AI6102: Machine Learning Methodologies & Applications

L2: Data & Operations

Sinno Jialin Pan

Nanyang Technological University, Singapore

Homepage: http://www.ntu.edu.sg/home/sinnopan

Outline

- Types of data
- Feature engineering
- Data operations

What is Data?

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
…	…	…	…	…	…
10	M	Student	8k	5k	No

- Data sets are made up of data instances
- A data instance represents an “entity”
- Alterative names of data instances:
  examples, data objects, data points, etc.
- Data instances are described/represented by features that capture the basic properties of a data instance
- Alterative names of features:
  variables, fields, dimensions, attributes, etc.

Feature Values

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
…	…	…	…	…	…
10	M	Student	8k	5k	No

- Feature values are numbers or symbols assigned to a feature
- Distinction between features and feature values
  - Same feature can be mapped to different feature values
    - Example: height can be measured in feet or meters
  - Different features can be mapped to the same set of values
    - Example: feature values for year and age are integers
    - But properties of feature values can be different
      - Year has no limit but age has a maximum and minimum value

Types of Features

- Categorical
  - Nominal: has no intrinsic ordering to its categories
    - Examples: ID numbers, color, zip codes
  - Ordinal: has a clear ordering
    - Examples: grades in {A, B, C, F}, height in {tall, medium, short}
- Numerical
  - The differences between values are interpretable
    - Examples: length, time, counts

Properties of Feature Values

- The type of an feature depends on which of the following properties (operations) it possesses:

1)	Distinctness:	= and ≠
2)	Order:	< , ≤ , > and ≥
3)	Addition:	+ and −
4)	Multiplication:	× and /

  - Nominal feature: distinctness
  - Ordinal feature: distinctness & order
  - Numerical feature: distinctness, order, addition, & multiplication

Alternative Categorization

- Distinguished by number of values
- Discrete Feature
  - Has only a finite or countably infinite set of values
  - Examples: zip codes, counts, etc.
  - Often represented as integer variables
- Continuous Feature
  - Has real numbers as feature values
  - Examples: temperature, height, or weight
  - Practically, real values can only be measured and represented using a finite number of digits

Binary Features

- A special case of discrete features
  - Nominal feature with only 2 states (e.g., 0 and 1)
  - Symmetric binary: both outcomes equally important
    - e.g., gender
  - Asymmetric binary: outcomes not equally important.
    - e.g., medical test (positive vs. negative)
    - Convention: assign 1 to most important outcome (e.g., COVID-19 positive)

Types of Data

- Structured data: data that adheres to a pre-defined data model (structure of data)
  - E.g., spreadsheets, transaction records, etc
- Unstructured data: information that neither has a pre-defined data model (structure of data) nor is organized in a pre-defined manner
  - E.g., text, images, sensor readings, etc
- Semi-structured data: a cross between the above two, and a type of structured data, but lacks the strict data model structure
  - E.g., webpages, xml, etc

Some Specific Types of Data

- Record
  - Relational records, Data matrix, Transaction data
- Graph & Network
  - Webpages in WWW, Social networks, Molecular structures
- Order
  - Time series data (video data, real-time financial data, dynamic sensor readings)
  - Sequence data (transaction sequences, DNA sequence)
- Spatial
  - Maps, Sensor networks

Record Data

- Data that consists of a collection of records, each of which consists of a fixed set of features

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
3	M	Teacher	56k	100k	Yes
4	F	Student	12k	15k	Yes
5	M	Lawyer	80k	60k	No
6	M	Lawyer	100k	250k	Yes
7	F	Teacher	70k	34k	Yes
8	M	Engineer	85k	110k	No
9	M	Teacher	90k	250k	Yes
10	M	Student	8k	5k	No

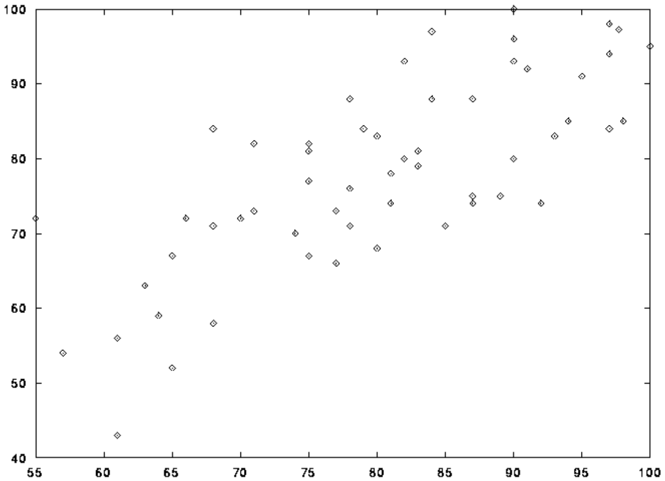
Transaction Data

- A special type of record data, where
  - Each record (transaction) involves a set of items
  - For example, consider a supermarket. The set of products purchased by a customer during one shopping trip constitute a transaction, while the individual products that were purchased are the items

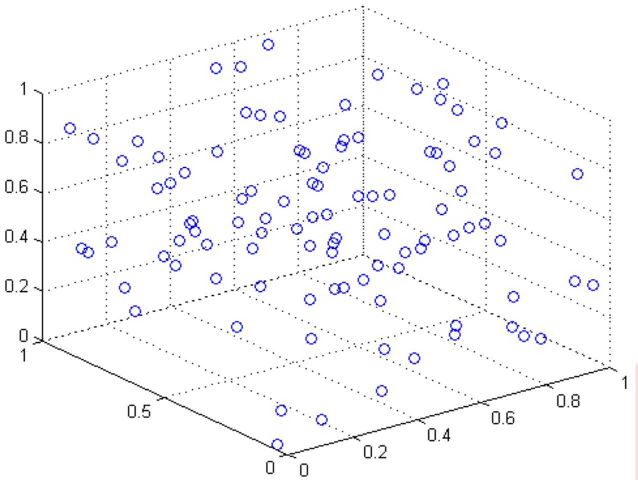
TID	Items
1	Egg, Coke, Milk, Rice, Oil
2	Coke, Bread
3	Rice
4	Milk, Coke, Egg
5	Bread, Egg

Data Matrix

- Data instances have the same fixed set of numerical features
- Each data instance can be thought of as a point in a multi-dimensional space, where each dimension represents a distinct feature



2D



3D

Data Matrix

- Such a dataset can be represented by a $N \times m$ matrix, where there are N rows, one for each data instance, and m columns, one for each feature
  - Or by a $m \times N$ matrix, where each column corresponds a data instance and each row corresponds a feature

ID	Age	Weight	Height
1	25	65	175
2	40	80	178

2×3 matrix

A grayscale image of 28×20 pixels

$$28\left\{\overbrace{\begin{array}{|c|c|c|c|}\hline 0 & 0 & \dots & 87 \\ \hline 12 & 0 & \dots & 79 \\ \hline \dots & \dots & \dots & \dots \\ \hline 255 & 223 & \dots & 0 \\ \hline \end{array}}^{20}\right.$$

0 for black, 255 for white, values in between make up the different shades of gray

Sparse Data Matrix

- A special case of data matrix
- In a recommender system, users' ratings on products can be represented by a sparse matrix or a binary sparse matrix (only like or dislike information is stored)

	Item 1	Item 2	…	Item M
User 1	1	?	5	?
User 2	?	1	?	2
…	…	…	…	…
User N	?	?	4	?

Ratings: 5 > 4 > 3 > 2 > 1 (Ordinal)

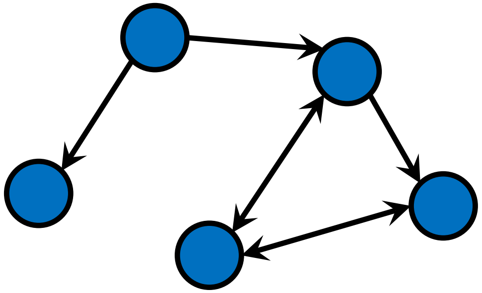
	Item 1	Item 2	…	Item M
User 1	1	?	1	?
User 2	?	0	?	0
…	…	…	…	…
User N	?	?	1	?

1: like, 0: dislike

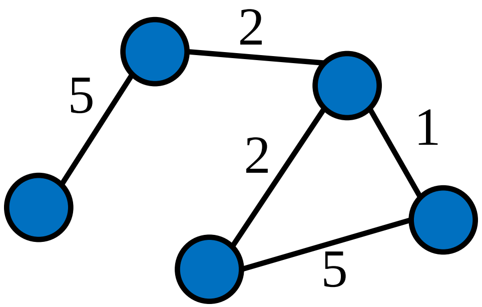
Graph Data

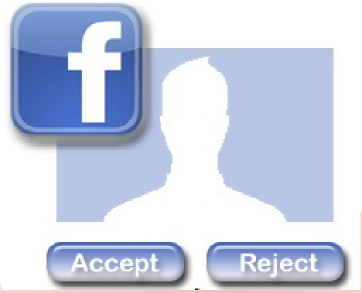
Each data instance is linked to some other data instance(s), and the whole dataset forms a graph

Directed Graph



Undirected Graph

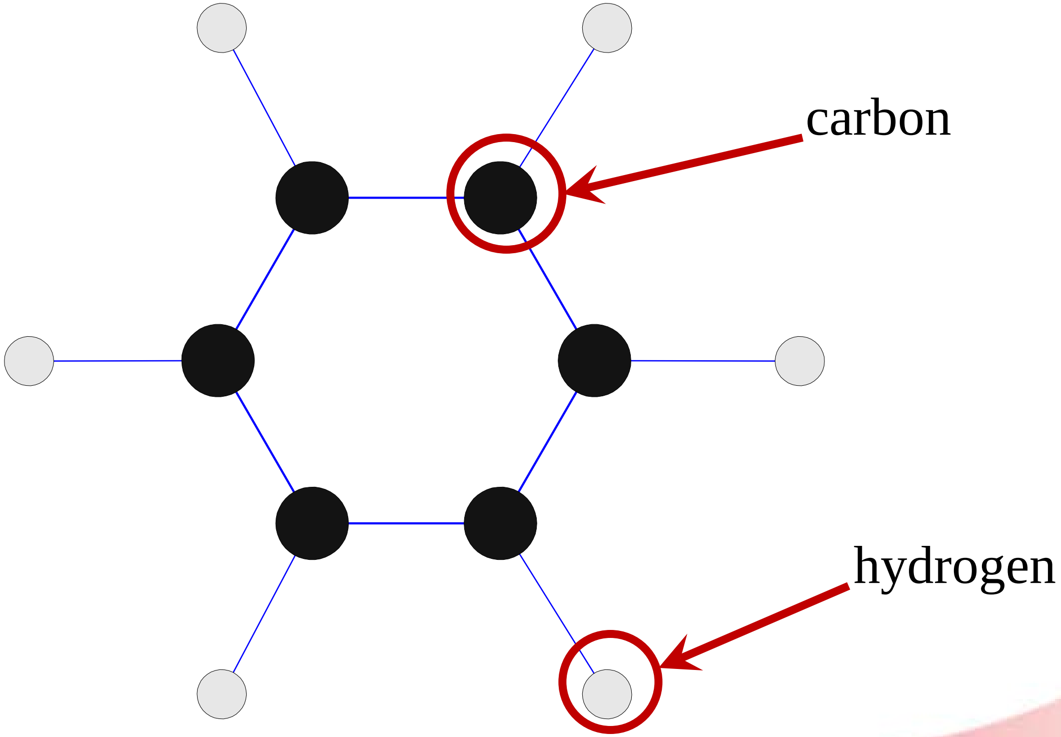




Graph Data (cont.)

Each data instance itself is a graph

Benzene Molecule: C_6H_6



A ball-and-stick diagram of the chemical compound Benzene

Order Data – Sequence

- Sequence transactions

Timeline

Time	Customer	Item Purchased
T1	C1	A, B
T2	C3	A, C
T2	C1	C, D
T3	C2	A, D
T4	C2	E
T5	C1	A, E

Customer	Item Purchased
C1	(T1: A, B) (T2: C, D) (T5: A, E)
C2	(T3: A, D) (T4: E)
C3	(T2: A, C)

A sequence

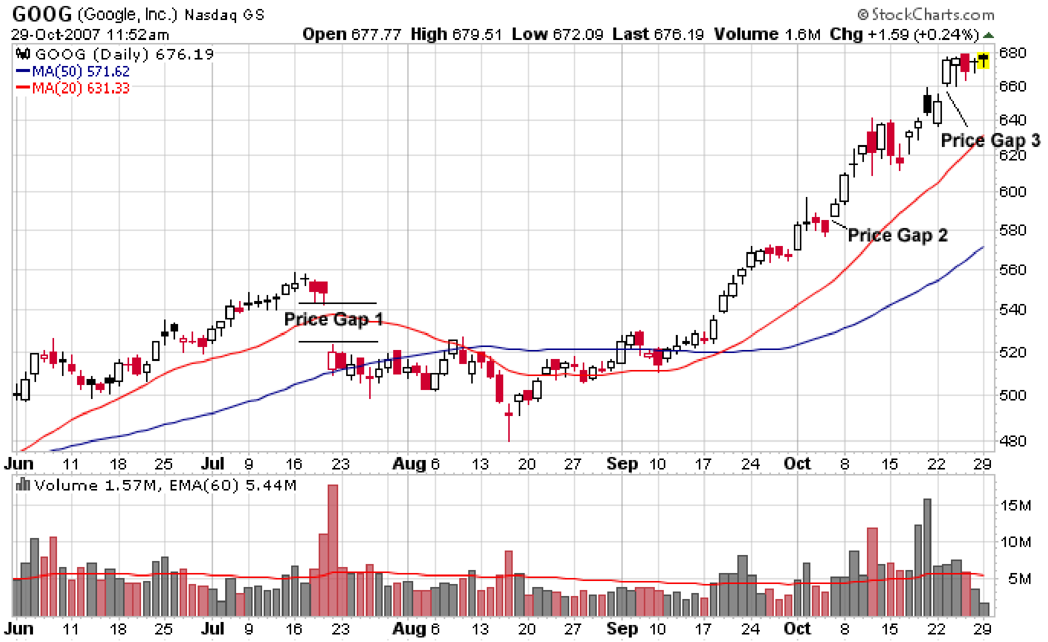
Order Data – Sequence (cont.)

- Genomic sequence data
  - Example: a section of the human genetic code expressed using the four nucleotides from which all DNA is constructed: **A**, **T**, **G**, and **C**

```
GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGGAGGCGGGGCCGCCCGAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCGGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG
```

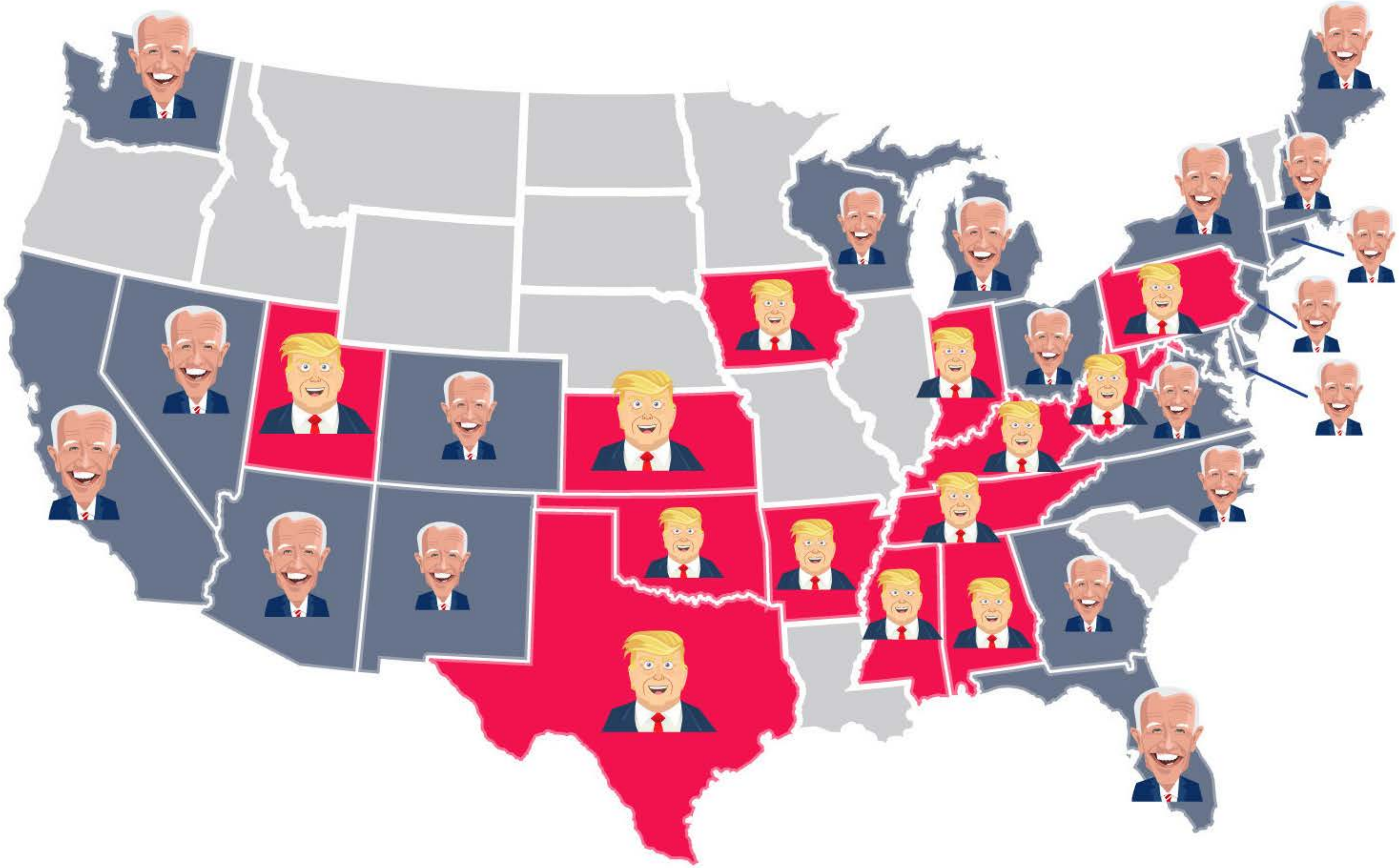
Ordered Data – Time Series

- A special type of sequence data in which each record is a time series, i.e., a series of measurements over (continuous) time.
  - Example: a time series of prices of a stock over days/months/years



Spatial Data

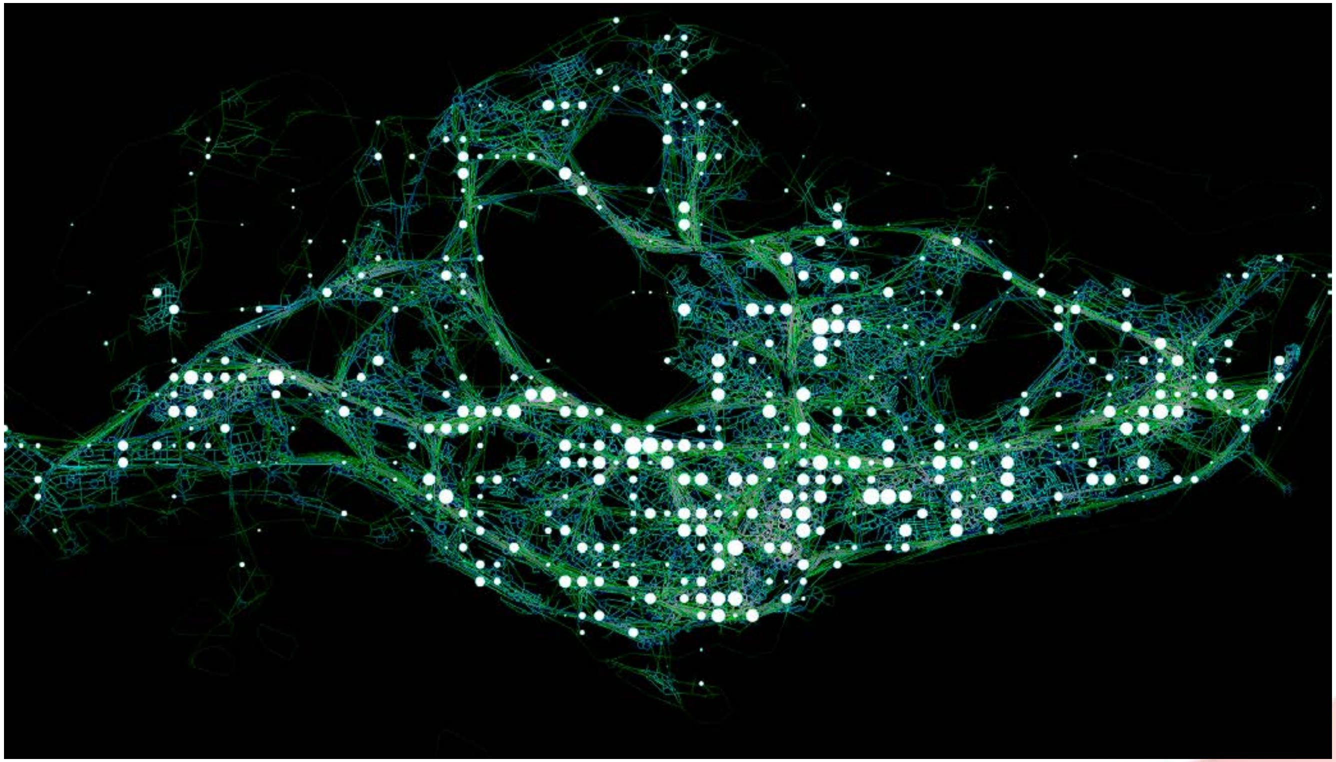
2020 US presidential elections

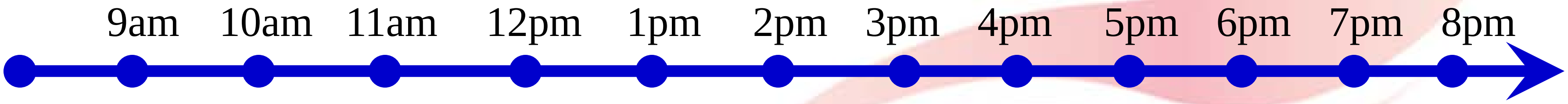


This image is downloaded from https://www.prnewswire.com/

Spatio-Temporal Data

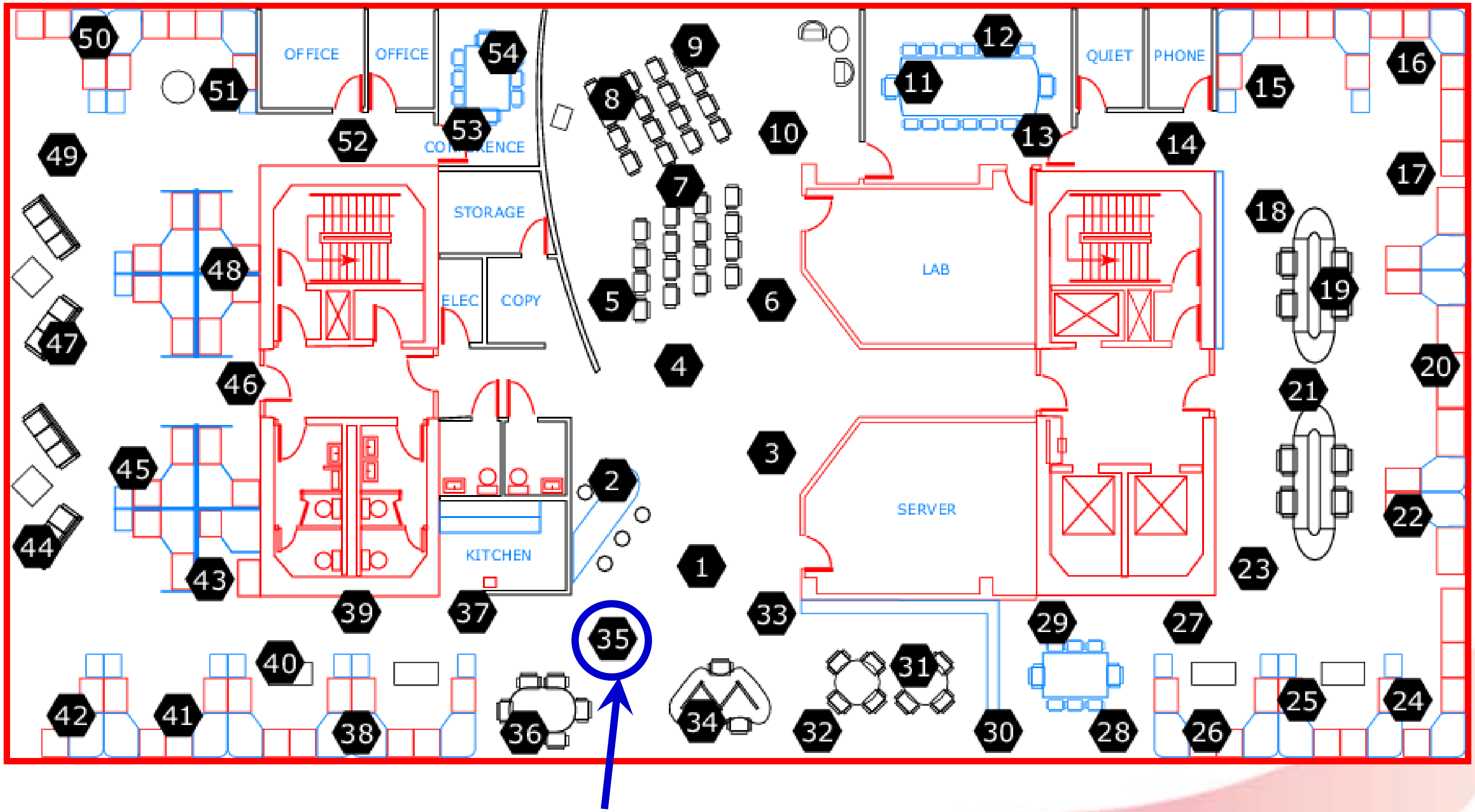
Maps of taxi trajectories over time





9am 10am 11am 12pm 1pm 2pm 3pm 4pm 5pm 6pm 7pm 8pm

Spatio-Temporal Data (cont.)



Sensors to monitor temperature, humidity, light, and voltage

The image is downloaded at http://db.csail.mit.edu/labdata/labdata.html

Outline

- Types of data
- Feature engineering
- Data operations

Feature Engineering

- The process of using domain knowledge and experience to construct features from raw data such that the performance of machine learning algorithms can be improved
- Note: feature engineering is an “engineering” process, and there is no “formula” telling you how to do it
  - Feature cleaning
  - Feature aggregation
  - Feature construction
  - Feature transformation
  - Feature normalization & discretization

Trial and error

Feature Cleaning

- Data in the real world is dirty: lots of potentially incorrect data, e.g., instrument faulty, human or computer error, transmission error
  - Incomplete (missing): lacking features values

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	**N/A**	20k	Yes
…	…	…	…	…	…

  - Noisy: containing noise, errors, or outliers

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	-10k	30k	Yes
2	M	Student	10k	20k	Yes
…	…	…	…	…	…

Dealing with Missing Values

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
~~2~~	~~M~~	~~Student~~	~~N/A~~	~~20k~~	~~Yes~~
…	…	…	…	…	…

- Eliminate the whole data instances
- Not effective when the % of data instances containing missing values is large

ID	Gender	Profession	Income	Saving	Repay
~~1~~	~~F~~	~~Engineer~~	~~N/A~~	~~200k~~	~~Yes~~
~~2~~	~~M~~	~~Student~~	~~N/A~~	~~20k~~	~~Yes~~
~~3~~	~~M~~	~~N/A~~	~~56k~~	~~100k~~	~~Yes~~
4	F	Student	12k	15k	Yes
5	M	Lawyer	80k	60k	No

Dealing with Missing Values (cont.)

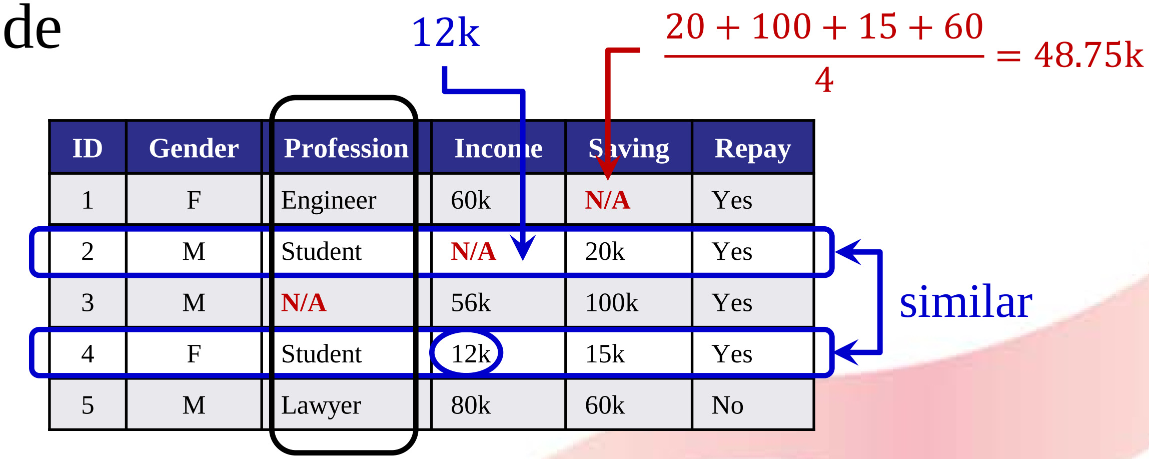
ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	N/A	20k	Yes
…	…	…	…	…	…

- Eliminate the feature that consists missing values
- Not effective when the % of features containing missing values is large
- Not effective when the features containing missing values are important to the machine learning task

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	N/A	Yes
2	M	Student	N/A	20k	Yes
3	M	N/A	56k	100k	Yes
4	F	Student	12k	15k	Yes
5	M	Lawyer	80k	60k	No

Dealing with Missing Values (cont.)

- Estimate missing values
  - Fill in the missing value manually based on prior knowledge
  - Fill in the missing value automatically
    - the feature mean/median
    - the value of other similar data objects
    - the mode



12k

$$\frac{20 + 100 + 15 + 60}{4} = 48.75k$$

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	N/A	Yes
2	M	Student	N/A	20k	Yes
3	M	N/A	56k	100k	Yes
4	F	Student	12k	15k	Yes
5	M	Lawyer	80k	60k	No

similar

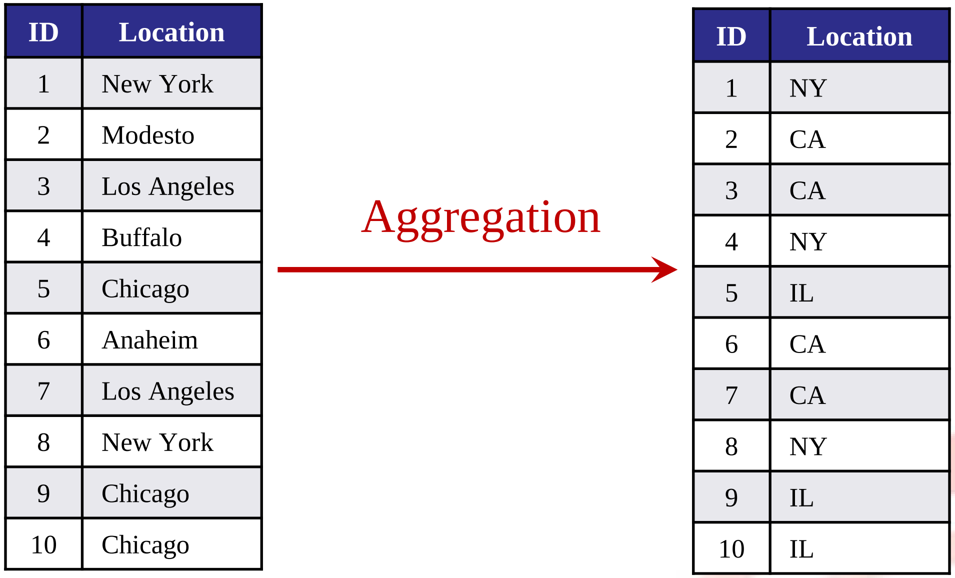
The mode: Student

Dealing with Noisy Values

- Define some rules, e.g., if the value is > the reasonably maximal value, then set it to be the reasonably maximal value
- Similar approaches as dealing with missing values
  - Eliminate the whole data instances
  - Eliminate the features that consists missing values
  - Estimate missing values

Feature Aggregation

- Combining two or more features or feature values into a single feature or feature value
- Example 1: For a feature “Location”, the dataset originally stores “cities”
  - There are a huge amount of distinct values (cities), and a lot of them may only appear one or two time(s)
  - Rescale (aggregation) the values to states, provinces or countries



ID	Location
1	New York
2	Modesto
3	Los Angeles
4	Buffalo
5	Chicago
6	Anaheim
7	Los Angeles
8	New York
9	Chicago
10	Chicago

ID	Location
1	NY
2	CA
3	CA
4	NY
5	IL
6	CA
7	CA
8	NY
9	IL
10	IL

Feature Aggregation (cont.)

- Example 2: Stock price over time
  - To analyze more coarse-grained patterns, the “hour price” features can be aggregated to “day price”, “month price” or “year price”

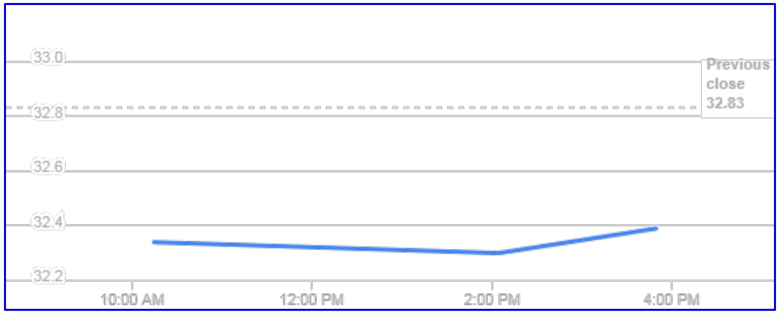
Stock ID	Jul 1 10am	Jul 1 11am	…	Jul 1 4pm	Jul 2 10am	…	Aug 1 10am	…	Sept 1 10am	…	Oct 1 10am	…	Dec 31 4pm
1001	10.5	10.8	…	10.6	10.7	…	8.5	…	11.6	…	13.5	…	12.7
1050	46.3	50.2	…	49.3	48.5	…	55.6	…	54.6	…	54.1	…	59.6
…	…	…	…	…	…	…	…	…	…	…	…	…	…
2055	101.2	99.5	…	100.6	100.1	…	97.3	…	94.5	…	88.2	…	85.6

Aggregation

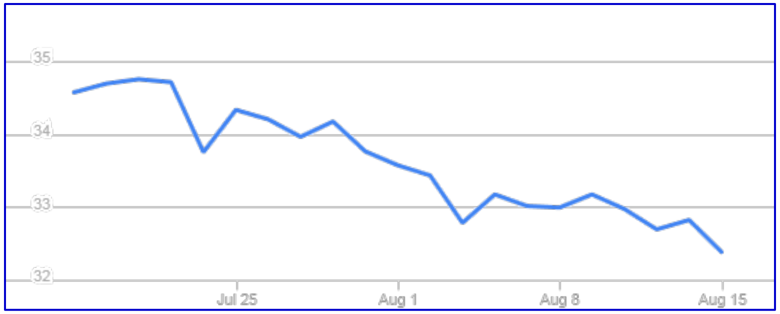
Stock ID	Jul	Aug	Sept	Oct	Nov	Dec
1001	10.6	9.4	11.4	13.4	13.1	12.6
1050	48.2	54.8	53.7	53.1	57.9	59.3
…	…	…	…	…	…	…
2055	100.1	98.5	94.9	87.6	89.7	84.9

Feature Aggregation (cont.)

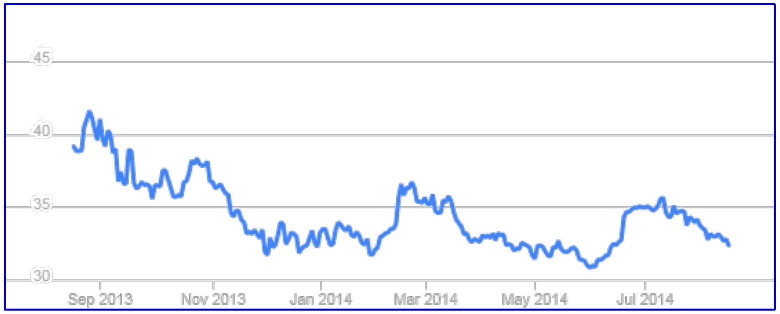
- Example 2: Stock price over time



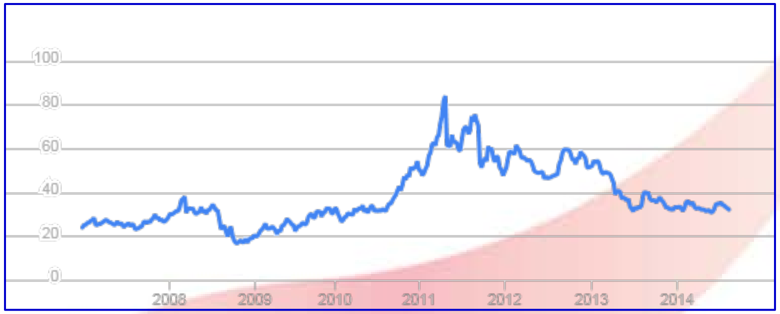
Over 1 day (unit: hour)



Over 1 month (unit: day)



Over 1 year (unit: month)



Over 20+ years (unit: year)

Features Construction

- To create new features to capture more important information of the data than the original features for a specific task

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
…	…	…	…	…	…
10	M	Student	8k	5k	No

→

income-saving ratio

ID	Gender	Profession	Income	Saving	I:S Ratio	Repay
1	F	Engineer	60k	200k	3/10	Yes
2	M	Student	10k	20k	1/2	Yes
…	…	…	…	…	…	…
10	M	Student	8k	5k	8/5	No

$$\text{BMI} = \frac{\text{weight (kg)}}{\text{height (m)}^2}$$

ID	Age	Weight	Height	…	Healthy
1	25	65	175	…	Yes
2	40	80	178	…	No
…	…	…	…	…	…

→

ID	Age	Weight	Height	BMI	…	Healthy
1	25	130	175	21.22	…	Yes
2	40	160	178	25.24	…	No
…	…	…	…		…	…

Features Construction (cont.)

ID	Expiry Date
1	13/08/2020
2	20/04/2018
…	…
10	04/07/2022

ID	Expiry Date Day	Expiry Date Month	Expiry Date Year
1	13	8	2020
2	20	4	2018
…	…	…	…
10	4	7	2022

Using current date information

ID	Expiry Date Day	Expiry Date Month	Expiry Date Year	Expired?	# Expired days
1	13	8	2020	Yes	10
2	13	8	2018	Yes	740
…	…	…	…	…	…
10	4	7	2022	No	0

Feature Transformation

- For most supervised learning algorithms, each input data instance needs to be represented by a numerical vector $\boldsymbol{x}_i$ of a fixed dimension (e.g., m)
- Categorical features → one-hot encoding
- Unstructured data → feature vector

One-hot Encoding

- Transform a feature of *k* distinct categorical values to *k* numerical features of binary values (0/1)

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
3	M	Teacher	56k	100k	Yes
4	F	Student	12k	15k	Yes
5	M	Lawyer	80k	60k	No
6	M	Lawyer	100k	250k	Yes
7	F	Teacher	70k	34k	Yes
8	M	Engineer	85k	110k	No
9	M	Teacher	90k	250k	Yes
10	M	Student	8k	5k	No

Engineer	Student	Teacher	Lawyer
1	0	0	0
0	1	0	0
0	0	1	0
0	1	0	0
0	0	0	1
0	0	0	1
0	0	1	0
1	0	0	0
0	0	1	0
0	1	0	0

Why One-hot Encoding?

Engineer:	1
Student:	2
Teacher:	3
Lawyer:	4

✗

Numerical values

ID	Profession
1	Engineer
2	Student
3	Teacher
5	Lawyer

ID	Profession
1	1
2	2
3	3
5	4

- Distance between IDs 1 & 2 (Engineer v.s. Student): 1
- Distance between IDs 1 & 5 (Engineer v.s. Lawyer): 3
- Each distinct values should be equally important
- The distance between them should be the same after transformation

Using one-hot encoding

ID	Engineer	Student	Teacher	Lawyer
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
5	0	0	0	1

Distances between IDs 1, 2, 3 and 5 are all $\sqrt{2}$

Binary Features

Unnecessary

Using one-hot encoding?

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
3	M	Teacher	56k	100k	Yes
4	F	Student	12k	15k	Yes
5	M	Lawyer	80k	60k	No
6	M	Lawyer	100k	250k	Yes
7	F	Teacher	70k	34k	Yes
8	M	Engineer	85k	110k	No
9	M	Teacher	90k	250k	Yes
10	M	Student	8k	5k	No

Female	Male
1	0
0	1
0	0
0	1
0	0
0	0
0	0
1	0
0	0
0	1

Gender
1
0
0
1
0
0
1
0
0
0

Female:	1
Male:	0

Distance between two same categories is 0
Distance between two distinct categories is 1

Extension of One-hot Encoding

Each distinct item over all the transactions
is used to construct a binary feature

TID	Items
1	Egg, Coke, Milk, Rice, Oil
2	Coke, Bread
3	Rice
4	Milk, Coke, Egg
5	Bread, Egg

ID	Bread	Coke	Egg	Milk	Oil	Rice
1	0	1	1	1	1	1
2	1	1	0	0	0	0
3	0	0	0	0	0	1
4	0	1	1	1	0	0
5	1	0	1	0	0	0

Unstructured Data – Text

Doc1	Compact; easy to operate; very good picture quality; looks sharp!
Doc2	It is also quite blurry in very dark settings. I will never_buy HP again.
…	…

Scan through the whole training dataset once to build a dictionary

F1	F2	F3	F4	F5	F6	…
compact	easy	quite	blurry	good	never_buy	…

	F1	F2	F3	F4	F5	F6	…
Doc1	1	1	0	0	1	0	…
Doc2	0	0	1	1	0	1	…
…	…	…	…	…	…	…	…

Bag-of-words representation

Unstructured Data – Images

Grayscale image

20×30

0	0	…	87
12	0	…	79
…	…	…	…
255	223	…	0

20×30

Concatenating rows to construct a single vector of $20 \times 30 = 600$ dimensions

- Use image processing algorithms to detect and isolate various desired portions or shapes (features) of an image
  - The scale-invariant feature transform (SIFT) is a feature detection algorithm to detect and describe local features in images
  - SIFT keypoints of objects are first extracted from the training dataset to construct a visual “words” dictionary
  - Bag-of-(visual)-words representation is used to represent each image

Feature Normalization & Discretization

- Normalization
  - A function that maps the entire set of values of a given feature to a smaller and specified-range new set of replacement values such that each old value can be identified with one of the new values
    - Min-max normalization
    - Standardization (*z*-score normalization)
- Discretization
  - Divide the range of a continuous features into intervals

Min-Max Normalization

- To rescale values to $[\min_{new}, \max_{new}]$
  - e.g. to normalize saving ranging from 5k to 250k to [0.0, 1.0]. What is the value for 100k after normalization?

ID	Saving
1	200k
2	20k
3	100k
4	15k
5	60k
6	250k
7	34k
8	110k
9	250k
10	5k

$$v_{new} = \frac{v_{old} - \min_{old}}{\max_{old} - \min_{old}}(\max_{new} - \min_{new}) + \min_{new}$$

$$100\text{k} \longrightarrow \frac{100\text{k} - 5\text{k}}{250\text{k} - 5\text{k}}(1.0 - 0) + 0 = 0.388$$

Standardization

- Also known as *z*-score normalization, rescale values such that the mean of new values is 0, and the standard deviation is 1 (μ: mean, σ: standard deviation)
  - e.g. the mean of saving is $\mu = 104.4$k, and the standard deviation of saving $\sigma = 91.38$k. What is the value for 100k after standardization?

ID	Saving
1	200k
2	20k
3	100k
4	15k
5	60k
6	250k
7	34k
8	110k
9	250k
10	5k

$$v_{new} = \frac{v_{old} - \mu_{old}}{\sigma_{old}} \Rightarrow \mu_{new} = 0, \text{and } \sigma_{new} = 1$$

$$\frac{100 - 104.4}{91.38} = -0.05$$

Discretization

- Some classification algorithm do not prefer continuous features (potentially a lot of distinct values)
- Solution: to discretize values of a continuous feature into intervals, interval “labels” are used to replace values
  - Binning
  - Binarization

Binning: Equal-frequency

- Divides the range into *K* intervals, each containing approximately same number of data

ID	F1
1	4
2	34
3	9
4	21
5	8
6	26
7	29
8	10
9	25
10	24
11	28
12	21

Divide into 3 intervals →

ID	F1
1	1
2	3
3	1
4	2
5	1
6	3
7	3
8	1
9	2
10	2
11	3
12	2

or

ID	F1
1	7.75
2	29.25
3	7.75
4	22.75
5	7.75
6	29.25
7	29.25
8	7.75
9	22.75
10	22.75
11	29.25
12	22.75

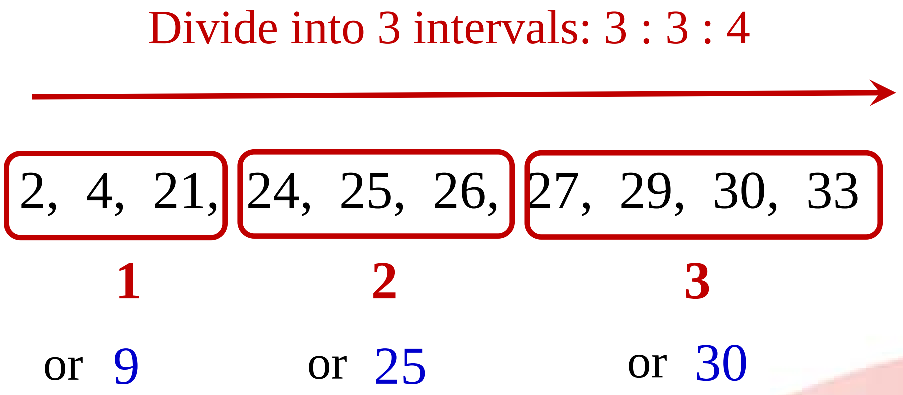
Sorted: 4, 8, 9, 10, | 21, 21, 24, 25, | 26, 28, 29, 34

1 or 7.75 | 2 or 22.75 | 3 or 29.25

Binning: Equal-frequency (cont.)

- Advantage
  - Data sizes of each interval are balanced
- Disadvantage
  - Variance of values in some interval(s) could be very large

ID	F1
1	2
2	4
3	27
4	21
5	30
6	26
7	30
8	33
9	25
10	24



Divide into 3 intervals: 3 : 3 : 4

2, 4, 21, | 24, 25, 26, | 27, 29, 30, 33

1 | 2 | 3

or 9 | or 25 | or 30

ID	F1
1	1
2	1
3	3
4	1
5	3
6	2
7	3
8	3
9	2
10	2

or

ID	F1
1	9
2	9
3	30
4	9
5	30
6	25
7	30
8	30
9	25
10	25

Binning: Equal-distance

- Divides the range into K intervals of equal size: uniform grid
- Denote by Max and Min the lowest and highest values of the feature, the width of intervals will be $\triangle = \frac{\text{Max} - \text{Min}}{K}$

ID	F1
1	4
2	34
3	9
4	21
5	8
6	26
7	29
8	10
9	25
10	24
11	28
12	21

$$\xrightarrow{\text{Divide into 3 intervals}}$$

$$\triangle = \frac{34 - 4}{3} = 10$$

$[4\ ,14),\qquad [14, 24),\qquad [24{,}34]$

$$\overset{\mathbf{1}}{\boxed{4,\ 8,\ 9,\ 10,}}\ \overset{\mathbf{2}}{\boxed{21,\ 21,}}\ \overset{\mathbf{3}}{\boxed{24,\ 25,\ 26,\ 28,\ 29,\ 34}}$$

ID	F1
1	1
2	3
3	1
4	2
5	1
6	3
7	3
8	1
9	2
10	2
11	3
12	2

Binning: Equal-distance

- Advantage
  - The most straightforward, but outliers may dominate
- Disadvantage
  - The instance sizes of each interval would be highly imbalanced on skewed dataset

Divide into 3 intervals

$$\triangle = \frac{29-2}{3} = 9$$

$[2\,,11),\qquad [11,20),\qquad [20,29]$

$$\underset{\mathbf{1}}{\boxed{2,}}\ \underset{\mathbf{2}}{\boxed{12,}}\ \underset{\mathbf{3}}{\boxed{21,\ 24,\ 25,\ 25,\ 27,\ 28,\ 28,\ 29,\ 29}}$$

Binarization

- A special case of discretization
- To transform each numerical value of an feature to one of the binary values
- Set a threshold value T, if the feature value $\geq T$, then it is mapped to 1, otherwise, 0

ID	Saving
1	200k
2	20k
3	100k
4	15k
5	60k
6	250k
7	34k
8	110k
9	250k
10	5k

$$\xrightarrow{T = 91\text{k}}$$

ID	Saving ≥ 91k?
1	1
2	0
3	1
4	0
5	0
6	1
7	0
8	1
9	1
10	0

Outline

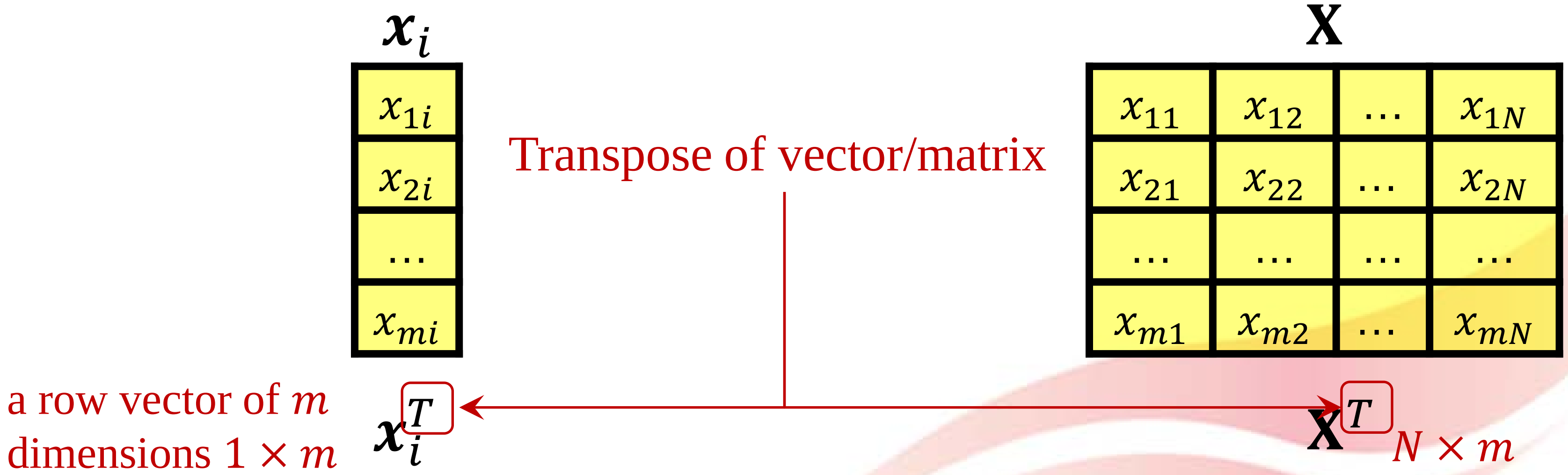
- Types of data
- Feature engineering
- Data operations
  - Proximity
  - Correlation

Proximity

- Distance (Dissimilarity)
  - Numerical measure of how different two data instances are
  - Lower when data instances are more alike
  - Minimum distance is 0
  - Upper limit varies
- Similarity
  - Numerical measure of how alike two data instances are
  - Higher when data instances are more alike
  - Often falls in the range $[0, 1]$

Notations

- For each m-dimensional data instance $\boldsymbol{x}_i$, we represent it by a column vector, i.e, $m \times 1$, where $x_{ki}, k = 1, \ldots, m$ is the value of the k-th feature or dimension of the data instance $\boldsymbol{x}_i$
- Given a dataset of N data instances, each of which is m-dimensional, we represent it by a $m \times N$ matrix $\mathbf{X}$, where x_{ki} indicates the value of the k-th feature of the i-th instance



$$\boldsymbol{x}_i = \begin{bmatrix} x_{1i} \\ x_{2i} \\ \ldots \\ x_{mi} \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1N} \\ x_{21} & x_{22} & \ldots & x_{2N} \\ \ldots & \ldots & \ldots & \ldots \\ x_{m1} & x_{m2} & \ldots & x_{mN} \end{bmatrix}$$

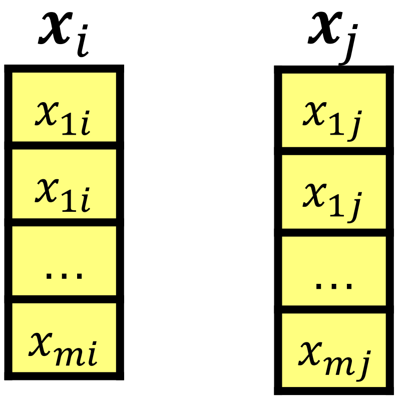
Transpose of vector/matrix

a row vector of m dimensions $1 \times m$ $\boldsymbol{x}_i^{\boxed{T}}$

$\mathbf{X}^{\boxed{T}}$ $N \times m$

Euclidean Distance

- Given two m-dimensional data instances $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, the Euclidean distance between them is defined as



$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{\sum_{k=1}^{m} (x_{ki} - x_{kj})^2}$$

- A more compact form of the Euclidean distance

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j) \odot (\boldsymbol{x}_i - \boldsymbol{x}_j)}$$

Inner product

Inner Product

- Given two m-dimensional data instances $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, the inner product between them is defined as

$$\boldsymbol{x}_i \cdot \boldsymbol{x}_j = \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle = \sum_{k=1}^{m} \left(x_{ki} \times x_{kj} \right) \; = \boldsymbol{x}_i{}^T \boldsymbol{x}_j$$

- The Euclidean distance can be rewritten as

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{\sum_{k=1}^{m} (x_{ki} - x_{kj})^2} = \sqrt{\sum_{k=1}^{m} \left(\left(x_{ki} - x_{kj} \right) \times \left(x_{ki} - x_{kj} \right) \right)}$$

$$= \sqrt{\left(\boldsymbol{x}_i - \boldsymbol{x}_j \right)^T \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right)}$$

$$\boldsymbol{x}_i - \boldsymbol{x}_j \quad \begin{array}{|c|} \hline x_{1i} - x_{1j} \\ \hline x_{2i} - x_{2j} \\ \hline \dots \\ \hline x_{mi} - x_{mj} \\ \hline \end{array}$$

$$= \sqrt{\left(\boldsymbol{x}_i - \boldsymbol{x}_j \right) \cdot \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right)} \quad \text{OR} \quad \sqrt{\langle \boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{x}_i - \boldsymbol{x}_j \rangle}$$

L2 Norm

- The Euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be written as

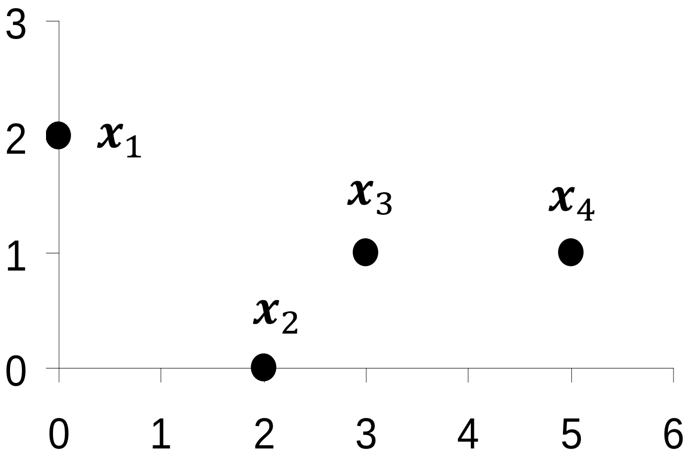
$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boxed{\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2}$$

$\|\boldsymbol{x}\|_2$ is known as the L2 norm of a m-dimensional vector $\boldsymbol{x}$, defined as $\|\boldsymbol{x}\|_2 = \sqrt{\sum_{k=1}^{m} x_k^2}$

$$\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2 = \sqrt{\sum_{k=1}^{m} (x_{ki} - x_{kj})^2}$$

Note: $\|\boldsymbol{x}\|_2$ can be viewed as the measure of Euclidean distance between $\boldsymbol{x}$ and the origin $\mathbf{0}$

An Example



	X_1	X_2
$\boldsymbol{x}_1$	0	2
$\boldsymbol{x}_2$	2	0
$\boldsymbol{x}_3$	3	1
$\boldsymbol{x}_4$	5	1

	$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$
$\boldsymbol{x}_1$	0	2.828	3.162	5.099
$\boldsymbol{x}_2$	2.828	0	1.414	3.162
$\boldsymbol{x}_3$	3.162	1.414	0	2
$\boldsymbol{x}_4$	5.099	3.162	2	0

Distance matrix

Manhattan Distance

- Given two m-dimensional data instances $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, the Manhattan distance between them is defined as

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{k=1}^{m} |x_{ki} - x_{kj}|$$

- The Manhattan distance is also known as the L1-norm distance

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boxed{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_1}$$

$\|\boldsymbol{x}\|_1$ is known as the L1 norm of a m-dimensional vector $\boldsymbol{x}$, defined as $\|\boldsymbol{x}\|_1 = \sum_{k=1}^{m} |x_k|$

An Example

A hash table

Binary bits

	X_1	X_2	X_3
$\boldsymbol{x}_1$	0	1	0
$\boldsymbol{x}_2$	1	0	0
$\boldsymbol{x}_3$	1	1	1
$\boldsymbol{x}_4$	1	1	0

Distance matrix

	$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$
$\boldsymbol{x}_1$	0	2	2	1
$\boldsymbol{x}_2$	2	0	2	1
$\boldsymbol{x}_3$	2	2	0	1
$\boldsymbol{x}_4$	1	1	1	0

A Survey on Learning to Hash, Wang et al., TPAMI 2017

Common Properties of Distances

- Distances have some well known properties:
  - Positive definiteness:
    - $d(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$ for any $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ and $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0$ only if $\boldsymbol{x}_i = \boldsymbol{x}_j$
  - Symmetry:
    - $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = d(\boldsymbol{x}_j, \boldsymbol{x}_i)$ for any $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$
  - Triangle inequality:
    - $d(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq d(\boldsymbol{x}_i, \boldsymbol{x}_k) + d(\boldsymbol{x}_k, \boldsymbol{x}_j)$for any $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$
- A distance that satisfies these properties is a metric

Similarity

- Recall that distance also known as dissimilarity is to measure how different two data instances are, while similarity is to measure how alike two data instances are
- Distance can be simply revised to measure similarity, e.g,

$$s(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{1}{d(\boldsymbol{x}_i, \boldsymbol{x}_j)}$$

where $s(\boldsymbol{x}_i, \boldsymbol{x}_j) \triangleq 1$ when $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0$

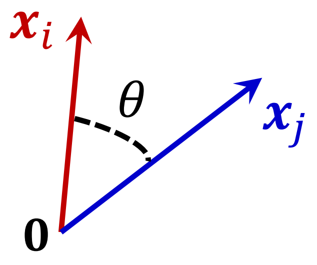
- In this way, for any $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, $s(\boldsymbol{x}_i, \boldsymbol{x}_j) \in (0, 1]$
- Set a threshold T: if $d(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq T$, then $s(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0$

Cosine Similarity

- Given two m-dimensional non-zero data instances $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, the Cosine similarity between them is defined as

$$s(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{\boxed{\boldsymbol{x}_i \cdot \boldsymbol{x}_j}}{\|\boldsymbol{x}_i\|_2 \|\boldsymbol{x}_j\|_2} = \cos(\theta)$$

$$\boldsymbol{x}_i \cdot \boldsymbol{x}_j = \sum_{k=1}^{m} (x_{ki} \times x_{kj}) = \|\boldsymbol{x}_i\|_2 \times \|\boldsymbol{x}_j\|_2 \times \cos(\boxed{\theta})$$



Angle between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$

- The outcome of Cosine similarity is in $[-1, 1]$
- Cosine similarity is particularly used in positive space, i.e., $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are of non-negative numerical values → outcome of Cosine similarity is in $[0, 1]$

Similarity Properties

- Maximum: $s(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1$ only if $\boldsymbol{x}_i = \boldsymbol{x}_j$
- Symmetry: $s(\boldsymbol{x}_i, \boldsymbol{x}_j) = s(\boldsymbol{x}_j, \boldsymbol{x}_i)$ for any $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$
- A general way to change distance to similarity is to define a strictly monotone decreasing function $f(x)$:

$$\text{similarity} = f(\text{distance})$$

- Some commonly used forms of the function $f(x)$ include

$$f(x) = \frac{1}{x+b}, \text{where } b \geq 0 \text{ is a parameter}$$

$$f(x) = e^{-x^b}, \text{where } b > 0 \text{ is a parameter}$$

Feature Correlation

$$\mathbf{X}$$

$$\begin{array}{c|c|c|c|c|}
 & \boldsymbol{x}_1 & \boldsymbol{x}_2 & & \boldsymbol{x}_N \\ \hline
X_1 & x_{11} & x_{12} & \dots & x_{1N} \\ \hline
X_2 & x_{21} & x_{22} & \dots & x_{2N} \\ \hline
 & \dots & \dots & \dots & \dots \\ \hline
X_m & x_{m1} & x_{m2} & \dots & x_{mN} \\ \hline
\end{array}$$

Similarity or distance is to measure the relationship between data instances, i.e., the columns of the data matrix **X**

Feature correlation is to measure the relationship between features e.g, what is the relationship between height and weight?

Given a data matrix **X**, each feature X_i can be represented by the corresponding column of the matrix

Pearson Correlation Coefficient

- Pearson correlation coefficient (PCC) is a statistic that measures linear correlation between two features (or variables)
- Its outcome is in $[-1, +1]$
  - $+1$ means the two features have a perfectly positive linear correlation
  - 0 means that there is no linear correlation between them
  - -1 means they have a perfectly negative linear correlation

$$\text{Person}\left(X_i, X_j\right) = \frac{\mathbb{E}\left[\left(X_i - \mu_{X_i}\right)\left(X_j - \mu_{X_j}\right)\right]}{\sigma_{X_i} \times \sigma_{X_j}}$$

where σ_{X_i} and σ_{X_j} are the standard deviations of X_i and X_j, respectively

PCC (cont.)

$$\mathbf{X}\quad\begin{array}{c|c|c|c|c|}
 & \boldsymbol{x}_1 & \boldsymbol{x}_2 & & \boldsymbol{x}_N \\
\hline
 & \cdots & \cdots & \cdots & \cdots \\
\hline
X_i & x_{i1} & x_{i2} & \cdots & x_{iN} \\
\hline
 & \cdots & \cdots & \cdots & \cdots \\
\hline
X_j & x_{j1} & x_{j2} & \cdots & x_{jN} \\
\hline
 & \cdots & \cdots & \cdots & \cdots \\
\hline
\end{array}$$

- In practice, PCC between two X_i and X_j can be computed as

$$\mathrm{Person}\left(X_i, X_j\right) = \frac{\sum_{k=1}^{N}\left(\left(x_{ik} - \hat{\mu}_{X_i}\right) \times \left(x_{jk} - \hat{\mu}_{X_j}\right)\right)}{\sqrt{\sum_{k=1}^{N}\left(x_{ik} - \hat{\mu}_{X_i}\right)^2}\sqrt{\sum_{k=1}^{N}\left(x_{jk} - \hat{\mu}_{X_j}\right)^2}}$$

where $\hat{\mu}_{X_i}$ and $\hat{\mu}_{X_j}$ are the (unbiased) sample means of the features X_i and X_j, respectively.

$$\hat{\mu}_{X_i} = \frac{1}{N}\sum_{k=1}^{N} x_{ik}$$

PCC (cont.)

$$\mathbf{X} \quad \begin{array}{c|c|c|c|c|} & \boldsymbol{x}_1 & \boldsymbol{x}_2 & & \boldsymbol{x}_N \\ \hline & \dots & \dots & \dots & \dots \\ \hline X_i & x_{i1} & x_{i2} & \dots & x_{iN} \\ \hline & \dots & \dots & \dots & \dots \\ \hline X_j & x_{j1} & x_{j2} & \dots & x_{jN} \\ \hline & \dots & \dots & \dots & \dots \\ \hline \end{array}$$

$$\text{Person}\left(X_i, X_j\right) = \frac{\sum_{k=1}^{N}\left(\left(x_{ik} - \hat{\mu}_{X_i}\right) \times \left(x_{jk} - \hat{\mu}_{X_j}\right)\right)}{\sqrt{\sum_{k=1}^{N}\left(x_{ik} - \hat{\mu}_{X_i}\right)^2}\sqrt{\sum_{k=1}^{N}\left(x_{jk} - \hat{\mu}_{X_j}\right)^2}}$$

$$= \frac{\sum_{k=1}^{N}\left(\left(x_{ik} - \hat{\mu}_{X_i}\right) \times \left(x_{jk} - \hat{\mu}_{X_j}\right)\right)}{(N-1)\sqrt{\frac{\sum_{k=1}^{N}\left(x_{ik} - \hat{\mu}_{X_i}\right)^2}{N-1}}\sqrt{\frac{\sum_{k=1}^{N}\left(x_{jk} - \hat{\mu}_{X_j}\right)^2}{N-1}}}$$

$$= \frac{\sum_{k=1}^{N}\left(\left(x_{ik} - \hat{\mu}_{X_i}\right) \times \left(x_{jk} - \hat{\mu}_{X_j}\right)\right)}{(N-1) \times \hat{\sigma}_{X_i} \times \hat{\sigma}_{X_j}}$$

where $\hat{\sigma}_{X_i}$ and $\hat{\sigma}_{X_j}$ are the (unbiased) sample standard deviations of the features X_i and X_j, respectively

$$\hat{\sigma}_{X_i} = \sqrt{\frac{1}{N-1}\sum_{k=1}^{N}\left(x_{ik} - \hat{\mu}_{X_i}\right)^2}$$

PCC (cont.)

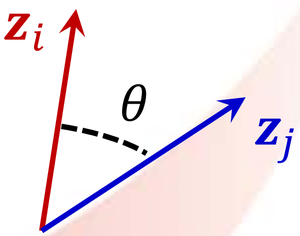
$$\mathbf{X} \quad \begin{array}{c|c|c|c|c|} & \boldsymbol{x}_1 & \boldsymbol{x}_2 & & \boldsymbol{x}_N \\ \hline & \dots & \dots & \dots & \dots \\ \hline X_i & x_{i1} & x_{i2} & \dots & x_{iN} \\ \hline & \dots & \dots & \dots & \dots \\ \hline X_j & x_{j1} & x_{j2} & \dots & x_{jN} \\ \hline & \dots & \dots & \dots & \dots \\ \hline \end{array}$$

$$\text{Person}\left(X_i, X_j\right) = \frac{\sum_{k=1}^{N}\left(\left(x_{ik} - \hat{\mu}_{X_i}\right) \times \left(x_{jk} - \hat{\mu}_{X_j}\right)\right)}{(N-1) \times \hat{\sigma}_{X_i} \times \hat{\sigma}_{X_j}}$$

$$= \frac{1}{N-1} \sum_{k=1}^{N} \left(\overset{\boldsymbol{x}'_{ik}}{\boxed{\left(\frac{x_{ik} - \hat{\mu}_{X_i}}{\hat{\sigma}_{X_i}}\right)}} \times \overset{\boldsymbol{x}'_{jk}}{\boxed{\left(\frac{x_{jk} - \hat{\mu}_{X_j}}{\hat{\sigma}_{X_j}}\right)}} \right)$$

Standardization on feature X_i Standardization on feature X_j

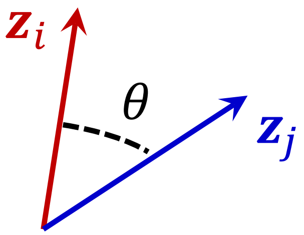
$$\boldsymbol{z}_i = \begin{pmatrix} \boldsymbol{x}'_{i1} \\ \boldsymbol{x}'_{i2} \\ \cdots \\ \boldsymbol{x}'_{iN} \end{pmatrix} \qquad \boldsymbol{z}_j = \begin{pmatrix} \boldsymbol{x}'_{j1} \\ \boldsymbol{x}'_{j2} \\ \cdots \\ \boldsymbol{x}'_{jN} \end{pmatrix}$$



$$\text{Person}\left(X_i, X_j\right) = \frac{\boldsymbol{z}_i \cdot \boldsymbol{z}_j}{N-1} = \frac{\|\boldsymbol{z}_i\|_2 \|\boldsymbol{z}_j\|_2}{N-1} \times \cos(\theta) = \frac{N-1}{N-1}\cos(\theta) = \cos(\theta)$$

As $\sqrt{\frac{1}{N-1}\sum_{k=1}^{N}\left(\boldsymbol{x}'_{ik} - 0\right)^2} = 1$, thus $\|\boldsymbol{z}_i\|_2 = \sqrt{\sum_{k=1}^{N} {\boldsymbol{x}'_{ik}}^2} = \sqrt{N-1}$

PCC (cont.)



- If Person$(X_i, X_j) > 0$, X_i and X_j are positively correlated: X_i's values increase (or decrease) as X_j's values increase (or decrease) and versa vice
  - The higher the value, the stronger the positive correlation
  - Maximum value: $+1$ when $\theta = 0°,$
- If Person$\left(X_i, X_j\right) = 0$, there is no correlation between values of X_i and X_j $(\theta = 90°,)$
- If Person$\left(X_i, X_j\right) < 0$, X_i and X_j are negatively correlated: X_i's values increase (or decrease) as X_j's values decrease (or increase) and versa vice
  - The lower the value, the stronger the negative correlation
  - Minimum value: -1 when $\theta = 180°$

Thank you!