NATIONAL ENGINEERING CENTER
University of the Philippines
Diliman, Quezon City

2.0 Data Preprocessing

Eugene Rex L. Jalao, Ph.D.

Associate Professor

Department Industrial Engineering and Operations Research

University of the Philippines Diliman

@thephdataminer

Module 3 of the Business Intelligence and Analytics Certification of UP NEC and the UP Center for Business Intelligence

Outline for This Training

1. Introduction to Data Mining
2. **Data Preprocessing**
   - **Case Study on Big Data Preprocessing using R**
3. Classification Methodologies
   - Case Study on Classification using R
4. Regression Methodologies
   - Case Study: Regression Analysis using R
5. Unsupervised Learning
   - Case Study: Social Media Sentiment Analysis using R

Outline for this Session

- **Introduction to Data Preprocessing**
- Data Integration
- Data Transformations
  - Data Discretization
  - Data Encoding
- Data Cleaning
- Data Reduction
- Case Study: Data Preprocessing Using R

Why Data Preprocessing?

- Data in the real world is dirty
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., occupation=“ ”
  - noisy: containing errors or outliers
    - e.g., Salary=“-10”
  - inconsistent: containing discrepancies in codes or names
    - e.g., Age=“42” Birthday=“03/07/1997”
    - e.g., Was rating “1,2,3”, now rating “A, B, C”
    - e.g., discrepancy between duplicate records

Why Is Data Dirty?

- Incomplete data may come from
  - “Not applicable” data value when collected
  - Different considerations between the time when the data was collected and when it is analyzed.
  - Human/hardware/software problems
- Noisy data (incorrect values) may come from
  - Faulty data collection instruments
  - Human or computer error at data entry
  - Errors in data transmission
- Inconsistent data may come from
  - Different data sources
  - Functional dependency violation (e.g., modify some linked data)
- Duplicate records also need data cleaning

Why Is Data Preprocessing Important?

- No quality data, no quality mining results!
  - Quality decisions must be based on quality data
    - e.g., duplicate or missing data may cause incorrect or even misleading statistics.
  - Data warehouse needs consistent integration of quality data
- Data extraction, cleaning, and transformation comprises the majority of the work of building a data warehouse

Major Tasks in Data Preprocessing

- Data integration
  - Integration of multiple databases, data cubes, or files
- Data transformation
  - Normalization and Aggregation
  - Encoding and Binning
- Data cleaning
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- Data reduction
  - Obtains reduced representation in volume but produces the same or similar analytical results

Outline for this Session

- Introduction to Data Preprocessing
- **Data Integration**
- Data Transformations
  - Data Discretization
  - Data Encoding
- Data Cleaning
- Data Reduction
- Case Study: Data Preprocessing Using R

Data Integration

- Data integration:
  - Combines data from multiple sources into a coherent store
- Schema integration: e.g., A.cust-id $\equiv$ B.cust-#
  - Integrate metadata from different sources
- Entity identification problem:
  - Identify real world entities from multiple data sources, e.g., Bill Clinton = William Clinton
- Detecting and resolving data value conflicts
  - For the same real world entity, attribute values from different sources are different
  - Possible reasons: different representations, different scales, e.g., metric vs. British units

Handling Redundancy in Data Integration

- Redundant data occur often when integration of multiple databases
  - Object identification: The same attribute or object may have different names in different databases
  - Derivable data: One attribute may be a "derived" attribute in another table, e.g., annual revenue
- Redundant attributes may be able to be detected by correlation analysis
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

Data Integration Types

heisenberg

trial	mass	velocity
A	10.0	12
A	11.0	14
B	5.0	8
B	6.0	10
A	10.5	13
B	7.0	11

trialcost

trial	cost
A	11.4
C	3.3
D	1.1

merge →

Inner Join

trial	mass	velocity	cost
A	10.0	12	11.4
A	11.0	14	11.4
A	10.5	13	11.4

Outer Join

trial	mass	velocity	cost
A	10.0	12	11.4
A	11.0	14	11.4
A	10.5	13	11.4
B	5.0	8	NA
B	6.0	10	NA
B	7.0	11	NA
C	NA	NA	3.3
D	NA	NA	1.1

Left Join

trial	mass	velocity	cost
A	10.0	12	11.4
A	11.0	14	11.4
A	10.5	13	11.4
B	5.0	8	NA
B	6.0	10	NA
B	7.0	11	NA

Right Join

trial	mass	velocity	cost
A	10.0	12	11.4
A	11.0	14	11.4
A	10.5	13	11.4
C	NA	NA	3.3
D	NA	NA	1.1

E.

Example

- `#read a CSV file`
- `heisenberg <- read.csv(file="simple.csv",head=TRUE,sep=",")`
- `trial <- c("A","C","D")`
- `cost <- c(11.4, 3.3, 1.1)`
- `trialcost <- data.frame(trial,cost)`

Example

- `#merge`
- `innerjoin = merge(x=heisenberg,y=trialcost,by=c("trial"))`
- `outerjoin = merge(x=heisenberg,y=trialcost,by=c("trial"), all= TRUE)`
- `leftjoin = merge(x=heisenberg,y=trialcost,by=c("trial"), all.x=TRUE)`
- `rightjoin = merge(x=heisenberg,y=trialcost,by=c("trial"), all.y=TRUE)`

Outline for this Session

- Introduction to Data Preprocessing
- Data Integration
- **Data Transformations**
  - **Data Discretization**
  - **Data Encoding**
- Data Cleaning
- Data Reduction
- Case Study: Data Preprocessing Using R

Data Transformation

- The process of transforming data from one format to another
- Some Transformations
  - Normalization: scaled to fall within a small, specified range
    - min-max normalization
    - z-score standardization
  - Encoding and Binning
  - Smoothing: remove noise from data
  - Aggregation: summarization, data cube construction
  - Attribute/feature construction
    - New attributes constructed from the given ones

Data Transformation: Normalization

- Min-max normalization :

$$v' = \frac{v - min}{max - min}(newmax - newmin) + newmin$$

  – Ex. Let income range PhP 12,000 to PhP 98,000 normalized to [0.0, 1.0]. Then PhP 73,000 is mapped to $\frac{73{,}000 - 12{,}000}{98{,}000 - 12{,}000}(1.0 - 0) + 0 = 0.709$

Old	New
PhP 73,000.00	0.709
PhP 80,000.00	0.791
PhP 14,000.00	0.023
PhP 58,000.00	0.535

Data Transformation: Standardization

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

$$v' = \frac{v - \mu}{\sigma}$$

  - Ex. Let $\mu = 54{,}000, \sigma = 16{,}000$. Then $\frac{73{,}000 - 54{,}000}{16{,}000} = 1.188$

Old	New
PhP 73,000.00	1.188
PhP 80,000.00	1.625
PhP 14,000.00	-2.500
PhP 58,000.00	0.250

eljalao@up.edu.ph

Example

- Transform the Heisenberg Data Mass attribute into a Scale of 1-5. 1 being the lowest and 5 being the highest.

trial	mass	velocity
A	10.0	12
A	11.0	14
B	5.0	8
B	6.0	10
A	10.5	13
B	7.0	11

eljalao@up.edu.ph

Example

- heisenberg <- read.csv(file="simple.csv",head=TRUE,sep=",")
- heisenberg$smass = (heisenberg$mass-min(heisenberg$mass))/(max(heisenberg$mass)-min(heisenberg$mass))*(5-1)+1
- heisenberg

```
> heisenberg
  trial mass velocity    smass
1     A 10.0       12 4.333333
2     A 11.0       14 5.000000
3     B  5.0        8 1.000000
4     B  6.0       10 1.666667
5     A 10.5       13 4.666667
6     B  7.0       11 2.333333
```

Example

- `heisenberg$svelocity = (heisenberg$velocity-mean(heisenberg$velocity))/sd(heisenberg$velocity)`
- `heisenberg`

```
> heisenberg
  trial mass velocity    smass  svelocity
1     A 10.0       12 4.333333  0.3086067
2     A 11.0       14 5.000000  1.2344268
3     B  5.0        8 1.000000 -1.5430335
4     B  6.0       10 1.666667 -0.6172134
5     A 10.5       13 4.666667  0.7715167
6     B  7.0       11 2.333333 -0.1543033
```

Binning

- Binning: Process of transforming numerical variables into categorical counterparts.
  - An example is to bin values for Age into categories such as 1 to 18, 18 to 49, and 49 onwards.
- Rationale: Some Data Mining Algorithms run better on Categorical Data: e.g. Decision Trees
- Allows easy identification of outliers
- Two Types
  - Equal Width
  - Equal Depth

Binning

- Equal-width (distance) partitioning
  - 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: $W = (B - A)/N$.
  - The most straightforward, but outliers may dominate presentation
  - Skewed data is not handled well
- Equal-depth (frequency) partitioning
  - Divides the range into N intervals, each containing approximately same number of samples
  - Good data scaling and outliers

Binning

- Example: Equal Width Binning
  - Raw data for price (in PhP): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
  - Partition into 3 Bins: $W = \frac{B-A}{N} = \frac{34-4}{3} = 10$
  - 3 Bins: [4,14), [14,24),[24,34]
    - Partition into equal-width bins:
      - Bin 1: 4, 8, 9
      - Bin 2: 15, 21, 21
      - Bin 3: 24, 25, 26, 28, 29, 34
  - Smoothed data for price (in PhP): 7, 7, 7, 19, 19, 19, 27.6, 27.6, 27.6, 27.6, 27.6, 27.6

Binning

- Example: Equal Depth Binning
  - Sorted data for price (in PhP): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
    - Partition into equal-frequency (equi-depth) bins:
      - Bin 1: 4, 8, 9, 15
      - Bin 2: 21, 21, 24, 25
      - Bin 3: 26, 28, 29, 34
    - Smoothing by bin means:
      - Bin 1: 9, 9, 9, 9
      - Bin 2: 23, 23, 23, 23
      - Bin 3: 29, 29, 29, 29
  - Sorted data for price (in PhP): 9, 9, 9, 9, 23, 23, 23, 23, 29, 29, 29, 29

Example

- Discretize the Heisenberg Mass Dataset into three Bins using Equal Width Binning

trial	mass	velocity
A	10.0	12
A	11.0	14
B	5.0	8
B	6.0	10
A	10.5	13
B	7.0	11

- Discretize the Heisenberg Velocity Dataset into three Bins using Equal Depth Binning

R Example

- `heisenberg <- read.csv(file="simple.csv",head=TRUE,sep=",")`
- `bins=3`
- `cutpoints=quantile(heisenberg$mass,(0:bins)/bins)`
- `heisenberg$discretemass =cut(heisenberg$mass, cutpoints, include.lowest=TRUE, labels=c("Low", "Med", "High"))`
- `heisenberg`

```
> heisenberg
  trial mass velocity discretemass
1     A 10.0       12       Medium
2     A 11.0       14         High
3     B  5.0        8          Low
4     B  6.0       10          Low
5     A 10.5       13         High
6     B  7.0       11       Medium
```

R Example

- `heisenberg$discretevelocity =cut(heisenberg$velocity,  bins, include.lowest=TRUE, labels=c("Low","Medium",  "High"))`
- `heisenbergheisenberg`

```
> heisenberg
  trial mass velocity discretemass discretevelocity
1     A 10.0       12       Medium           Medium
2     A 11.0       14         High             High
3     B  5.0        8          Low              Low
4     B  6.0       10          Low              Low
5     A 10.5       13         High             High
6     B  7.0       11       Medium           Medium
```

Data Encoding

- Encoding or continuation is the transformation of categorical variables to binary or numerical counterparts.
  - An example is to treat male or female for gender as 1 or 0.
- Some Data Mining Methodologies require all data to be numerical, e.g. Linear Regression
- Two Types
  - Binary Encoding (Unsupervised)
  - Class-based Encoding (Supervised)

Binary Encoding

- Transformation of categorical variables by taking the values 0 or 1 to indicate the absence or presence of each category.
- If the categorical variable
  has *k* categories we would need to create *k* binary variables

Trend	Trend_Up	Trend_Down	Trend_Flat
Up	1	0	0
Up	1	0	0
Down	0	1	0
Flat	0	0	1
Down	0	1	0
Up	1	0	0
Down	0	1	0
Flat	0	0	1
Flat	0	0	1
Flat	0	0	1

Class-Based Encoding: Discrete Class

- Replace the categorical variable with just one new numerical variable and replace each category of the categorical variable with its corresponding probability of the class variable

Trend	Class=No	Class=Yes	Probability (Yes)
Up	1	2	0.66
Down	2	1	0.33
Flat	2	2	0.5

Trend	Class	Trend_Encoded
Up	Yes	0.66
Up	Yes	0.66
Down	No	0.66
Flat	No	0.5
Down	Yes	0.33
Up	No	0.33
Down	No	0.66
Flat	No	0.5
Flat	Yes	0.5
Flat	Yes	0.5

Class-Based Encoding: Continuous Class

- Replace the categorical variable with just one new numerical variable and replace each category of the categorical variable with its corresponding average of the class variable

Trend	Class	Trend_Encoded
Up	21	23.7
Up	24	23.7
Down	8	10.3
Flat	15	14.5
Down	11	10.3
Up	26	23.7
Down	12	10.3
Flat	16	14.5
Flat	14	14.5
Flat	13	14.5

Trend	Class - Average
Up	23.7
Down	10.3
Flat	14.5

Example

- Encode the Trial Field using Binary Encoding and Continuous Encoding

trial	mass	velocity
A	10.0	12
A	11.0	14
B	5.0	8
B	6.0	10
A	10.5	13
B	7.0	11

Example

- heisenberg <- read.csv(file="simple.csv",head=TRUE,sep=",")
- indicators=model.matrix(~ trial-1, data = heisenberg)
- heisenberg = cbind(heisenberg,indicators)
- heisenberg

```
> heisenberg
  trial mass velocity trialA trialB
1     A 10.0       12      1      0
2     A 11.0       14      1      0
3     B  5.0        8      0      1
4     B  6.0       10      0      1
5     A 10.5       13      1      0
6     B  7.0       11      0      1
```

Example

- `library("reshape2")`
- `heisenberg.m = melt(heisenberg, id=c('trial'), measure=c('mass'))`
- `heisenberg <- read.csv(file="simple.csv",head=TRUE,sep=",")`
- `heisenberg.c = dcast(heisenberg.m, trial ~ variable, mean)`
- `names(heisenberg.c)[names(heisenberg.c) == 'mass'] <- 'trialencoded’`
- `heisenberg = merge(x=heisenberg,y=heisenberg.c,by=c("trial"))`

Example

```
> heisenberg
  trial mass velocity trialencoded
1     A 10.0       12         10.5
2     A 11.0       14         10.5
3     A 10.5       13         10.5
4     B  5.0        8          6.0
5     B  6.0       10          6.0
6     B  7.0       11          6.0
```

Outline for this Session

- Introduction to Data Preprocessing
- Data Integration
- Data Transformations
  - Data Discretization
  - Data Encoding
- **Data Cleaning**
- Data Reduction
- Case Study: Data Preprocessing Using R

Data Cleaning

- Importance
  - "Data cleaning is one of the three biggest problems in data warehousing"—Ralph Kimball
- Data cleaning tasks
  - Fill in missing values
  - Identify outliers and smooth out noisy data
  - Correct inconsistent data
  - Resolve redundancy caused by data integration

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
  - inconsistent with other recorded data and thus deleted
  - data not entered due to misunderstanding
  - data may not be considered important at the time of entry
  - not register history or changes of the data
- Missing data may need to be inferred.

How to Handle Missing Data?

- Ignore the Row: not effective if there are lots of missing data
- Fill in the missing value manually: tedious + infeasible?
- Data Imputation
  - Fill in it automatically with
    - a global constant : e.g., “unknown”, a new class?
    - the attribute mean
    - the attribute mean for all samples belonging to the same class: smarter

Data Cleaning

- Tax Income
- Avg = 93.6 K

- Yes Tax Income
- Avg = 87.5 K

- No Tax Income
- Avg = 96 K

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	?	**No**
4	Yes	Married	120K	**No**
5	?	Divorced	?	**Yes**
6	?	Married	60K	**No**
7	Yes	Divorced	?	**No**
8	No	Single	85K	**Yes**
9	?	Married	75K	**No**
10	No	Single	90K	**Yes**

Noisy Data

- Noise: random error or variance in a measured variable
- Incorrect attribute values may due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention
- Other data problems which requires data cleaning
  - duplicate records
  - incomplete data
  - inconsistent data

eljalao@up.edu.ph

How to Handle Noisy Data?

- Binning
  - first sort data and partition into (equal-frequency) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Regression
  - smooth by fitting the data into regression functions
- Clustering
  - detect and remove outliers
- Combined computer and human inspection
  - detect suspicious values and check by human (e.g., deal with possible outliers)

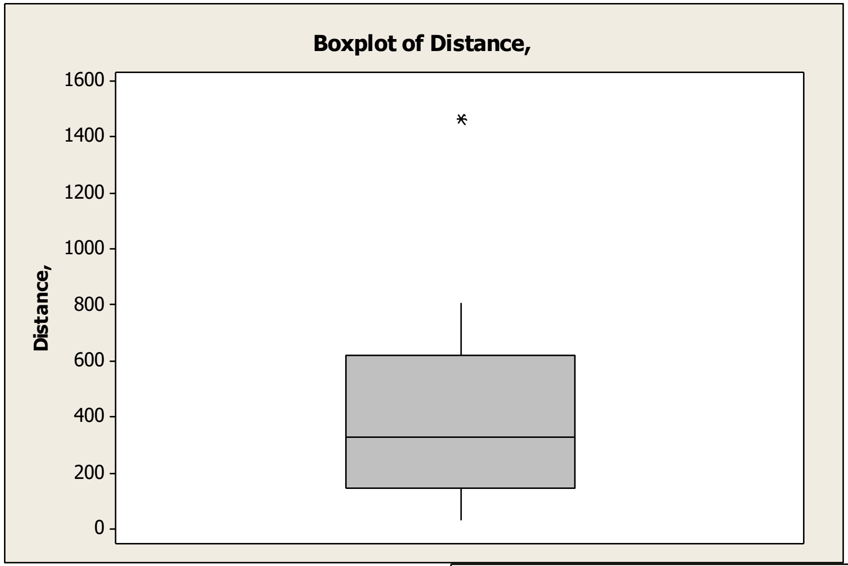
Binning Methods for Data Smoothing

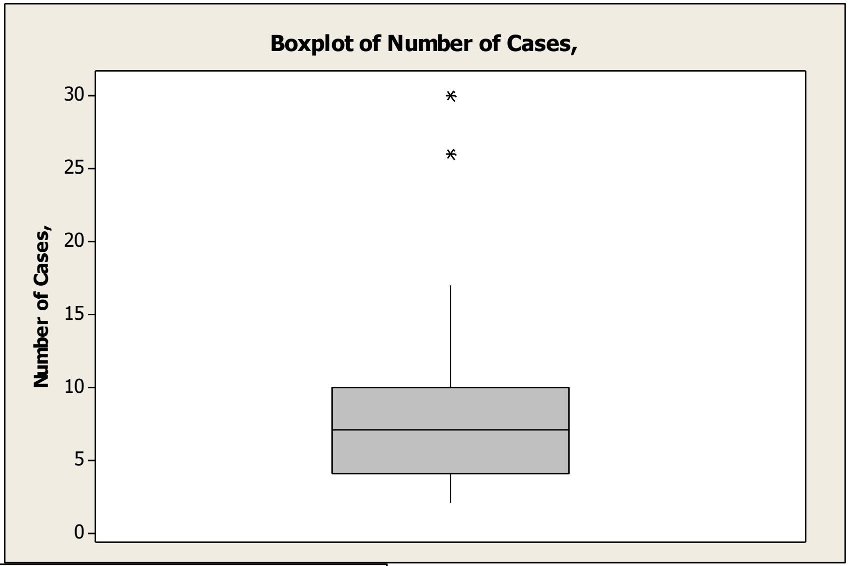
- Sorted data for price (in PhP): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
  - Partition into equal-frequency (equi-depth) bins:
    - Bin 1: 4, 8, 9, 15
    - Bin 2: 21, 21, 24, 25
    - Bin 3: 26, 28, 29, 34
  - Smoothing by bin means:
    - Bin 1: 9, 9, 9, 9
    - Bin 2: 23, 23, 23, 23
    - Bin 3: 29, 29, 29, 29

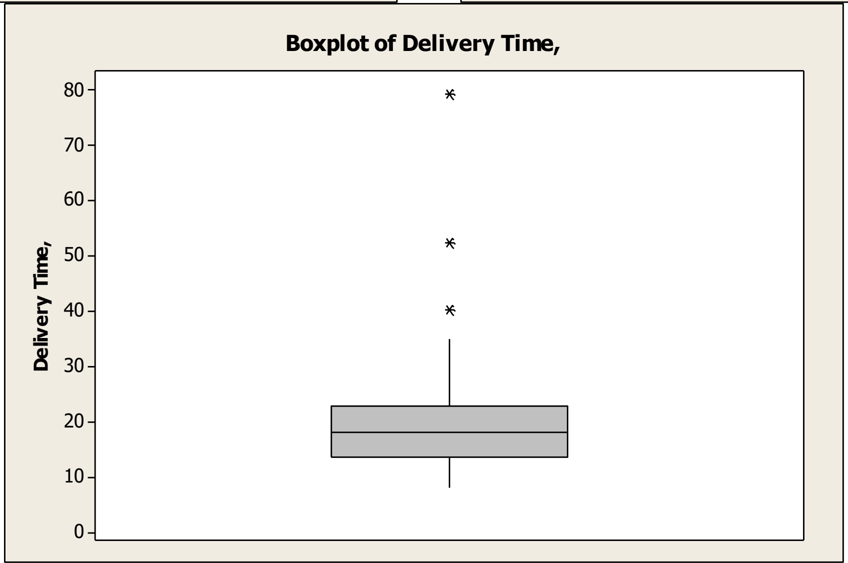
Outlier Identification

- An outlier is an observation that appears to **deviate from other observations** in the sample.
  - An outlier may indicate bad data
  - May need to consider the use of robust statistical techniques
- In Box Plots:
  - lower inner fence: Q1 - 1.5*IQR
  - upper inner fence: Q3 + 1.5*IQR
  - lower outer fence: Q1 - 3*IQR
  - upper outer fence: Q3 + 3*IQR
- A point beyond an inner fence on either side is considered a **mild outlier**.
- A point beyond an outer fence is considered an **extreme outlier**.

Delivery Time Data Box Plots







Outline for this Session

- Introduction to Data Preprocessing
- Data Integration
- Data Transformations
  - Data Discretization
  - Data Encoding
- Data Cleaning
- **Data Reduction**
- Case Study: Data Preprocessing Using R

Data Reduction/Manipulation

- Data may not be balanced.
- E.g.: Medical Dataset with 9900 negative cases and only 100 positive cases.

eljalao@up.edu.ph

Sampling

- Sampling is the main technique employed for data selection.
  - It is often used for both the preliminary investigation of the data and the final data analysis.

- Statisticians sample because obtaining the entire set of data of interest is too expensive or time consuming.

- Sampling is used in data mining because processing the entire set of data of interest is too expensive or time consuming.

eljalao@up.edu.ph

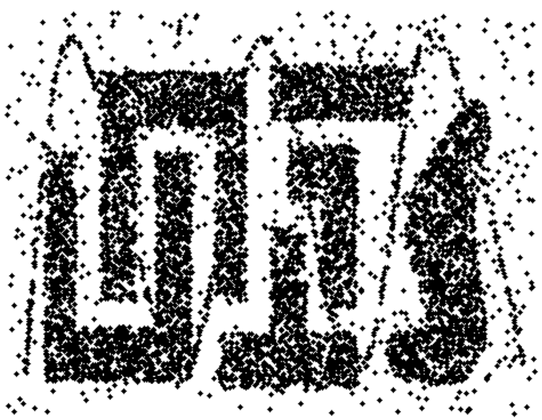
Sampling

- The key principle for effective sampling is the following:
  - using a sample will work almost as well as using the entire data sets, if the sample is representative
  - A sample is representative if it has approximately the same property (of interest) as the original set of data
- Types of Classification Sampling
  - Upsampling: Randomly select tuples from minority class to increase samples (sometimes called Bootstrapping)
  - Downsampling: Randomly select records from majority class to decrease samples

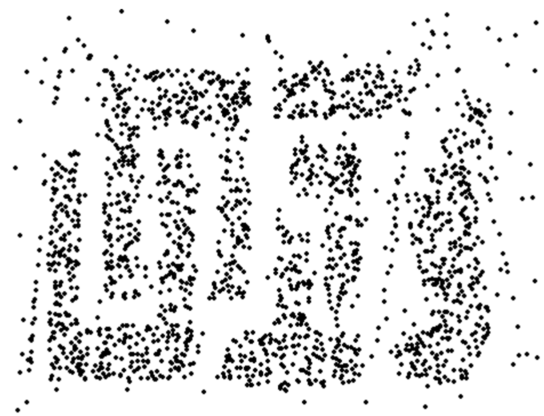
Types of Sampling

- Simple Random Sampling
  - There is an equal probability of selecting any particular item
- Sampling without replacement
  - As each item is selected, it is removed from the population
- Sampling with replacement
  - Objects are not removed from the population as they are selected for the sample.
    - In sampling with replacement, the same object can be picked up more than once
- Stratified sampling
  - Split the data into several partitions; then draw random samples from each partition

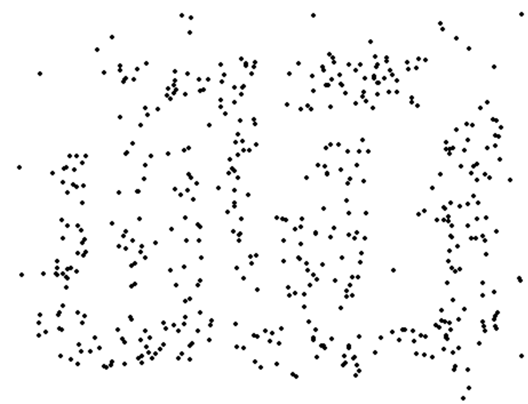
Sample Size



8000 points



2000 Points



500 Points

Example

- Using a reduced Heisenberg Data, do upsampling and downsampling on the dataset

trial	mass	velocity
A	10.0	12
A	11.0	14
B	5.0	8
B	6.0	10
A	10.5	13
B	7.0	11

Example

- `library(caret)`
- `heisenbergimbalanced = heisenberg[2:6,]`
- `heisenbergdown = downSample(heisenbergimbalanced,heisenbergimbalanced$trial)`
- `heisenbergup = upSample(heisenbergimbalanced,heisenbergimbalanced$trial)`
- `heisenbergdown`
- `heisenbergup`

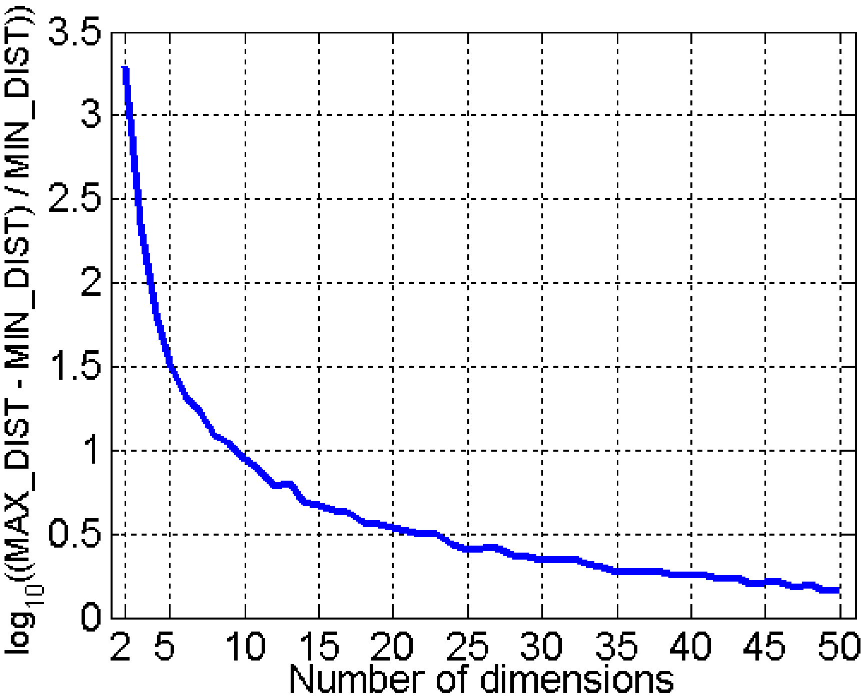
Sample R Code

```
> heisenbergdown
  trial mass velocity Class
1     A 10.5       13     A
2     A 11.0       14     A
3     B  7.0       11     B
4     B  5.0        8     B
> heisenbergup
  trial mass velocity Class
1     A 11.0       14     A
2     A 10.5       13     A
3     A 11.0       14     A
4     B  5.0        8     B
5     B  6.0       10     B
6     B  7.0       11     B
```

Curse of Dimensionality

- When dimensionality increases, data becomes increasingly sparse in the space that it occupies

- Definitions of density and distance between points, which is critical for clustering and outlier detection, become less meaningful



Feature Subset Selection

- A way to reduce dimensionality of data
- Redundant features
  - duplicate much or all of the information contained in one or more other attributes
  - Example: purchase price of a product and the amount of sales tax paid
- Irrelevant features
  - contain no information that is useful for the data mining task at hand
  - Example: students' ID is often irrelevant to the task of predicting students' GPA

Feature Subset Selection

- Techniques:
  - Brute-force approach:
    - Try all possible feature subsets as input to data mining algorithm
  - Embedded approaches:
    - Feature selection occurs naturally as part of the data mining algorithm
  - Filter approaches:
    - Features are selected before data mining algorithm is run
  - Wrapper approaches:
    - Use the data mining algorithm as a black box to find best subset of attributes

Feature Creation

- Create new attributes that can capture the important information in a data set much more efficiently than the original attributes

- Three general methodologies:
  - Feature Extraction
    - domain-specific
  - Mapping Data to New Space
  - Feature Construction
    - combining features

Outline for this Session

- Introduction to Data Preprocessing
- Data Integration
- Data Transformations
  - Data Discretization
  - Data Encoding
- Data Cleaning
- Data Reduction
- **Case Study: Data Preprocessing Using R**

Case 1: Data Preprocessing with R

- Preprocess the Bank Data

References

- James Lee Notes From: http://www.sjsu.edu/people/james.lee/courses/102/s1/asDescriptive_Statistics2.ppt
- Tan et al. Intro to Data Mining Notes
- www.cs.gsu.edu/~cscyqz/courses/dm/slides/ch02.ppt