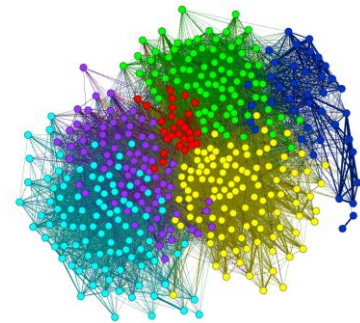


Data Cleaning

GATES



Overview

Data cleaning is used to describe anything that needs to be done to prepare data for analysis, visualization, and/or modeling.

For this reason, **“cleaning”** is a broad term that can and will mean different things depending on the **nature of the data** (such as text, record, biological, image, etc.), the **methods or models** you plan to apply, and the **programming language** you plan to use.

This is not a one-size-fits-all, but rather an active and situation-specific process.

There are some common elements that are employed when generally cleaning record data, such as managing incorrect, missing, or improperly formatted data, as well as dealing with outliers.

Examples of Cleaning Steps

Example 1: Suppose you use an API to download text data from a news station in a csv file (with each row being all the text from the new story). Then, to “clean” this data, you will need to tokenize, vectorize, remove punctuation, remove capital letters, and remove stop words,. You may also elect to reduce the dimensionality by keeping only critical words (which will depend on the goals and applications). You may also choose to normalize the data – such as with min-mx or tf – idf.

Example 2: Suppose you work for a medical company, and they give you a large dataset with patient data, including dates, times, visits, age, weight, height, etc. This is record data and is mixed (some qualitative and some quantitative). It also likely contains location information (where the patient lives) and temporal data (when different things happened). Cleaning this data will be completely different than cleaning the text data. Here, you will need to be very careful not to remove any critical information. You will need to manage missing values, incorrect values, incomplete values, values in the wrong format, odd (outlier) values, etc.

The above only illustrates the first steps. The next steps will involve formatting, normalizing, transforming, feature generation, and any steps needed to use specific model, methods or languages. For example, if you plan to use Naïve Bayes in Python with sklearn, you will need all numeric data and it will need to be labeled.

Outline

Consider/Update Types, Formats, and Model Goals

Introduction to “Cleaning”

Missing Values

Incorrect Values

Duplications

Outliers

Formatting for Models

Transformation

Normalization

Types, Formats, and Model/Method Goals

Data can be:

- 1) Quantitative (numeric) (discrete or continuous, interval or ratio)
- 2) Qualitative (ordinal or nominal)

And...can be temporal (time-based like dates) or geographical (locations).

Datasets can be:

- 1) Record
- 2) Text
- 3) Matrix
- 4) Transaction
- 5) Sequential
- 6) Image, etc.

Introduction to Cleaning

Data is the **beginning** of information discovery.

To retrieve information from data, the data must first be **prepared**, then explored/visualized, then analyzed and modeled, and then its information can be presented/visualized.

Data cleaning is a cyclic and iterative set of processes that includes exploring, visualizing, updating, formatting, transforming, normalizing, discretization, and correcting data. It is different depending on the data and the goals.

Data preparation (cleaning) often **also** includes steps such as the identification and management of outliers, as well as the generation of new features. **These processes are repeated** until the data is prepared for analysis, or *clean*, **AND, are different for different goals and data types.**

Missing Values

A **missing value** is a single attribute (variable) observation or value that is not available.

For example, if a dataset has a variable called Age and one of the Age values is missing for one of the rows, this is a **missing value**.

A missing value may result from a **lack of data** during collection, such as a survey question left blank, or may result from **errors** in rendering the data, such as data-entry issues.

Missing values can have a significant impact on the **quality** of a dataset and on the **information** that it contains.

If a given column in a dataset contains too many missing values, it may not be a usable variable for analysis.

Cleaning Record Data – Basic Steps

While data cleaning can be unique for each dataset, each model or method, and each goal, there are some core commonalities when cleaning/preparing record style data. Recall that record data is organized into rows and columns.

STEPS:

- 1) Managing Missing Values
- 2) Managing Incorrect (or incorrectly formatted) Values
- 3) Dealing with duplicates
- 4) Managing outliers

This tutorial is long and covers introductory and complex topics. It also include code examples.

Missing Values: Examples

>gi|2978501|gb|AAC06133.1| vacuolar ATPase proteolipid subunit [Giardia intestinalis]

```
MSSIDSPVAVEKCPAGASFWMLGQVVAVVFSSIGAAYGTAKAGSGLGV  
AGLINPAPVTKLTLPI AGILSIYGLITSLINSRVRSYTNGMPLYVS  
YAHFGAGLCCGLAALAAGLAIGVSGSAAVKAVAKQPSLFVVMLIVLIFS  
EALALYGLIHALIL TKSADSNFCVNNVNQ
```

ID	Class	Group	Gender	Age	Ticket	Cost	Type
1	0	1	Male	21	Yes	3.25	
2	1	1	female	32	4521	8.23	B11
3	1	3	1	167	AAB44	77.45	
4	1	3	female	55	10194	92.13	D12
5	0	1	male	12	A783		
6	0	3	M	67	No	100045	
7	0	2	male	-1	0	45.32	F34
8	A	3	0	apple	2323	2.77	
9	1	3	F.	72	1347742	210.13	
10	1	2	FEMALE	0.14	2336	31.11	
11	1	2	FEMALE	0.14	2336	31.11	



0	0	2	0	0	0	0	0	0	0
0	0	1	0	0	0	0	2	0	0
0	1	0	4	0	0	0	0	0	0
0	0	0	0	3	0	0	0	0	0
0	0	0	0	0	0	6	0	4	0
0	0	0	0	3	0	0	4	0	0
0	0	0	0	0	0	0	0	0	5
0	0	0	0	0	0	5	0	0	0
3	0	0	0	0	3	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Finding Missing Values

Generally (but of course not always) – finding Missing Values can be straightforward.

However, **correcting** missing values can be difficult or impossible.

Why?

Incorrect Values

Incorrect values are often **more challenging to locate** within a dataset than missing values.

Unlike missing values, which tend to adhere to a given format, such as blank or NA, **incorrect values can be anything.**

Therefore, incorrect values must first be **discovered.**

This is done using code – such as R or Python.

This requires knowledge about the dataset and the domain.

Incorrect Values: Easier

Where are the incorrect values?

How would you FIND them with code?

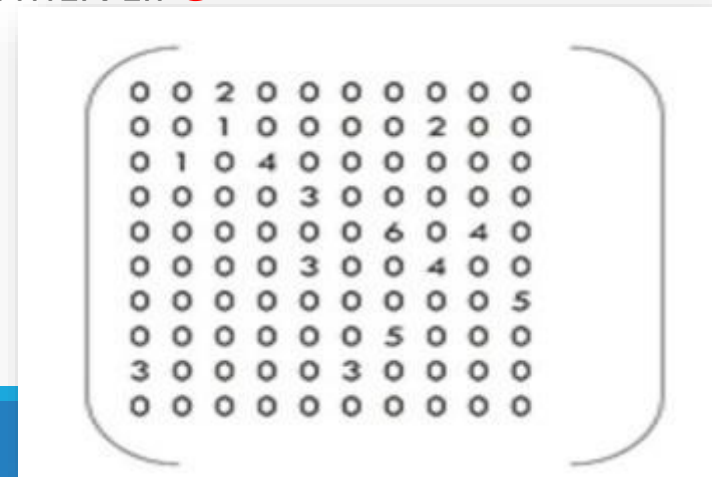
ID	Class	Group	Gender	Age	Ticket	Cost	Type
1	0	1	Male	21	Yes	3.25	
2	1	1	female	32	4521	8.23	B11
3	1	3	1	167	AAB44	77.45	
4	1	3	female	55	10194	92.13	D12
5	0	1	male	12	A783		
6	0	3	M	67	No	100045	
7	0	2	male	-1	0	45.32	F34
8	A	3	0	apple	2323	2.77	
9	1	3	F.	72	1347742	210.13	
10	1	2	FEMALE	0.14	2336	31.11	
11	1	2	FEMALE	0.14	2336	31.11	

Incorrect Values: Harder

actually	alicia	babysitter	...	worth	year	zero
0.000000	0.510737	0.255368	...	0.000000	0.028774	0.408589
0.122472	0.000000	0.000000	...	0.000000	0.042761	0.000000
0.000000	0.000000	0.000000	...	0.161917	0.056533	0.000000
0.124596	0.000000	0.000000	...	0.062298	0.000000	0.000000
0.000000	0.000000	0.000000	...	0.000000	0.036179	0.000000

>gi|2978501|gb|AAC06133.1| vacuolar ATPase proteolipid subunit [Giardia intestinalis]

MSSIDSPVAVEKCPAGASFWMLGQVVAVVFSSIGAAYGTAAGSGLGV
AGLINPAPVTKLTPVIAGILSIYGLIT**A**LLINSRVRSYTNGMPLYVS
YAHFGAGLCCGLAALAAGLAIGVSGSAAVKAVAKQPSLFVVMLIVLIF**G**
EALALYGLIILITKSADSNFCVNNVNQ



Duplications

1	bread	coffee	soymilk	quinoa
2	coffee	quinoa		
3	bread	coffee	soymilk	quinoa

Datasets can sometimes have **duplicated data**.

These duplicates may be **true duplicates** - **errors** caused by the same exact data being **repeated by accident**.

→ To identify a TRUE duplicate, the data row (in record data) must have a **primary key**. This is a column that is a **unique identifier**.

Duplicates may NOT be true duplicates.

→ For example, in transaction data, two transactions may be identical, but made by different people.

Can you think of examples of true and not true duplication?

How could you determine this?

Which duplicate are you SURE of and why?

SS#	Name	Age	Hobby
343-39-5674	Bob	23	Hiking
123-45-3737	Sally	30	Swimming
343-39-5674	Robert	23	Hiking

Lastname	Firstname	Hobby	Major
Xu	Jia	Basketball	Math
Wang	Ben	Football	CS
Wang	Ben	Football	CS

Outliers: How do you Define “too different” or “far”.

- 1) **An outlier**, while noisy and perhaps incorrect, is in its own data –cleaning category.
- 2) An outlier is a value that is **significantly “far” or “different”** from the other values and from what is expected.

Examples:

- (a) The value -1 for AGE is incorrect – but NOT an outliers.
- (b) The value 125 for AGE is likely incorrect, but also **not an outlier**.
- (c) However, a value of 13456 is an outlier.

An outlier value is **significantly different (non-similar) to expected values**.

****The concept of significantly different requires a measure of similarity.**

Outlier Definition: Nope!

There is no universal definition for an outlier.

Common definitions include:

- an observation that differs **so much** from other values that it raises suspicion (Hawkins, 1980), or
- an observation that appears to be **very inconsistent** with the other data (Barnett and Lewis, 1994).

In both of these definitions, the notion of an outlier is that it is rare (so very few or just one data point), and very different (far) from all the other data points.

However, even the idea of “different” is based on a measure of similarity or distance, and different distance measures can offer different results.

Finding Outliers?

- 1) This depends on the nature of the data.
- 2) This depends on knowledge of the domain.
- 3) This depends on measures of “distance” or “similarity”

Different might mean NEW and not WRONG!

Methods:

- 1) Outside of known ranges (such as with AGE).
- 2) Outside of +/- 3 st. dev from the mean (if close to normal and numeric)
- 3) Outside of all clusters...

ETC...

Formatting for Models

Cleaning is NOT the same for all methods and models and is NOT the same for all data types.

1) Cleaning data to use with an SVM will be different than for DT or NB, or R, or Python, or text data, or sequential data...etc.

2) The same dataset may be cleaned, prepared, formatted, etc. DIFFERENTLY for each model or method it will be applied to.

Transformation

The **look** or expression of data can be altered without destroying its relative integrity.

1) Math Transformations:

- Sqrt
- Log
- Square, cube, ...
- kernel

2) Discretization:

- Binning
- Conversion to Binary
- Aggregation

Normalization

Example:

Suppose California has 1000 cases of Covid and New Hampshire has 500.

Which state needs support first?

Normalization Methods

1) TEXT

- Tf-Idf
- Division across rows by the total number of words

2) RECORD

- Change “counts” to “relative frequencies”.
- Use the **z-value** to compare variables with different means and std devs.
- Use **min-max** to force all values between any two values (such as between 0 and 1)

Example:

There are 15 sections of ANLY 501. Each section has a different number of students. All sections awarded 5 A grades. Which teacher gave the most A grades? How can you tell?

Sampling and Subsetting

Not all variables or parts of a dataset can be used with all methods or models.

1) In some cases, only the quantitative (numeric) data can be used.

This is true for SVM, NB in Python (with the common library), clustering with k-means.

2) In some cases, one might choose to look at only categorical data – such as with DT, NB (in R), and ARM.

3) Often, datasets are so large, that samples or subsets are used to create and test models before the entire dataset is applied.

Aggregation

Columns (variables) in record data can be aggregated (combined using a function) so as to create a new column and/or reduce dimensionality.

Example: What can we aggregate here?

Passenger	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. Jc	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss.	female	26	0	0	STON/O2.	7.925		S
4	1	1	Futrelle, Mrs. Jac	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. William	male	35	0	0	373450	8.05		S
6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q
7	0	1	McCarthy, Mr. Tim	male	54	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master.	male	2	3	1	349909	21.075		S
9	1	3	Johnson, Mrs. O	female	27	0	2	347742	11.1333		S
10	1	2	Nasser, Mrs. Nicol	female	14	1	0	237736	30.0708		C
11	1	3	Sandstrom, Miss	female	4	1	1	PP 9549	16.7	G6	S
12	1	1	Bonnell, Miss. El	female	58	0	0	113783	26.55	C103	S
13	0	3	Saunderscock, Mr	male	20	0	0	A/5. 2151	8.05		S
14	0	3	Andersson, Mr. A	male	39	1	5	347082	31.275		S

Joining (Combining Datasets)

It is often necessary to combine datasets so as to build more robust models and/or more descriptive visualizations.

For example – how might you JOIN these two datasets and why?

OilName	Lymph	Detox	Liver	Kidney	Digestion	Fe
Yarrow	NO	NO	NO	NO	NO	NO
Buchu	NO	YES	NO	NO	YES	NO
Lemon Ve	NO	NO	NO	NO	NO	YE
Galangal	NO	NO	NO	NO	YES	NO
Dill	NO	YES	NO	NO	YES	NO
Angelica	NO	YES	NO	NO	NO	YE
Rosewood	NO	NO	NO	NO	NO	NO

OilName	CountryOrigin
Yarrow	Australia
Buchu	Australia
Lemon Ve	USA
Galangal	Greece
Dill	France
Angelica	Greece
Rosewood	USA

Feature Generation and Data Harmonization

It is often the case that **multiple datasets** are utilized in data analysis.

They may have arisen from **different sources** and may have **different formats** and naming conventions.

Harmonizing data involves blending and combining multiple datasets together such that all information is properly retained and represented.

Feature generation (also known as feature engineering) is the process of creating new variables (or features) **using the current variables in the dataset.**

New features may be created through discretization, the application of a function, or the alteration of a format.

Feature Generation Example

Example:

Suppose a dataset contains the variable **DateOfBirth**.

This is important information, but will create a challenge when performing analysis.

However, the DateOfBirth variable may be used **to generate a new feature (column) called Age**, which is created using a function that subtracts the date of birth from the current date.

DateOfBirth is a “date” type and is not quantitative.

Age is a numeric data type that can be used more broadly in analyses.

Remapping

1) Words to numbers...

Example 1: Text data to tokenized and vectorized frequency count dataframe.

Example 2: Male and Female mapped to 0 and 1.

Example 3: Theatre Ticket cost (quantitative) mapped to an ordinal variable (such as Group1, Group2, and Group3 – this is discretization).

Example 4: Converting qualitative values to quantitative values via weights.

Example1: remapping

Label	Numeric remap
Single	0
Divorced	.25
Widowed	.5
Remarried	.75
Married	1

Label	Numeric remap
English	0
French	1
Spanish	2
Mandarin	3

Example 1: Titanic (Kaggle)

Passenger	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. Jc	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss.	female	26	0	0	STON/O2.	7.925		S
4	1	1	Futrelle, Mrs. Jac	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. William	male	35	0	0	373450	8.05		S
6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q
7	0	1	McCarthy, Mr. Tim	male	54	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master.	male	2	3	1	349909	21.075		S
9	1	3	Johnson, Mrs. Os	female	27	0	2	347742	11.1333		S
10	1	2	Nasser, Mrs. Nicol	female	14	1	0	237736	30.0708		C
11	1	3	Sandstrom, Miss	female	4	1	1	PP 9549	16.7	G6	S
12	1	1	Bonnell, Miss. Eli	female	58	0	0	113783	26.55	C103	S
13	0	3	Saunderscock, Mr	male	20	0	0	A/5. 2151	8.05		S
14	0	3	Andersson, Mr. A	male	39	1	5	347082	31.275		S

Missing Values

Options for correction:

- Replacing with a measures such as the mean, median, or mode.

What are some concerns with this?

When should rows be removed?

When should columns be removed?

How can the BEFORE and AFTER be measured?

Incorrect Values

- 1) What can be done?
- 2) What are issues?
- 3) How can BEFORE and AFTER be measured?
- 4) What are examples of incorrect values?

Format Issues

Example:

Suppose a variable is **Gender**.

Suppose the data is:

M, M, F, F, F, M, F, 0, M, F, 1, Female, M, MALE

What can be done here?

How can Python be used to do it?

Value re-assignment - Normalization

Normalization:

- Every dimension (attribute) is constructed so that its maximum and minimum values are the same.
- Typically 0 is the minimum and 1 is the maximum. This creates a **unit state space**.

Linear scaling (Min-Max Normalization)

- $$X_{\text{normalized}} = \frac{x_i - \min(x_1 \dots x_n)}{\max(x_1 \dots x_n) - \min(x_1 \dots x_n)}$$

where x_i = the i^{th} value of the variable

Value re-assignment – Normalization cont.

Z-Score standardization

- $x_{\text{normalized}} = \frac{x_i - \text{mean}}{\text{standard deviation}}$

Decimal scaling

- $x_{\text{normalized}} = x_i / 10^j$

where j is the smallest integer such that $\text{Max}(|x_i|) < 1$

Normalizes by moving the decimal point

Normalization - Know What You Are Doing and Why.

	hike	mountain	dog	coffee
Doc1	3	4	0	11
Doc2	1	10	4	3
Doc3	10	5	11	5

Doc 1 has 1000 words in it.

Doc 2 has 100 words in it.

Doc 3 has 100,000 words in it.

Should you normalize? How?

Outliers?

An **outlier** is a value or an **observation that is distant (far or dissimilar) from other observations,**

→ A data point (row or vector) that **differs significantly** from other data points.

Is an Outlier:

1) A mistake?

2) An extreme or different value that is unusual but correct?

→ There is no strict or unique rule that says that outliers should be removed from a dataset.

→ What should you do? How should you find outliers?

Deep Reference: <http://www.eng.tau.ac.il/~bengal/outlier.pdf>

The NASA Ooops of Outliers

<http://www.realclimate.org/index.php/archives/2017/12/what-did-nasa-know-and-when-did-they-know-it/>

A machine collecting data on the ozone removed extremely low values because it logged them as outliers.

In fact, there were correct and represented a hole in the ozone layer.

Is it OK to remove outliers? Sometimes.

Are there rules for removing outliers? No.

Can an outlier represent a true datapoint? Yes.

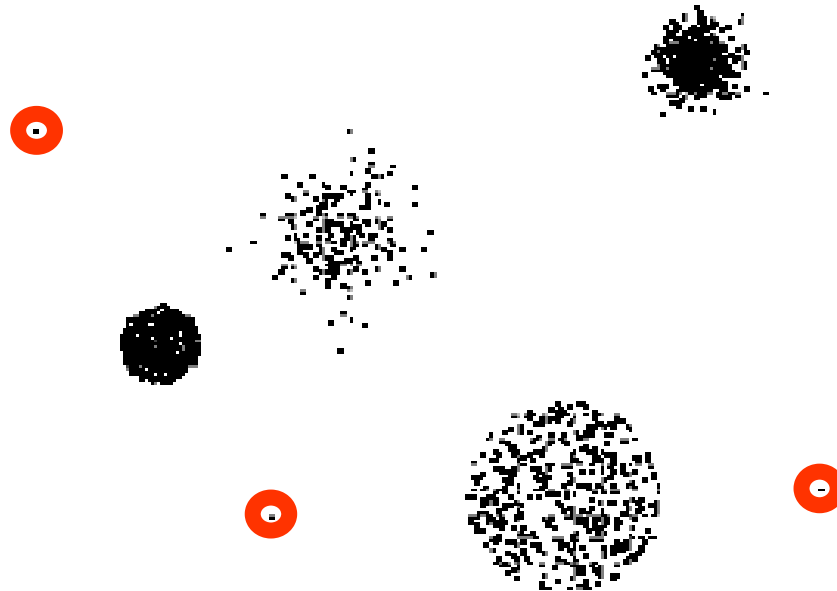
Outliers and R: Example

<https://drive.google.com/file/d/19asjLgGxwGF9bCTSKUwLtXdzCLC7f62m/view?usp=sharing>

Including IQR, Vis, Grubbs, and more....

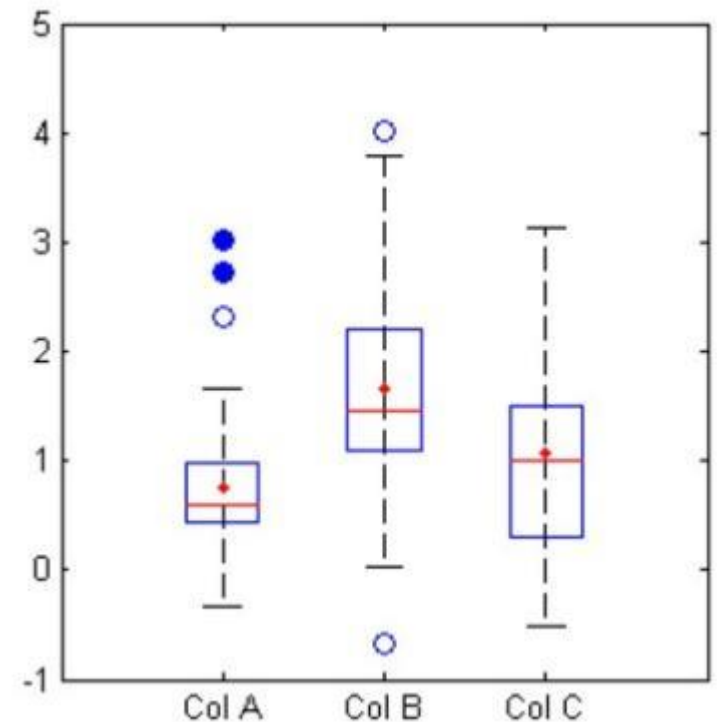
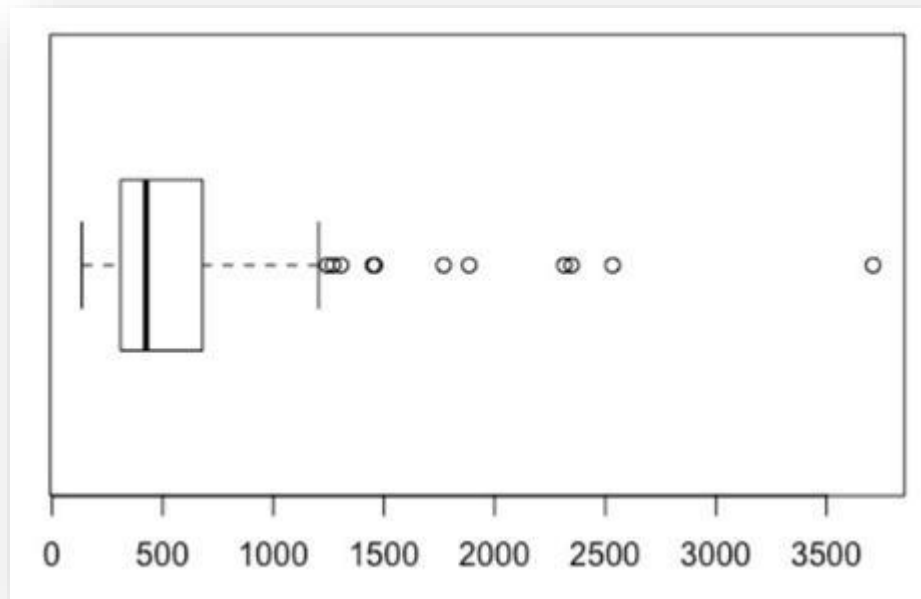
Outliers

Outliers are data objects with characteristics that are considerably different than most of the other data objects in the data set



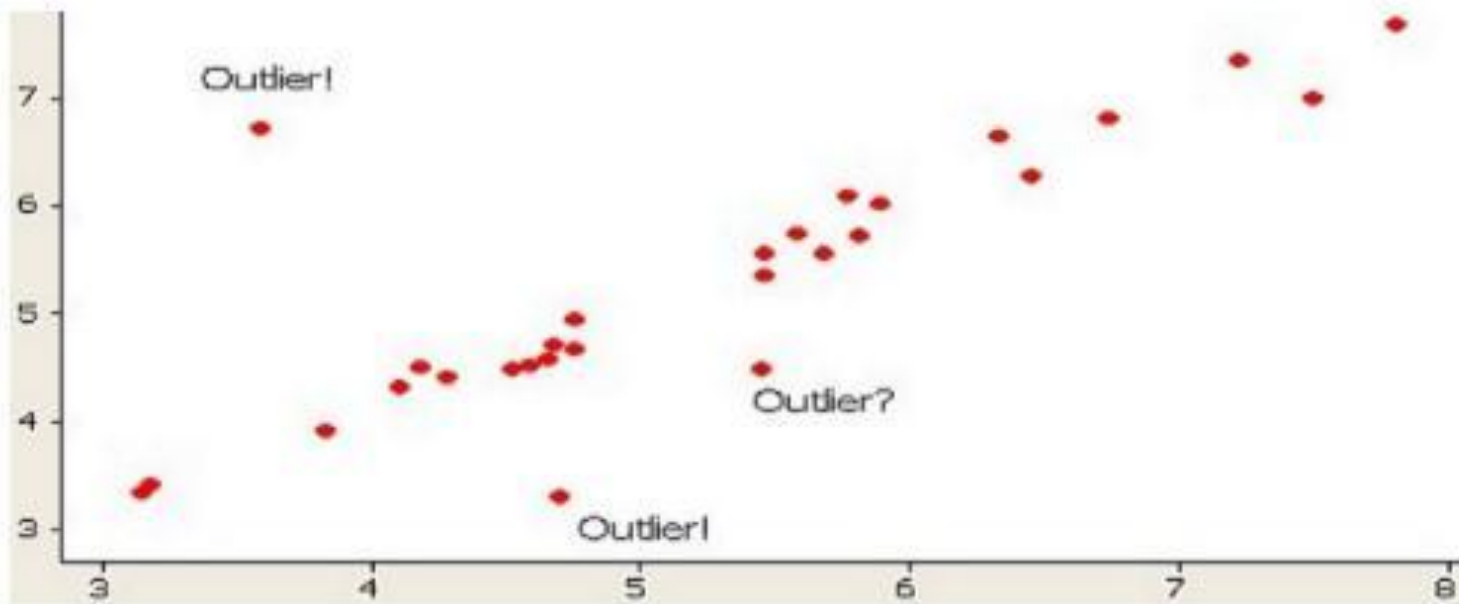
Example 1: Visualizing Outliers

Boxplots and IQR

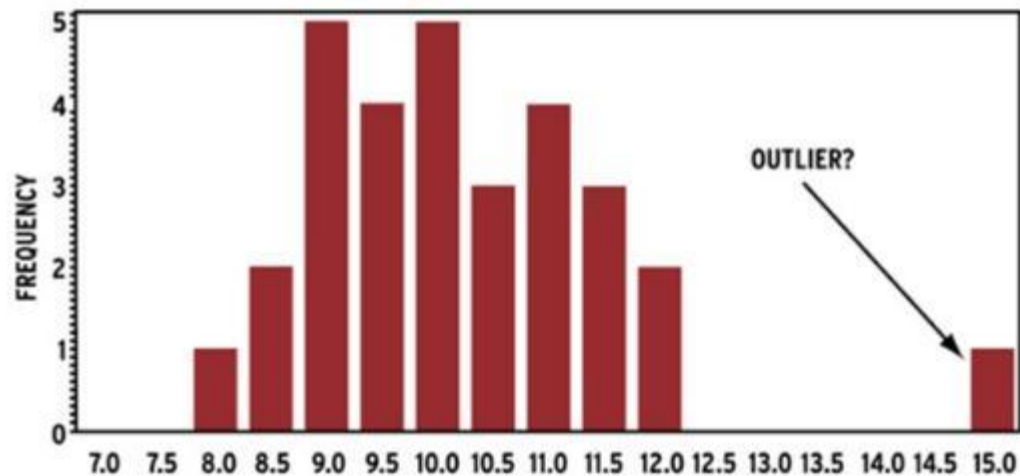


<http://wiki.eigenvector.com/index.php?title=Boxplot>

Example 2: Visualizing Outliers Scatterplots



Example 3: Visualizing Outliers Histograms



Outliers

Extreme values that lie near the limits of the data range or go against the trend of the remaining data.

Example:

2, 3, 7, 9, 1000, 4, 5

Important: **Outliers are not always wrong!**

Outlier detection using IQR

The IQR is defined as the difference between the upper quartile and the lower quartile of a data set, sometimes written as $Q3 - Q1$.

A data value is an outlier if:

- It is $1.5(IQR)$ or more below $Q1$
- It is $1.5(IQR)$ or more above $Q3$

Grubbs: About

What is Grubbs' Test for Outliers?

Grubbs' test is used to find a **single outlier** in a normally distributed data set. The test finds if a minimum value or a maximum value is an outlier.

Cautions:

The test is only used to find a *single* outlier in normally distributed data (excluding the potential outlier). If you think that your data set has more than one outlier, use the **generalized extreme studentized deviate test** or **Tietjen-Moore test** instead.

Using this test on non-normal distributions will give false results.

Run a test for normality (like the Shapiro-Wilk test) *before* running Grubbs' test. If you find your data set isn't normally distributed, try removing the potential outlier from the data set and running the normality test again. **If your data still isn't normal, don't run this test.**

<https://www.statisticshowto.com/grubbs-test/>

Grubbs By Hand

Find the G test statistic.

Find the G Critical Value.

Compare the test statistic to the G critical value.

Reject the point as an outlier if the test statistic is greater than the critical value.

Ho: There is no outlier

Ha: There is an outlier

Grubbs Test

1. Find the G Test Statistic

Step 1: Order the data points from smallest to largest.

Step 2: Find the **mean** (\bar{x}) and **standard deviation** of the data set.

Step 3: Calculate the G test statistic using one of the following equations:

The Grubbs' test statistic for a **two-tailed test** is:

$$G = \frac{\max_{i=1, \dots, N} |Y_i - \bar{Y}|}{s}$$

Where:

\bar{y} is the **sample mean**,

s = sample **standard deviation**.

A left-tailed test uses the test statistic:

$$G = \frac{\bar{Y} - Y_{\min}}{s}$$

Where Y_{\min} is the minimum value.

For a right-tailed test, use:

$$G = \frac{Y_{\max} - \bar{Y}}{s}$$

Where Y_{\max} is the maximum value.

Grubbs Test

2. Find the G Critical Value.

N	Alpha				
	0.1	0.075	0.05	0.025	0.01
3	1.15	1.15	1.15	1.15	1.15
4	1.42	1.44	1.46	1.48	1.49
5	1.6	1.64	1.67	1.71	1.75
6	1.73	1.77	1.82	1.89	1.94
7	1.83	1.88	1.94	2.02	2.1
8	1.91	1.96	2.03	2.13	2.22
9	1.98	2.04	2.11	2.21	2.32
10	2.03	2.1	2.18	2.29	2.41
11	2.09	2.14	2.23	2.36	2.48
12	2.13	2.2	2.29	2.41	2.55
13	2.17	2.24	2.33	2.46	2.61
14	2.21	2.28	2.37	2.51	2.66
15	2.25	2.32	2.41	2.55	2.71
16	2.28	2.35	2.44	2.59	2.75
17	2.31	2.38	2.47	2.62	2.79

Manually, you can find the G critical value with a formula.

$$G > \frac{N - 1}{\sqrt{N}} \sqrt{\frac{t_{\alpha/(2N), N-2}^2}{N - 2 + t_{\alpha/(2N), N-2}^2}}$$

Where:

$t_{\alpha/(2N), N-2}$ is the upper critical value of a **t-distribution** with N-2 **degrees of freedom**.

For one-tailed test, replace $\alpha/(2N)$ with α/N .

Accept or Reject the Outlier

Compare your G test statistic to the G critical value:

$G_{\text{test}} < G_{\text{critical}}$: keep the point in the data set; it is **not** an outlier.

$G_{\text{test}} > G_{\text{critical}}$: reject the point as an outlier.

Other Examples of Noise

Random error or **variance** in a measured variable.



Examples:

- Audio (voice poor quality on phone)
- Video (“snow on TV)
- Sensor noisy
- Survey data collection by different people

Noisy values may be due to:

- Faulty data collection instruments
- Data entry problems
- Data transmission problems
- Technology limitations
- Inconsistent naming conventions

Noisy Tweets: What is noise?

1. #ISIS is the end of human slavery! #IslamicState
2. OMG check this out: <http://t.co/g5fmtCOGPJ>
3. Brother Abu Suhaib Al Jazrawi May Allah accept him #ISIS #IslamicState   <http://t.co/g5fmtCOGPJ>
4. a crowded fruit & vegetable market in Jarablus, #Aleppo #IslamicState #IS #ISIS #MessageTolsis
5. If it's not clear to you at this point that twitter is allowing #ISIS to terrorize people you're a fool

How to Handle Noisy Data?

Binning method

- Looked at neighborhood and locally **smooth** data.

Regression

- Smooth by fitting the data into regression functions

Clustering/Anomaly Detection algorithms

- Detect and remove outliers

Combined computer and **human inspection**

- Detect suspicious values and check by human and then build a classifier

Binning



54.876985

63.345339

52.478001

62.047462

59.692849

63.116668

52.394022

60.539069

56.14605

56.553081

57.899076

59.550554

61.852781

55.549742

50.020629

62.877525

60.620873

54.269527

50.999137

59.763285

61.565763

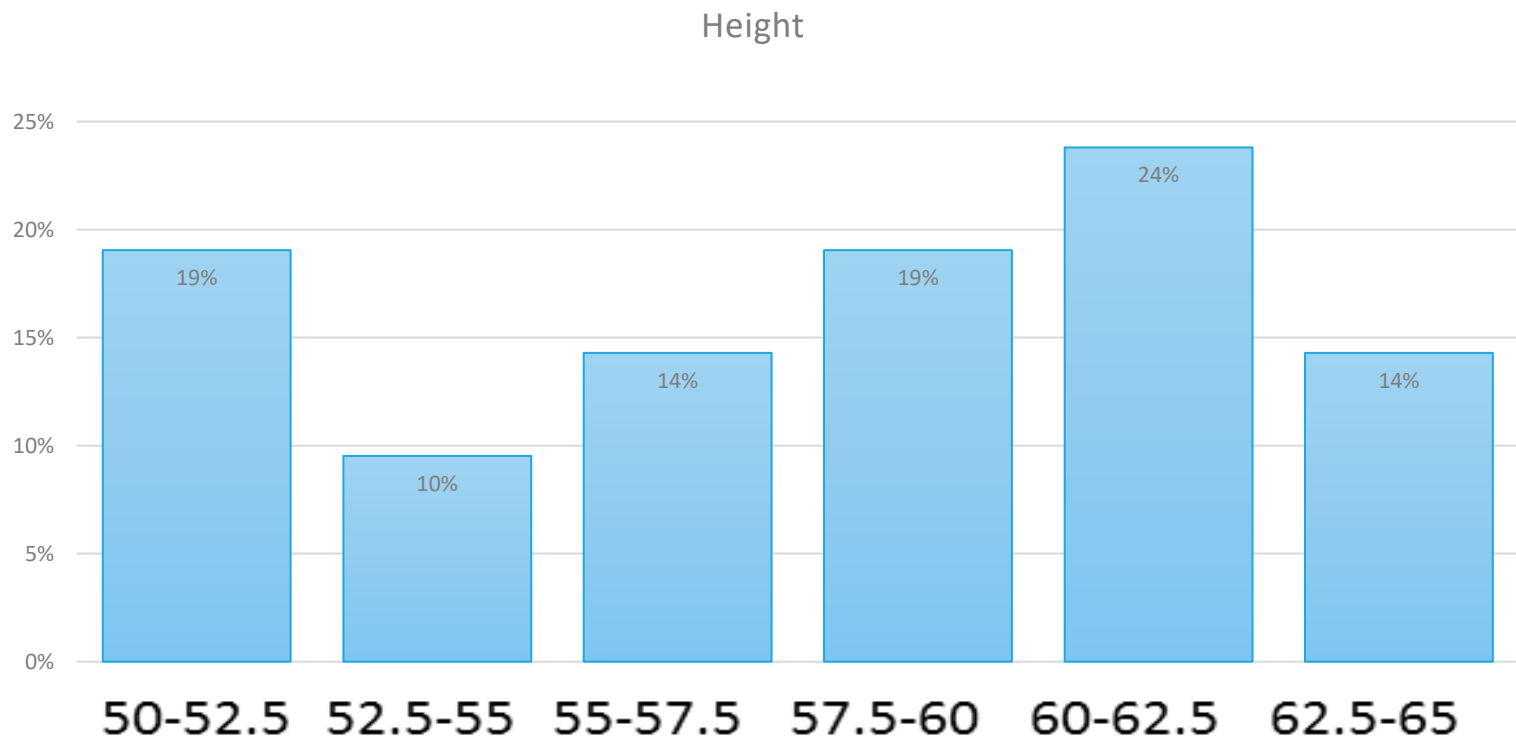
Example Dataset: height in inches

$(65 - 50) / 6 = 15 / 6 = 2.5$ So, the WIDTH of each bin is 2.5.

Height	Range	Absolute Frequency	Relative Frequency	Cumulative Frequency
Bin 1	50 – 52.5	4	4/21=19.1%	19.1%
Bin 2	52.5 - 55	2	2/21=9.5%	19.1%+9.5%
Bin 3	55 – 57.5	3	3/21=14.3%	19.1%+9.5%+14.3%
Bin 4	57.5 - 60	4	4/21=19.1%	19.1%+9.5%+14.3%+19.1%
Bin 5	60 – 62.5	5	5/21=23.8%	19.1%+9.5%+14.3%+19.1%+23.8%
Bin 6	62.5 - 65	3	3/21=14.3%	19.1%+9.5%+14.3%+19.1%+23.8%+14.3%
Total		21	100%	100%



Bar Graph of binned data



Regression Equation Smoothing

Smooth by fitting the data into **regression functions**

Two approaches:

- First remove outliers and then update values to fit regression line
- Do not remove outliers and update values to fit regression line

Regression Equation Smoothing

<u>x</u>	<u>y</u>
1	13
2	12
3	14
4	30
5	18
6	19

1) Clean Data Equation

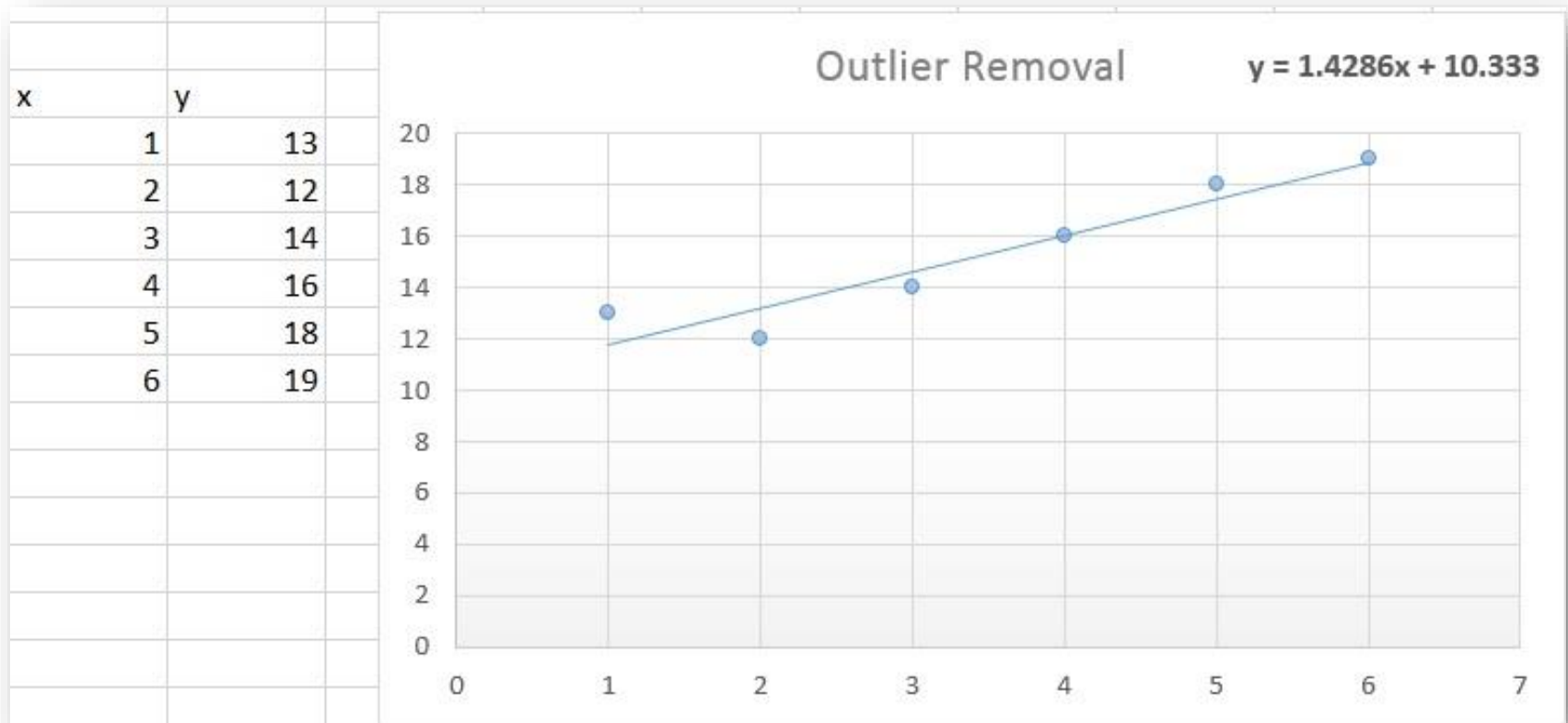
2) No outliers removed:
Equation is...

One Outlier removed (**30**):
Equation is...

Outliers Not Removed



Outlier Removed (30 removed)



More On Missing Values

Reasons for missing values

- Information is not collected
- Attributes may not be applicable to all cases
- Equipment malfunction
- Not important at time of data entry

Handling missing values

- Ignore the missing value during data analysis
- Eliminate data objects
- Fill in each missing value manually
- Estimate missing value with global constant (mean or mode)
- Replace with all possible values (weighted by their probabilities, e.g. use most probably value)
- Randomly select based on regression line

Maintaining Original Knowledge

When replacing a missing value - make sure to note that the value was originally missing - using a different attribute.

For each variable in the data set, we set a **flag** for present or empty. Creating such flags has a series of patterns.

While we may not do this in class – it is often critical when cleaning data for a company or job.

Also – always retain the original raw data.

Unbiased Estimator

A method for guessing a value of a particular attribute without changing important characteristics of the values in the data set.

Statistically, an **unbiased estimator** produces an estimate whose expected value is the value that would be estimated from the population.

Using the mean to replace a missing or incorrect value - what is the benefit and what is the risk?

Using the median to replace a missing or incorrect value - what is the benefit and what is the risk?

To Note:

The missing value estimate depends as much on which characteristic is to be unbiased as it does on the actual value. Therefore, we need to determine **which relationships need to be preserved**, both within and between variables.

If many missing values are replaced with the mean, the **confidence level** for statistical inference will be **overoptimistic** since the spread of the data will be reduced.

It is better to replace the value with **random draws** from the **variable distribution observed**. This means that the values will draw proportionally to the distribution and the center and spread should remain close to the original.

Example

Position	Original Sample	Position 11 Missing	Position 1 Missing
1	0.0886	0.0886	?
2	0.0684	0.0684	0.0684
3	0.3515	0.3515	0.3515
4	0.9874	0.9874	0.9874
5	0.4713	0.4713	0.4713
6	0.6115	0.6115	0.6115
7	0.2573	0.2573	0.2573
8	0.2914	0.2914	0.2914
9	0.1662	0.1662	0.1662
10	0.4400	0.4400	0.4400
11	0.6939	?	0.6939

Mean: 0.4023
STD: 0.2785

Mean: 0.3731
STD: 0.2753

Mean: 0.4336
STD: 0.2723

Mean: 0.3731
STD: 0.2612

Mean: 0.4336
STD: 0.2584

Key Ideas

- A **key point** is that although the replacement values are predictions, it is not the accuracy of these predictions that is of most importance.
- The key concern is that the predictions produce a workable estimate that **least distorts the values that are actually present**.
- The purpose of replacing the missing values is not to use the values themselves, but to make available the information that is available in the other variables' values that are present.
- If the missing values are not replaced, the whole instance may have to be ignored.

Linear Estimation

In a linear relationship, if the value of one variable changes a certain amount, the value of another variable changes by another certain amount in a specific direction.

In practice, assuming a linear relationship for missing data determination introduces very little bias.

The purpose of replacing missing values is not to use the values themselves, but to make available the information contained in the other variable-values that are present. **If the missing values are not replaced, the whole instance may be ignored.**

Other missing value advanced approaches: FYI

Nonlinear submodels

Neural networks

Nearest neighbor estimators

Aggregation

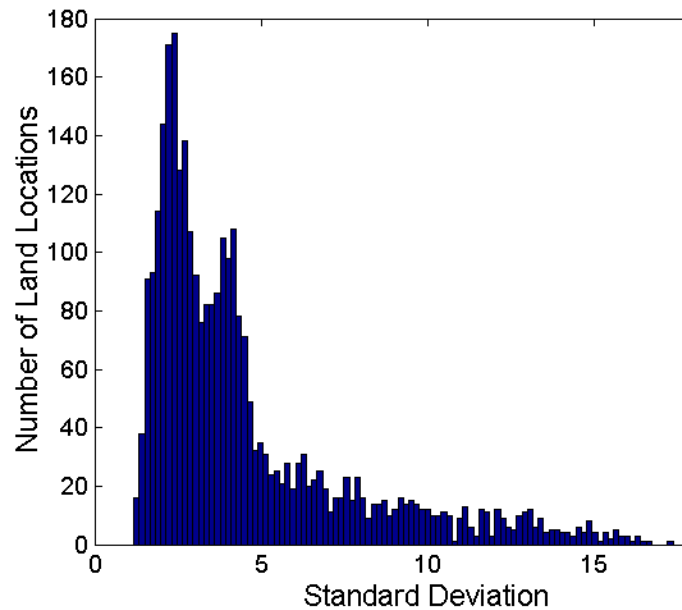
Combining two or more attributes (or objects) into a single attribute (or object)

Purpose

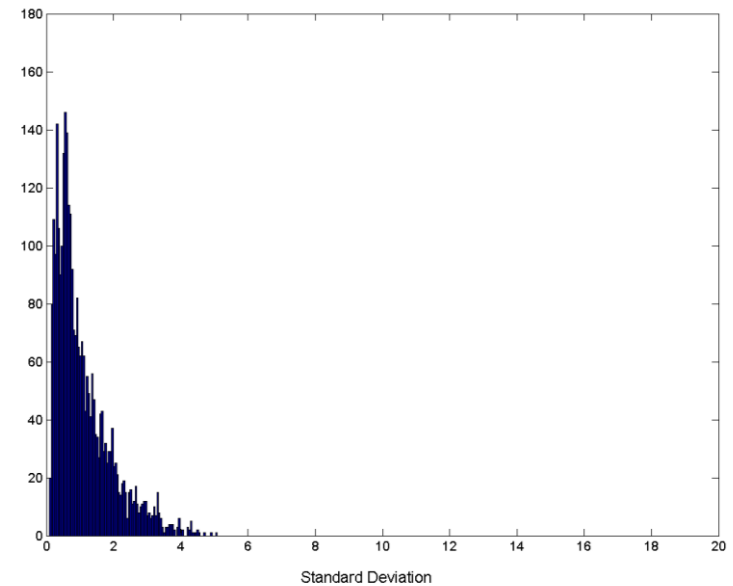
- Data reduction/dimension reduction
 - Reduce the number of attributes or objects
- Change of scale
 - Cities aggregated into regions, states, countries, etc
- More “stable” data
 - Aggregated data tends to have less variability

Aggregation

Variation of Precipitation in Australia



Standard Deviation of
Average Monthly
Precipitation



Standard Deviation of
Average Yearly Precipitation

Sampling

Sometimes (often) it is not possible to obtain all the data for a population. When this happens, you need a **sample** of data.

Sometimes it is not possible to run a data mining algorithm on all the data. When this happens, you need a sample of data.

How do we choose a representative sample?

Sampling cont.

We need to choose a **representative** subset of the data

A Random sample:

- Any sample where each member of the population has a calculable, non-zero chance of selection.

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

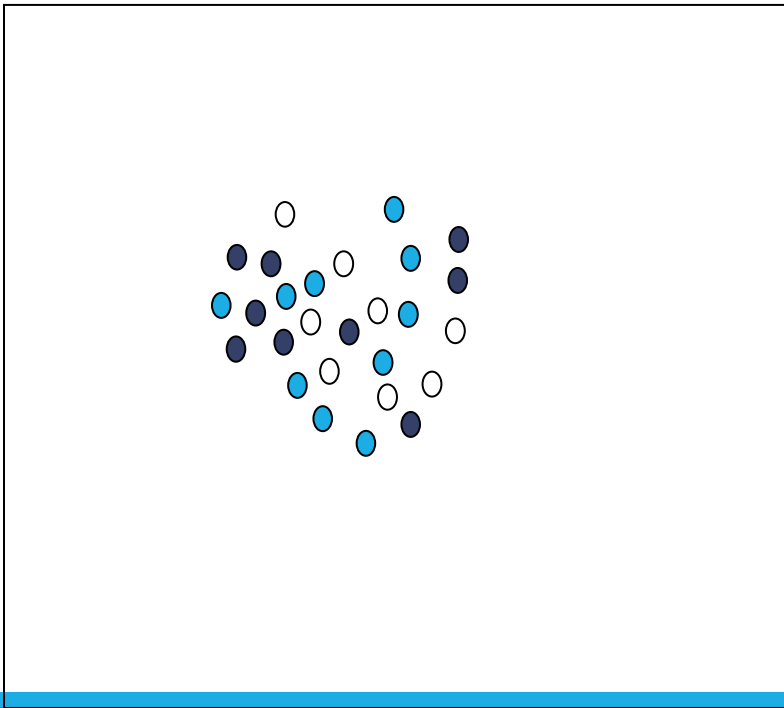
Stratified sampling

- Split the data into several **partitions**; *then draw random samples from each partition*

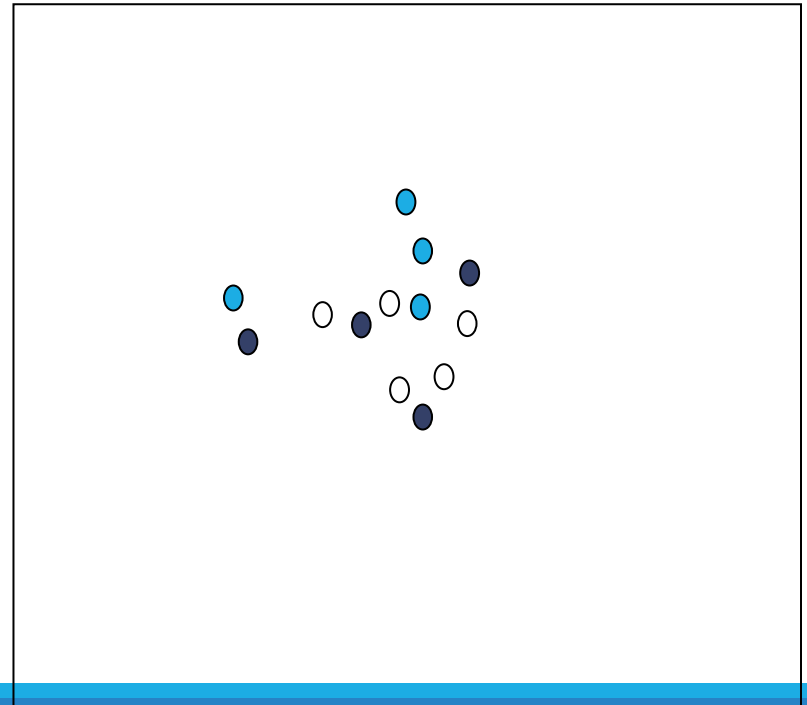
Sampling example - random

Simple random sampling may have
very poor performance in the
presence of skew

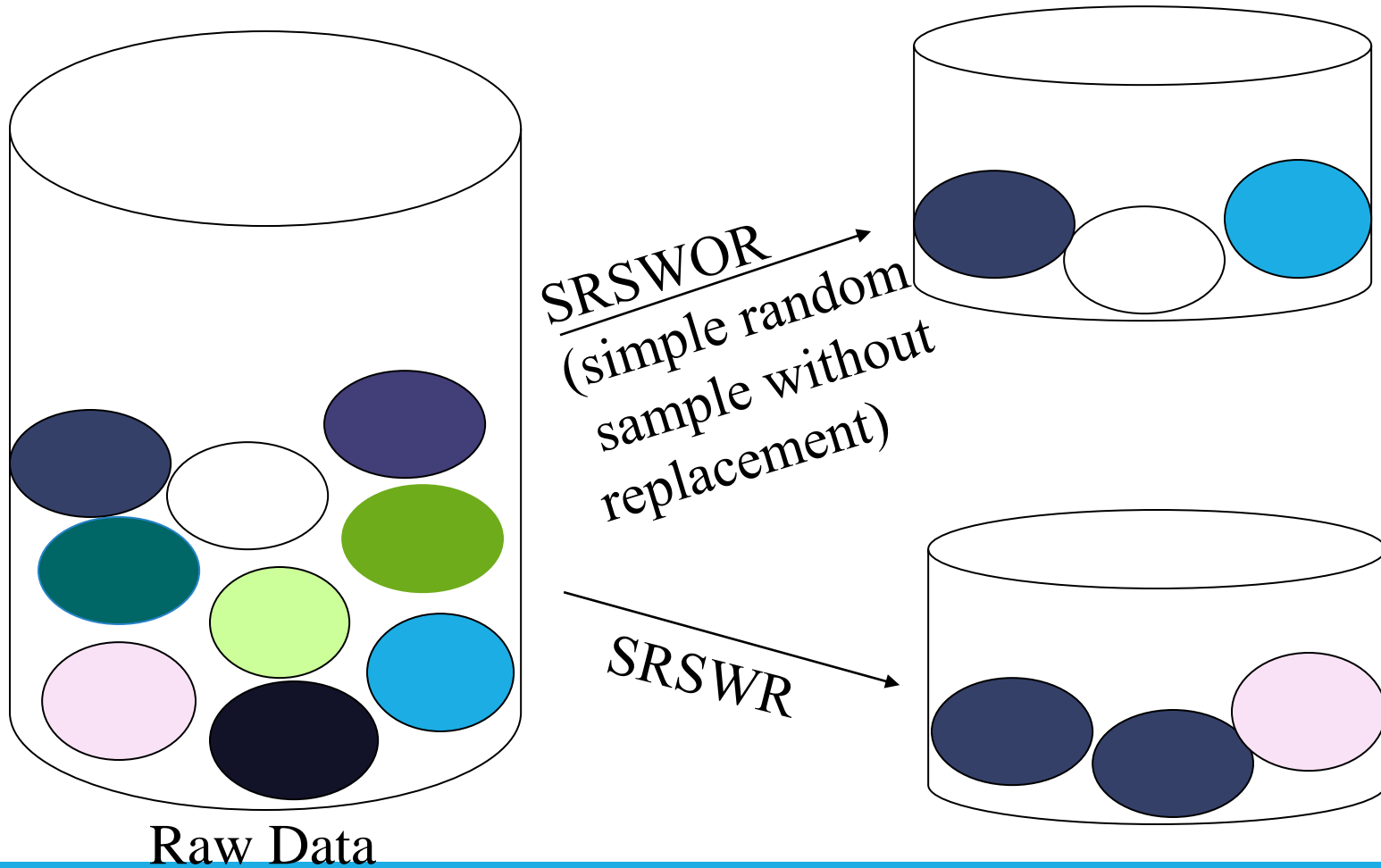
Raw Data



Random Data



Sampling With Replacement



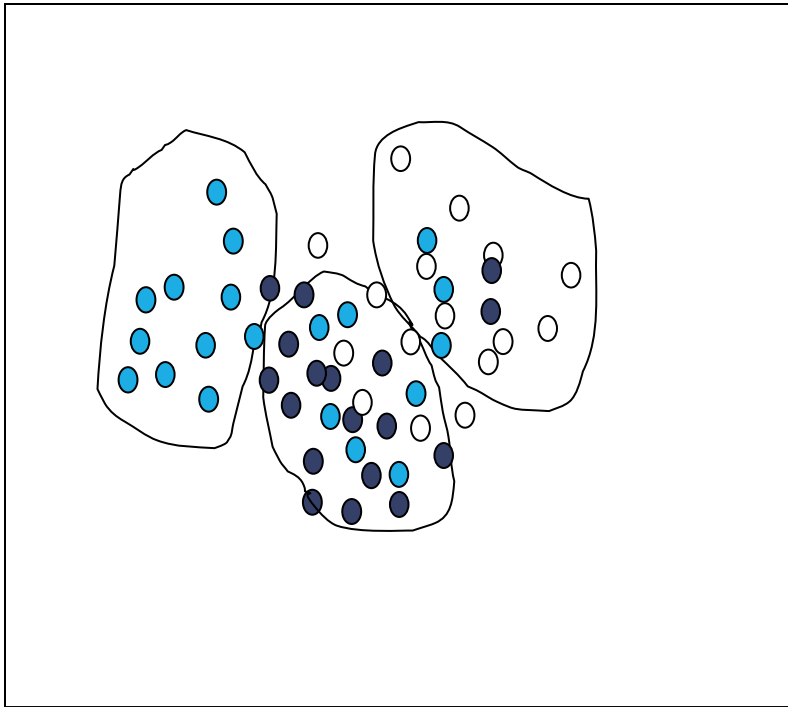
Adaptive Sampling Methods

Stratified sampling:

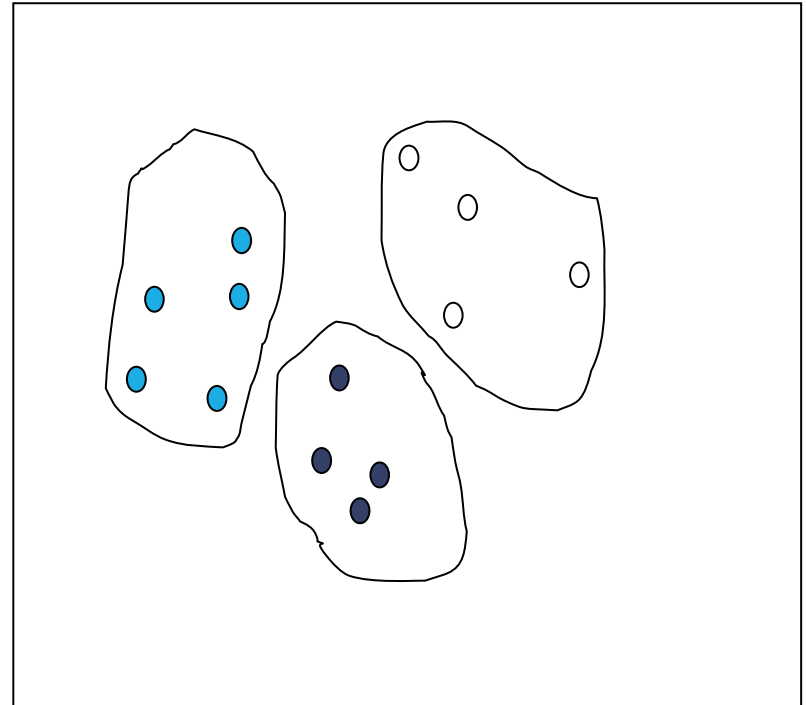
- The population is divided into non-overlapping groups (strata). Samples are drawn from each **strata** separately and the results are pooled together.
- Approximate the percentage of each class (or subpopulation of interest) in the overall database
- Used in conjunction with skewed data

Stratified Sampling

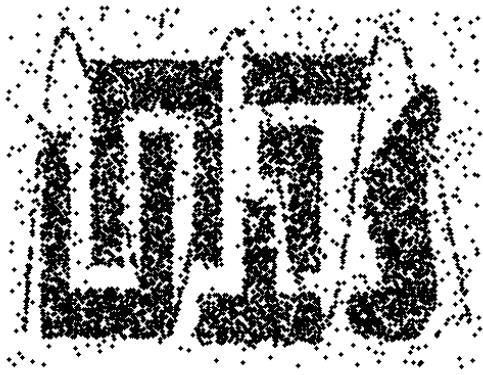
Raw Data



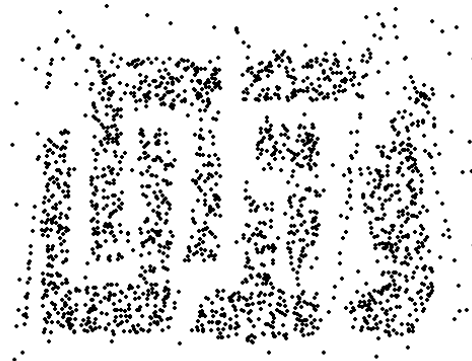
Cluster/Stratified Sample



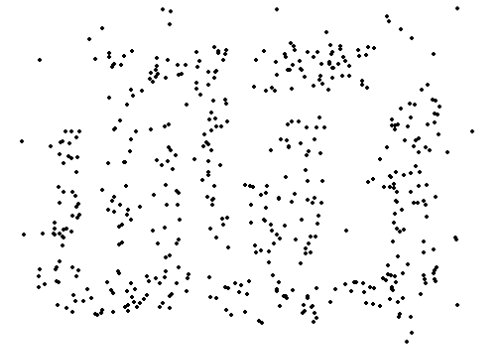
Sample Size Example



8000 points



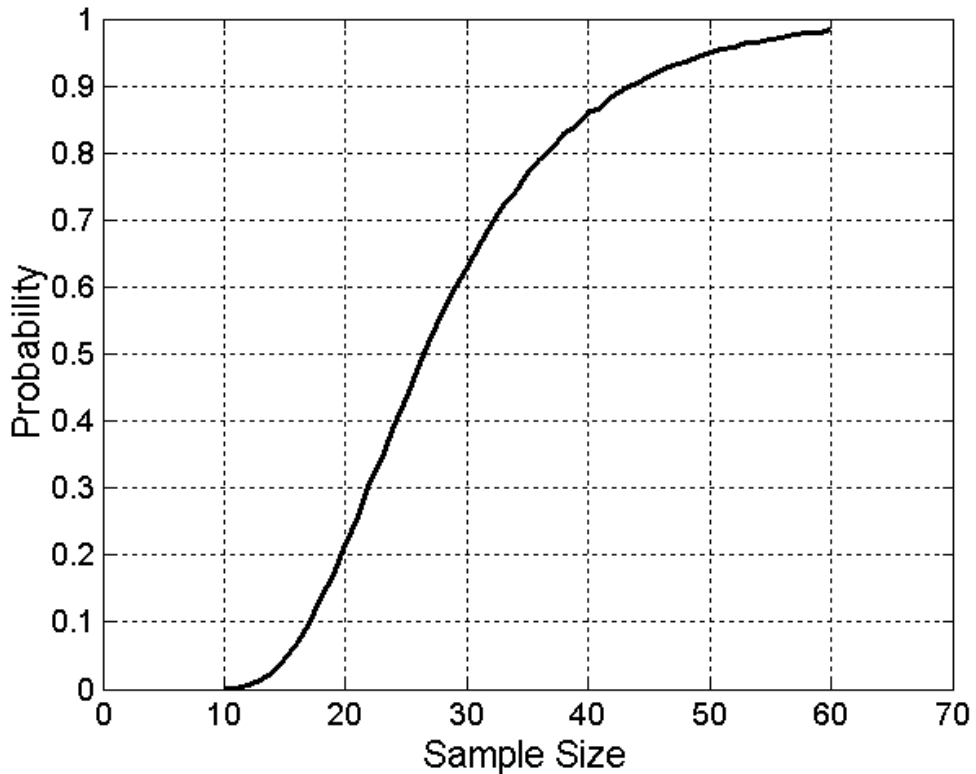
2000 Points



500 Points

Sample Size

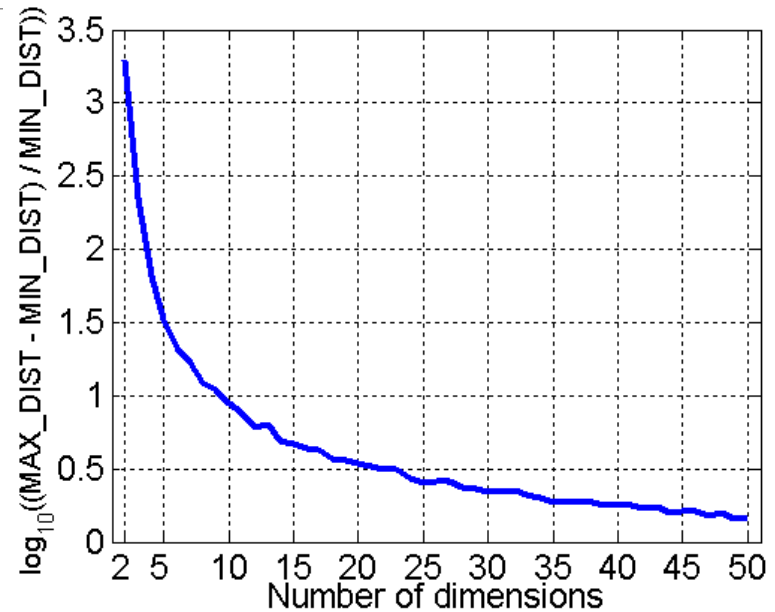
- Suppose you have 10 groups with 10 items in each group
- What sample size is necessary to get at least one object from each of 10 groups (choosing randomly)?



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



- Randomly generate 500 points
- Compute difference between max and min distance between any pair of points

Dimensionality Reduction

Purpose:

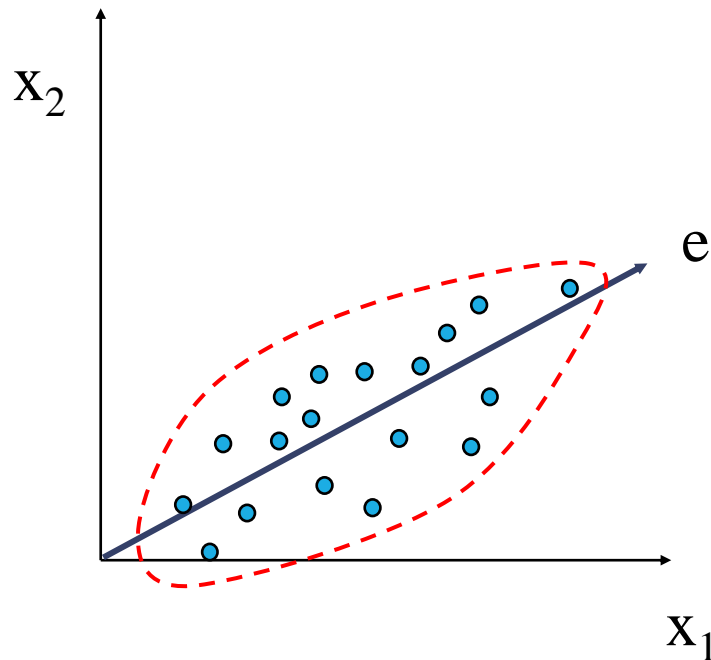
- Avoid curse of dimensionality
- Reduce amount of time and memory required by data mining algorithms
- Allow data to be more easily visualized
- May help to eliminate irrelevant features or reduce noise

Techniques

- Principle Component Analysis (PCA or ICA)
- Singular Value Decomposition (SVD)
- Others: supervised and non-linear techniques
- Removing non- critical variables or parameters by hand.

Dimensionality Reduction: PCA

Goal is to find a projection that captures the largest amount of variation in data. Here – can you reduce 2D to 1D? How? What is lost?

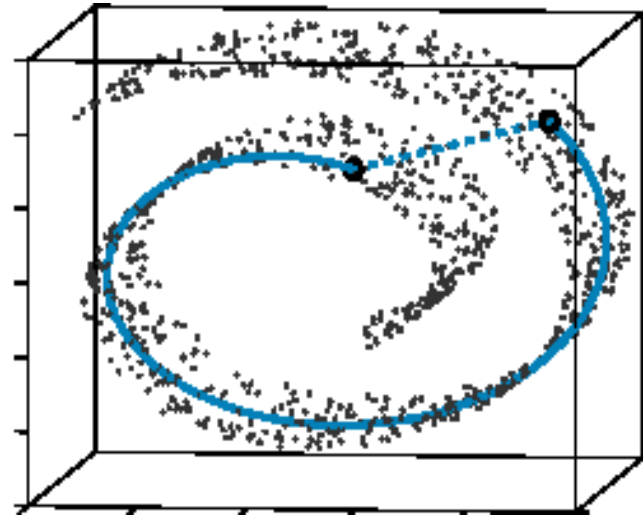


Nonlinear Dimensionality Reduction: ISOMAP

Construct a neighbourhood graph where each point is connected to its k nearest neighbours.

For each pair of points in the graph, compute the shortest path distances – geodesic distances

Use distances to compute a lower dimension embedding



Tenenbaum, de Silva, Langford
(2000)

Feature Subset Selection

Another way to reduce dimensionality of data is to remove features that are redundant or irrelevant.

Doing so also reduces overfitting, reduces training time for machine learning algorithms, and may improve accuracy of algorithms.

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

Approaches for Feature Subset Selection

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. Which features contribute most to high accuracy of algorithm.
- Use regularization methods, e.g. LASSO

Filter approaches:

- Features are selected before data mining algorithm is run by applying a statistical measure and assigning a score to each feature.
- Methods: Chi squared test, information gain

Wrapper approaches:

- Use the data mining algorithm as a black box to find best subset of attributes
- Consider feature selection a search problem in which different combinations are evaluated and compared to each other.
- Example: Recursive feature elimination algorithms

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
 - Transform data to a different space, e.g. Fourier transform
- Feature Construction
 - combining features

Feature Extraction Example

Suppose you are building a classifier that determines how many syllables are in a word.

What **features** would be useful to extract/create?

Here are a few possibilities:

- 1) the length of the word: (WordLen)
- 2) type of character of the first letter (FChar)
- 3) number of vowels in the word (Vcount)

etc.

Pandas introduction

GATES AND SINGH

Introduction to Python 3 pandas

Python pandas (<http://pandas.pydata.org/>) is an open source library that offers excellent data structures, such as the pandas **dataframe**, as well as a number of analysis tools.

The pandas library is installed with Anaconda and can be used by including the following import statement:

```
import pandas as pd
```

pandas: Series

```
import numpy as np
import pandas as pd

# Create an array from 0 to 4
myData=np.arange(5)

# Note the index (row) value names
indexValue=["C1", "C2", "C3", "C4", "C5"]
mySeries=pd.Series(myData, index=indexValue)
print(mySeries)
```

The output:

C1	0
C2	1
C3	2
C4	3
C5	4

pandas: Series and Dictionaries

```
myDict={"Name":"Bob", "Age":29,  
        "Degree":"MS"}
```

```
print(pd.Series(myDict))
```

The Output:

```
Age      29  
Degree   MS  
Name     Bob
```

pandas: Series

```
myDict2={"Grade1":90.1, "Grade2":88.5,  
"Grade3":93.6}
```

```
mySeries=pd.Series(myDict2)
```

```
print(mySeries)
```

```
print("Grade 2 is: ", mySeries[1])
```

```
print("The mean of the grades:",  
mySeries.mean())
```

```
print("Grades plus 5 points added  
is:\n", mySeries+5)
```

```
print("Grade 1 is: ",  
mySeries.get("Grade1"))
```

The Output:

```
Grade1    90.1  
Grade2    88.5  
Grade3    93.6
```

```
Grade 2 is: 88.5
```

```
The mean of the  
grades: 90.73
```

```
Grades plus 5  
points added is:
```

```
Grade1    95.1  
Grade2    93.5  
Grade3    98.6
```

```
Grade 1 is: 90.1
```

pandas: DataFrame

```
import pandas as pd

gradebook={"Student1": pd.Series([89.3, 78.7,
92.2], index=['Grade1', 'Grade2', 'Grade3']),
          "Student2": pd.Series([77.3, 83.4,
91.8], index=['Grade1', 'Grade2', 'Grade3']),
          "Student3": pd.Series([97.1, 88.6,
98.5], index=['Grade1', 'Grade2', 'Grade3'])
        }

gradeBookDF=pd.DataFrame(gradebook)

print(gradeBookDF)
```

Output: Data Frame

	Student1	Student2	Student3
Grade1	89.3	77.3	97.1
Grade2	78.7	83.4	88.6
Grade3	92.2	91.8	98.5

pandas DF: Create Empty DF and add value

```
#Create an empty dataframe
```

```
Gradebook2 = pd.DataFrame(Gradebook, index=['G1', 'G2', 'G3'],  
columns=['Bob Smith', 'Sandy Stern'])
```

```
print(Gradebook2)
```

```
#Fill in values
```

```
Gradebook2.ix["G1", "Bob Smith"]=98.1
```

```
print(Gradebook2)
```

The Output:

	Bob Smith	Sandy Stern
G1	NaN	NaN
G2	NaN	NaN
G3	NaN	NaN

	Bob Smith	Sandy Stern
G1	98.1	NaN
G2	NaN	NaN
G3	NaN	NaN

pandas DF: Add New Column

```
#Create an empty dataframe
```

```
Gradebook2 = pd.DataFrame(Gradebook, index=['G1', 'G2', 'G3'], columns=['Bob  
Smith', 'Sandy Stern'])
```

```
print(Gradebook2)
```

```
#Create a new column
```

```
Gradebook2["NewColumn"]="NaN"
```

```
print(Gradebook2)
```

The Output

	Bob Smith	Sandy Stern	NewColumn
G1	NaN	NaN	NaN
G2	NaN	NaN	NaN
G3	NaN	NaN	NaN

pandas DF: Add Values

```
import random
for i in range(len(Gradebook2.BobSmith)):
    Gradebook2.ix[i,"BobSmith"]=random.randint(50,100)
print(Gradebook2)
```

The Output:

	BobSmith	SandyStern	NewColumn
G1	91	NaN	NaN
G2	56	NaN	NaN
G3	63	NaN	NaN

pandas DF: Convert Dict and Add

```
MyDict=[{"Name":"Bob", "Age":29, "Degree":"MS"}, {"Name":"Rob",  
"Age":34, "Degree":"PhD"}]
```

```
DictDF=pd.DataFrame.from_dict(MyDict)
```

```
DictDF.insert(2, 'NewColumn', [20007, 23604])
```

```
print(DictDF)
```

The Output:

	Age	Degree	NewColumn	Name
0	29	MS	20007	Bob
1	34	PhD	23604	Rob

pandas DF: Dropping Rows and Columns

```
MyDict=[{"Name":"Bob", "Age":29, "Degree":"MS"}, {"Name":"Rob", "Age":34, "Degree":"PhD"}]
```

```
DictDF=pd.DataFrame.from_dict(MyDict)
```

```
DictDF.insert(2, 'NewColumn', [20007, 23604])
```

```
#REMOVE the "Degree" column
```

```
DictDF=DictDF.drop("Degree", axis=1)
```

```
#axis=1 is the column, axis=0 is the row
```

```
#Remove the first row (row 0)
```

```
DictDF=DictDF.drop(0)
```

```
print(DictDF)
```

Read CSV to Pandas DF

```
csvFile="MyCSVFile3.csv"
File2=open(csvFile, "w", newline="")
Header=(["FirstName", "Lastname", "Grade1", "Grade2", "Grade3"])
Data1=(["John", "Smith", 90.3, 87.5, 77.2])
Data2=(["Bob", "Benson", 88.8, 77.7, 66.6])
Fwriter=csv.writer(File2)
Fwriter.writerow(Header)
Fwriter.writerow(Data1)
Fwriter.writerow(Data2)
File2.close()
csvDataFrame=pd.read_csv(csvFile)
print(csvDataFrame)
```

The Output:

	FirstName	Lastname	Grade1	Grade2	Grade3
0	John	Smith	90.3	87.5	77.2
1	Bob	Benson	88.8	77.7	66.6

pandas DF: Adding a New Feature PART 1

```
import pandas as pd

import csv

csvFile="MyCSVFile4.csv"

File2=open(csvFile, "w", newline="")

Header=(["FirstName", "Lastname", "Grade1", "Grade2", "Grade3"])

Data1=(["John", "Smith", 90.3, 97.5, 97.2])

Data2=(["Bob", "Benson", 88.8, 77.7, 66.6])

Data3=(["Sally", "Sue", 78.8, 71.7, 76.6])

Data4=(["Annie", "Apple", 58.8, 67.7, 69.6])

Fwriter=csv.writer(File2)

Fwriter.writerow(Header)

for i in [Data1, Data2, Data3, Data4]:

    Fwriter.writerow(i)

File2.close()

csvDataFrame=pd.read_csv(csvFile)
```

pandas DF: Adding a New Feature PART 2

```
csvDataFrame["NewFeature"]="NaN"
```

```
for i in range(len(csvDataFrame.Grade1)):
    Avg=mean([csvDataFrame.ix[i,"Grade1"], csvDataFrame.ix[i,"Grade2"]])
    if Avg > 89.9:
        csvDataFrame.ix[i,"NewFeature"]="A"
    elif 79.9 < Avg < 90:
        csvDataFrame.ix[i,"NewFeature"]="B"
    elif 69.9 < Avg < 80:
        csvDataFrame.ix[i,"NewFeature"]="C"
    else:
        csvDataFrame.ix[i,"NewFeature"]="D"
print(csvDataFrame)
```

OUTPUT

	FirstName	Lastname	Grade1	Grade2	Grade3	NewFeature
0	John	Smith	90.3	97.5	97.2	A
1	Bob	Benson	88.8	77.7	66.6	C
2	Sally	Sue	78.8	81.7	86.6	B
3	Annie	Apple	58.8	67.7	69.6	D