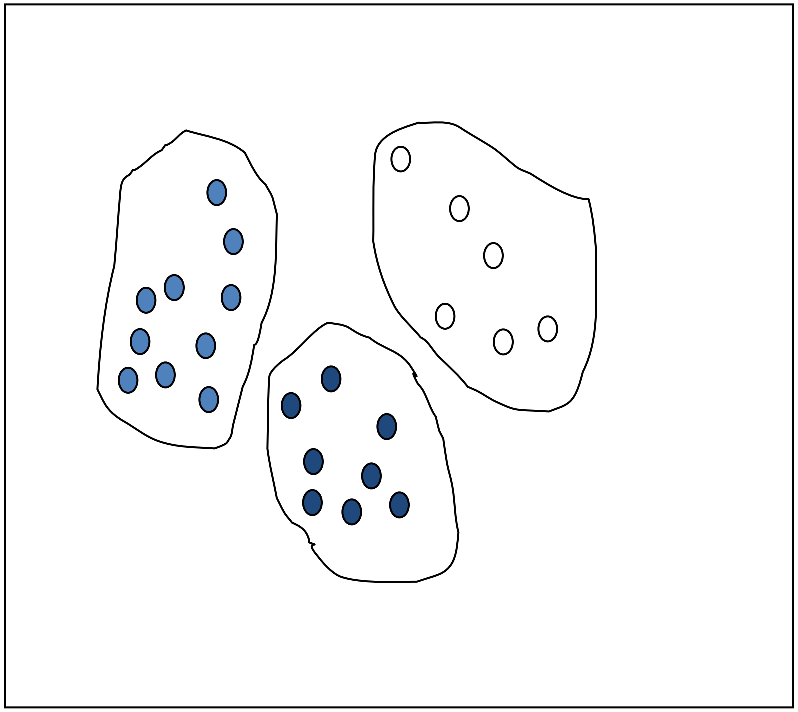
- Simple random sampling (SRS)
  - There is an equal probability of selecting any particular item
  - Sampling without replacement (SRSWOR)
    - Once an object is selected, it is removed from the population
  - Sampling with replacement (SRSWR)
    - A selected object is not removed from the population
- Stratified sampling:
  - Partition the data set, and draw samples from each partition (proportionally, i.e., approximately the same percentage of the data)

# Sampling: with or without Replacement

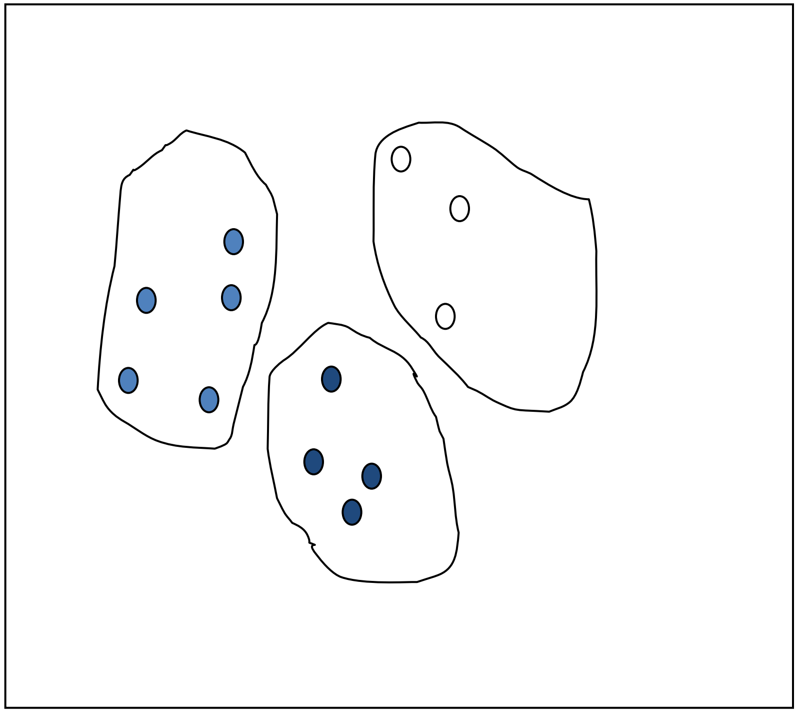


# Sampling: Stratified Sampling

Raw Data



Stratified Sample



# Chapter 3: Data Preprocessing

- Data Preprocessing: An Overview
  - Data Quality
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation
- Summary

# Data Transformation

- A function that maps the entire set of values of a given attribute to a new set of replacement values
- Methods
  - Normalization: Scaled to fall within a smaller (specified) range
  - Discretization: Divide the range of a continuous attribute into intervals
  - Concept Hierarchy Generation: Organizes concepts (i.e., attribute values) hierarchically

# Normalization

- **Min-max normalization**: to [new_min$_A$, new_max$_A$]

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

  - Ex. Let income range \$12,000 to \$98,000 normalized to [0.0, 1.0]. Then \$73,600 is mapped to $\frac{73{,}600 - 12{,}000}{98{,}000 - 12{,}000}(1.0 - 0) + 0 = 0.716$

- **Z-score normalization** (μ: mean, σ: standard deviation):

$$v' = \frac{v - \mu_A}{\sigma_A}$$

  - Ex. Let μ = 54,000, σ = 16,000. Then $\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$

- **Normalization by decimal scaling**

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

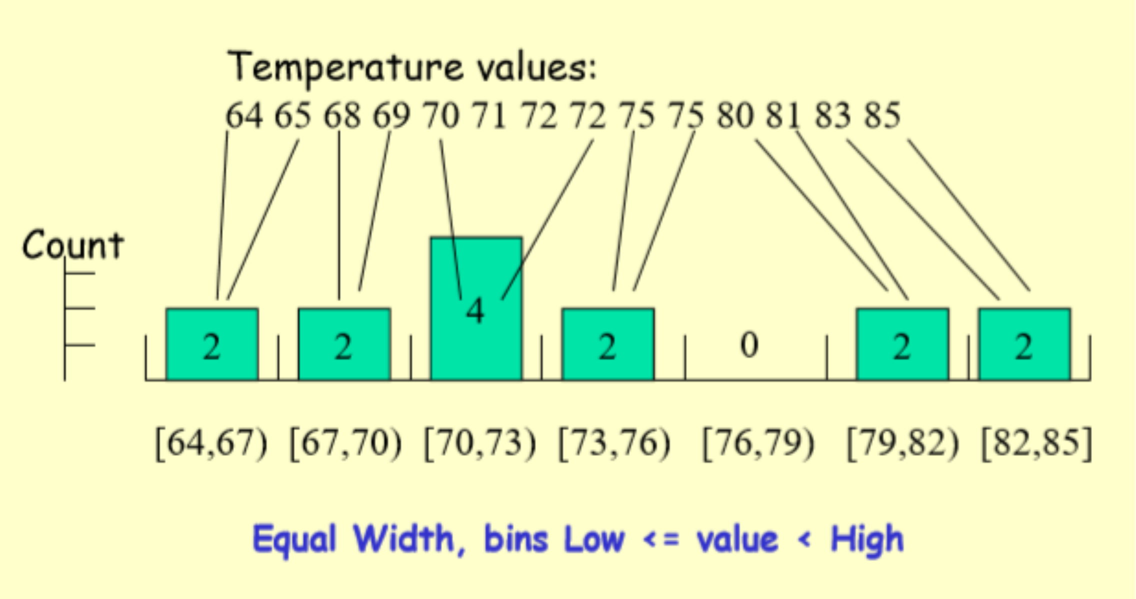
Where *j* is the smallest integer such that *max*(|v' |) <= 1

# Discretization

- Discretization: Divide the range of a continuous attribute into intervals
  - Interval labels can then be used to replace actual data values
  - Discretization can be performed recursively on an attribute
  - Reduce data size by discretization
  - Some methods require discrete values, e.g. Naïve Bayes
- Methods
  - Unsupervised (class-independent): binning
  - Supervised (class-dependent): decision tree

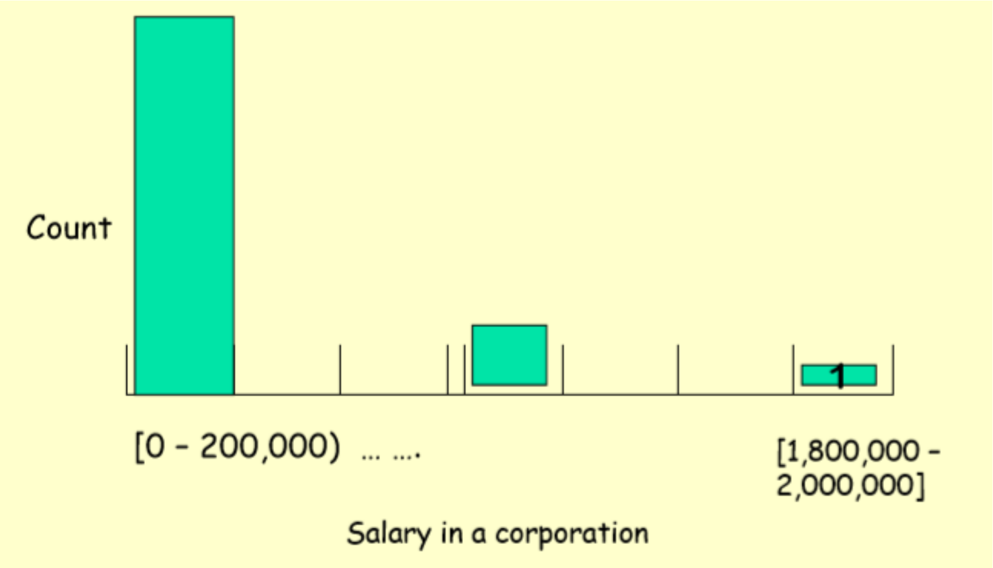
# Unsupervised Discretization: Binning

- Equal-width (distance) partitioning/binning
  - 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: $W = (B - A)/N$.
  - Most straightforward approach



# Unsupervised Discretization: Binning

- Equal-width (distance) partitioning/bining
  - Disadvantage
    - Where does N come from?
    - Sensitive to outliers
    - Skewed data is not handled well



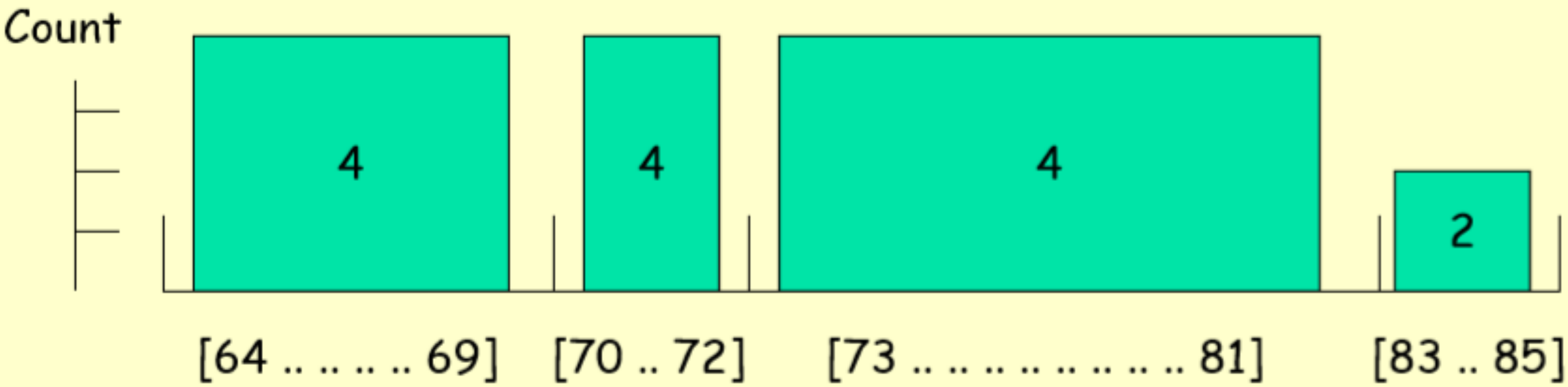
# Unsupervised Discretization: Binning

- Equal-depth (height or frequency) partitioning/binning
  - Divides the range into *N* intervals, each containing approximately same number of samples
  - In practice, "almost-equal" height binning is used to give more intuitive breakpoints
  - Some considerations
    - don't split frequent values across bins
    - create separate bins for special values (e.g. 0)
    - readable breakpoints (e.g. round breakpoints)

# Unsupervised Discretization: Binning

- Equal-depth (height or frequency) partitioning/binning

Temperature values:
64 65 68 69 70 71 72 72 75 75 80 81 83 85



Equal Height = 4, except for the last bin

# Binning Exercise

- Discretize the following values using EW and ED binning (#bin = 4)
- 13, 15, 16, 16, 19, 20, 21, 22, 22, 25, 30, 33, 35, 35, 36, 40, 45

# Concept Hierarchy Generation

- **Concept hierarchy** organizes concepts (i.e., attribute values) hierarchically and is usually associated with each dimension in a data warehouse
- Concept hierarchies facilitate drilling and rolling in data warehouses to view data in multiple granularity
- Concept hierarchy formation: Recursively reduce the data by collecting and replacing low level concepts by higher level concepts
  - Specification of a partial/total ordering of attributes explicitly at the schema level by users or experts, e.g., *street* < *city* < *state* < *country*
  - Specification of a hierarchy for a set of values by explicit data grouping, e.g., Dairy Product = {milk, cheese, yogout}
- Concept hierarchies may be explicitly specified by domain experts

# Chapter 3: Data Preprocessing

- Data Preprocessing: An Overview
  - Data Quality
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation
- Summary

# Summary

- **Data quality**: accuracy, completeness, consistency, timeliness, believability, interpretability
- **Data cleaning**: missing/noisy values, outliers
- **Data integration** from multiple sources:
  - Detect inconsistencies
  - Remove redundancies
- **Data reduction**
  - Dimensionality reduction
  - Numerosity reduction
- **Data transformation**
  - Normalization
  - Discretization
  - Concept hierarchy generation