

Preprocessing Data

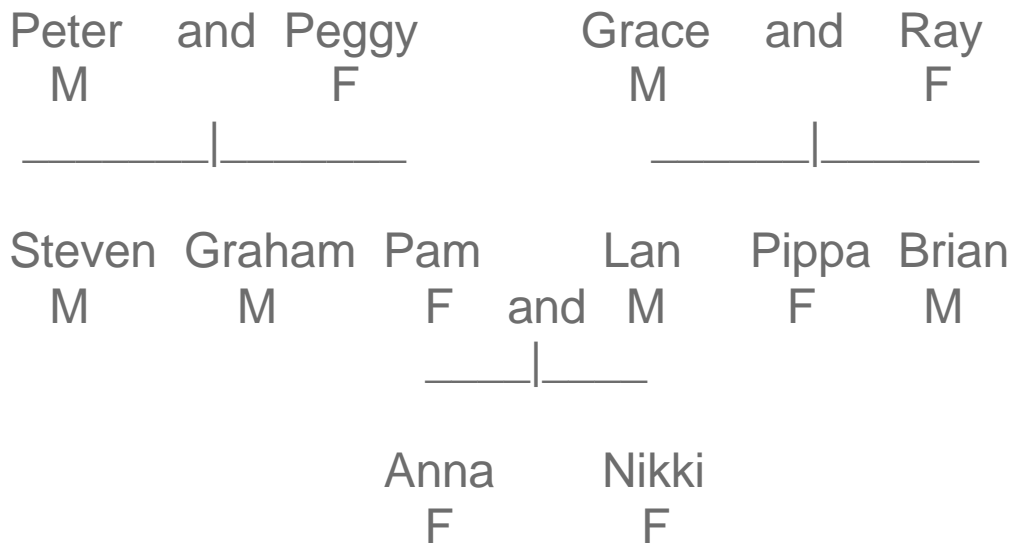
- ✍✍ Normalization and denormalization
- ✍✍ Missing values
- ✍✍ Outliers detection and removing
- ✍✍ Noisy Data
- ✍✍ Variants of Attributes
- ✍✍ Meta Data
- ✍✍ Data Transformation

Reading material:

- ✍✍ Chapters 2 and 3 of textbook by Witten,
- ✍✍ Chapter 1, Sections 3.1,3.2 and 5.2 of textbook book by Han.

Normalization and denormalization

Consider a family tree



Data in Table

Name	Gender	Parent1	Parent2
Peter	M	?	?
Peggy	F	?	?
Grace	M	?	?
Ray	F	?	?
Steven	M	Peter	Peggy
Graham	M	Peter	Peggy
Pam	F	Peter	Peggy
Lan	M	Grace	Ray
Pippa	F	Grace	Ray
Brian	M	Grace	Ray
Anna	F	Lan	Pam
Nikki	F	Lan	Pam

Two Tables for the 'Sister of Relation'

First person	Second person	Sister of ?
Peter	Peggy	no
...
Steven	Peter	no
Steven	Peggy	no
Steven	Pam	Yes
...
Lan	Pippa	Yes
...
Anna	Nikki	Yes
...
Nikki	Anna	Yes

Quite confusing without the tree!

First person	Second person	Sister of ?
Steven	Pam	Yes
Graham	Pam	Yes
Lan	Pippa	Yes
Brian	Pippa	Yes
Anna	Nikki	Yes
Nikki	Anna	Yes

All the rest	No
--------------	----

Not much helpful without consulting the tree.

Denormalization: Join two or more relations to make a new one. A process of flattening. Each old relation is cast as an independent attribute regarding the new relation.

First person				Second person				Sister of ?
name	g.	parent1	parent2	name	g.	parent1	parent2	
Steven	M	Peter	Peggy	Pam	F	Peter	Peggy	Yes
Graham	M	Peter	Peggy	Pam	F	Peter	Peggy	Yes
Lan	M	Grace	Ray	Pippa	F	Grace	Ray	Yes
Brian	M	Grace	Ray	Pippa	F	Grace	Ray	Yes
Anna	F	Lan	Pam	Nikki	F	Lan	Pam	Yes
Nikki	F	Lan	Pam	Anna	F	Lan	Pam	Yes
All the rest								No

If second person's gender = female and first person's parent = second person's parent then sister-of = yes

Denormalization in Business:

Transaction ID	Date	Buy product
A1	01/Sep/02	Pen, Notebook
A2	02/Sep/02	Books, Case

More Tables: Product and Supplier,
Supplier and its address...

Data mining might find some relations among the buy products as well the relations between date and people's shopping behavior.

— Denormalization may produce spurious regularities that reflect structure of database

Example: "supplier" predicts "supplier address"

— **Infinite relations require recursion**

If person1 is a parent of person2
then person1 is an ancestor of
person2

If person1 is a parent of person2
and person2 is an ancestor of
person3 then person1 is an ancestor
of person3

Variants of Normalization:

Database normalization: a process of efficiently organizing data in a database to eliminate redundant data (for example, storing the same data in more than

one table) and ensure data dependencies make sense (only storing related data in a table).

Example: The data structure of the web.

Normalization for Attributes: scaling the attribute values so they fall within a specified range.

Table 1:

Project number	Project name	Employee number	Employee name	Rate category	Hourly rate
1023	Madagascar travel site	11	Vincent Radebe	A	\$60
		12	Pauline James	B	\$50
		16	Charles Ramoraz	C	\$40
1056	Online estate agency	11	Vincent Radebe	A	\$60
		17	Monique Williams	B	\$50

Table 2: employee_project table

Project number	Project name	Employee number	Employee name	Rate category	Hourly rate
1023	Madagascar travel site	11	Vincent Radebe	A	\$60
1023	Madagascar travel site	12	Pauline James	B	\$50
1023	Madagascar travel site	16	Charles Ramoraz	C	\$40

1056	Online estate agency	11	Vincent Radebe	A	\$60
1056	Online estate agency	17	Monique Williams	B	\$50

Table 3: employee_project table

Project number	Employees
1023	11
1023	12
1023	16
1056	11
1056	17

Table 4: Employee table

Employee number	Employee name	Rate category	Hourly rate
11	Vincent Radebe	A	\$60
12	Pauline James	B	\$50
16	Charles Ramoraz	C	\$40
17	Monique Williams	B	\$40

Table 5: Project table

Project number	Project name
1023	Madagascar travel site
1056	Online estate agency

Table 6: Employee table

Employee number	Employee name	Rate category
11	Vincent Radebe	A
12	Pauline James	B
16	Charles Ramoraz	C
17	Monique Williams	B

Table 7: Rate table

Rate category	Hourly rate
A	\$60
B	\$50
C	\$40

First step: Raw data to table. Then we define the primary keys:

- ✍ Project number - primary key
 - Project name
 - Employee number - primary key
 - Employee name
 - Rate category
 - Hourly rate

Apply the same idea to the new table to narrow our search down to get additional tables.

Attributes: nominal, ordinal, interval, ...

Nominal quantities are ones whose values are distinct symbols that serve only as labels or names

— Example: “outlook”: “sunny”, “overcast”, and “rainy”

— No relation is implied among nominal values (no ordering or distance measure)

— Only equality tests can be performed

Ordinal quantities are ones with imposed ordered values

— Example: “temperature”: “hot” > “mild” > “cool”

— Very hard to define distance and operations such as addition and subtraction.

Nominal vs. ordinal

— Attribute “age” nominal

— Attribute “age” ordinal (e. g. “young” < “pre-presbyopic” < “presbyopic”)

If age = young and astigmatic = no and tear production rate = normal then recommendation = soft

If age = pre-presbyopic and astigmatic

= no and tear production rate = normal
then recommendation = soft

Using the ordering, we obtain

If age \leq pre-presbyopic and astigmatic
= no and tear production rate = normal
then recommendation = soft

Interval quantities have ordered values that measured in fixed and equal unit.

- Examples: attribute “temperature” expressed in degrees, attribute “year”
- Difference of two values makes sense
- Sum or product doesn’t make sense

Question: How to define the zero point?

Ratio quantities are ones for which the measurement scheme defines a zero point

- Example: attribute “distance”
- Ratio quantities are treated as real numbers

All mathematical operations are allowed.

Is there an “inherently” defined zero point?

Answer depends on scientific knowledge (e. g. Fahrenheit knew no lower limit to temperature)

Transforming ordinal to boolean

— Simple transformation allows to code ordinal attribute with n values using n-1 boolean attributes

— Example: attribute “temperature”

Better than coding it as a nominal attribute

Original Data

Temperature
Hot
Medium
Cold

Transformed Data

Temperature>cold	Temperature>medium
True(1)	True(1)
True (1)	False(0)
False(0)	False(0)

Transforming nominal to boolean

Original Data

Outlook
Sunny
Rainy
Overcast

Transformed Data

Outlook = Sunny	Outlook=Rainy
True(1)	False(0)
False(0)	True(1)
False(0)	False(0)

If the attribute has n values, then $n-1$ synthetic Boolean variable is needed for the transformation.

Metadata: Information about the data that encodes background knowledge

— Can be used to restrict search space

1,SEPTEMBER: labor day, long weekend, the day before the new semester, the last day of summer holidays...

Preparing the input

— Denormalization is necessary

— Problem: different data sources (e. g. departments of sales and customer billing)

Data must be assembled, integrated, cleaned up

Table 1

lastname	Firstname	Street#	St.Name	City	State
Smith	John	123	_elm_St	Charlotte	NC
Jones	Bill_	_456	Maple_ave	Chicago	IL
Williams	Mike	789	Main St	Brooklyn	NY
Smith	Suzie	123	Elm_St	Charlotte	NC

Table 2

Full name	Account	Trans-type	Date	Amount
Mr. Mike Williams	43659439	Deposit	980414	50.83
Mr. Mike Williams	54968584	Deposit	980418	74.16
Mr. Bill Jones	54943920	Deposit	980422	129.69
Ms. Sue Smith	8839493	Withdrawal	980512	95.33

Integrating Table 1 and 2

Full Name	Acc Numb	Acc Type	Trans Type	Date	Mth	Week Day	AMT1	AMT2	INDX
Mike Williams	43659463	Savg	D	980414	Apr	Tue	50.85	50	1
Mike Williams	54968584	Chck	D	980418	Apr	Sat	74.16	75	2
Bill Jones	54943920	Chck	D	980422	Apr	Wed	129.69	150	3
Sue Smith	8839493	Savg	W	980512	May	Tue	95.33	100	4
...

Missing values:

— Frequently indicated by symbol ‘?’

Reasons: malfunctioning equipment, changes in experimental design, collation of different datasets, measurement not possible

— Missing value may have significance in itself (e. g. missing test in a medical examination)

Most schemes assume that is not the case

‘missing’ may need to be coded as additional value

Dealing with Missing Values:

1. Ignore the tuple, in particular when the class label is missing.

---- Not recommended

2. Manually fix the missing values, too time-consuming

3. Use a global value to replace the missing values

----need to understand the domain space very well

4. Use the mean to fill the missing values

----works for numeric attributes

5. Use the attribute mean for all samples in the same class to fill the missing value

6. Use the most probable value to fill in the missing value regarding to all the instances in the data set or the instances in the same class

Methods 3 to 6 are biased to different learning schemes, and 6 is the most popular. In particular, it is the only reasonable way to deal with nominal attribute with missing values in many learning scheme.

Inaccurate values

— Reason: data has not been collected for mining it

— Result: errors and omissions that don't affect original purpose of data (e. g. age of customer)

— Typographical errors in nominal attributes??

----Values need to be checked for consistency

— Typographical and measurement errors in numeric attributes??

----Outliers need to be identified

— Errors may be deliberate (e. g. wrong zip codes)

Dealing with Noisy Data:

1. Binning: Binning methods smooth a sorted data value by consulting its neighborhood. There are two ways, smoothing by bin means or bin boundaries.

Example: 4,8,15,21,21,24,25,28,34

Partition into Bins

Bin1: 4 8 15 {9,9,9} {4,4,15}

Bin2: 21 21 24 {22,22,22} {21,21,24}

Bin3: 25 28 34 {29,29,29} {25,25,34}

2. Use Clustering to group similar values and detect outliers. One method is as follows. For each instance, we can define a neighborhood around it, then count all the instances in this neighborhood. If the total instances in the neighborhood exceed a certain fraction (pre-specified) of the total instances in the data set, then the instance is not an outlier, otherwise is.

3. Combine computer and human inspection: time-consuming

1. Use regression to smooth out the noise data

2. Method based on statistical model.

Assume the values of the attribute follow some distribution model(pre-assumed or extracted from the data set), and then compute the probability of each instance based on the distribution model. If the probability is below a certain threshold, then the instance is an outlier.

Data Integration and Transformation:

Matching up equivalent real-world entities from multiple data sources, very tricky

Redundancy detecting: Whether an attribute can be determined by another one?

We use the correlation to characterize this kind of relation

$$R(A,B) = \frac{(\sum(A - \bar{A})(B - \bar{B})) / ((n-1)\sigma(A)\sigma(B))$$

Where

$\bar{A} = \sum A / n$ is the mean of A and

$$\sigma(A) = \sqrt{\sum (A - \bar{A})^2 / (n-1)}$$

is the standard deviation of A

More relations among different attributes can be detected by using association...

Data Transformation involves

smoothing, aggregation (aggregate daily data to get the monthly total amount),

generalization (from low level to high level, Street to city, age from years to young, mid,

senior), normalization (scale the attribute value to be in a certain interval such as [0,1] or [-1,1]), removal attributes...

A typical way for normalization is

$$V' = (v - \min(v)) / (\max(v) - \min(v)), \text{ scaled the values of } v \text{ to } [0,1]$$