v2016

Oppgave 1 - Various

a. List four usages for clustering validation/evaluation

   1. To avoid finding patterns in noice
   2. To compare clustering algorithms
   3. To compare two sets of clusters
   4. To compare two clusters

b. Explain four techniques for data cleaning in the context of web usage data.

   1. Removing extraneous references to embedded objects that may not be important for the purpose of analysis, including references to style files, graphics, or sound files
   2. The cleaning process also may involve the removal of at least some of the data fields (e.g. number of bytes transferred or version of HTTP protocol used, etc.) that may not provide useful information in analysis or data mining tasks
   3. Data cleaning also entails the removal of references due to crawler navigations.
   4. Add missing refrences caused by caching

c.

   ▼ Task

   Assume two bit vectors p and q,
   $p = \{1, 0, 1, 0, 0, 0, 0, 1, 1, 1\}$
   $q = \{1, 0, 0, 0, 0, 0, 1, 1, 0, 1\}$
   Calculate the Jaccard coefficient for the bit vectors p and q.

   $f_{00} = 4$, number of attribuetes where p and q is 0

   $f_{01} = 1$, number of attribuetes where p is 0 and q is 1

   $f_{10} = 2$, number of attribuetes where p is 1 and q is 0

   $f_{11} = 3$, number of attribuetes where p and q is 1

   $J = \frac{f_{11}}{f_{01}+f_{10}+f_{11}} = \frac{3}{1+2+3} = \frac{3}{6} = 0.5$

Oppgave 2 - Modeling

▼ Task

In this task we ask you to model a data warehouse for car damages for the insurance company Lillebrand. Lillebrand wants a data warehouse in order to be able to analyze events that have resulted in insurance payments.
Examples of analysis one should be able to perform using the data warehouse:
 Number of damages in 2015.
 Average number of damages per month.
 Number of damages for each quarter in 2015.
 Total amount paid for each car type.
 Number of damages of type “collision” for each city.
The description is somewhat imprecisely formulated and it is part of the task to select what should be included. We are primarily looking for you to show modeling principles for data warehousing. Explain any assumptions you find it necessary to do.
Make a star schema for the described case.

Dimention | Time - All > Year > Quarter > Month > Day

Dimention | Car > All > Model > Type

Dimention | Location > All > Country > City > Address

Dimention | Damage > All > type

Dimention | Customer > All > name >

Facts | time, car, location, damage, customer | price, {sum(payment), count(damage), avg(damage)}

Oppgave 3 - OLAP

a.

▼ Task

Given a cube with dimensions:

`Time(day < month < quarter < year)`

`Item(item_name < brand < type)`

`Location(street < city < province_or_state < country)`

Assume the following materialized cuboids:

1. `{year, item_name, city}`
2. `{year, brand, country}`
3. `{year, brand, province_or_state}`
4. `{item_name, province_or_state} where year = 2004`

Given the following OLAP query: `{item_name, province_or_state} with condition "year = 2006"`

Which of the materialized cuboids can be used to process the query? Justify the answer

Not 4 since it is restricted to `year = 2004`

Not 2 since `country` is a more general concept than `province_or_state`. Finer-granularity data cannot be generated from coarser-granularity data

Not 3 because `brand` is a more general concept that `item_name`

Then we are left with 1 where each element is at a lower level.

b.

▼ Task

Given a data warehouse with three tables Location/Item/Sales, where Sales is the fact table and the two others are dimension tables. We want to use join indexes in order to process queries more efficiently. Show the structure and contents of the join indexes Location/Sales and Item/Sales based on the contents of the three tables below.

Location	
LocKey	**CityName**
L1	Oslo
L2	Athen
L3	Trondheim

Item	
ItemKey	**ItemName**
I1	Sony-TV
I2	Rolex
I3	Lexus

Sales			
TransID	**LocKey**	**ItemKey**	**Price**
T1	L1	I1	5
T2	L2	I2	8
T3	L1	I1	6
T4	L3	I1	5
T5	L3	I3	9
T6	L1	I2	8
T7	L1	I1	4

Item/Sales

ItemKey	TransID
I1	{T1, T3, T4, T7}
I2	{T2, T6}
I3	{T5}

Location/Sales

LocKey	TransID
L1	{T1, T3, T6, T7}

L2	{T2}
L3	{T4, T5}

Oppgave 4 - Clustering

▼

Assume a two-dimensional dataset as shown in the table to the right. Cluster this dataset using DBSCAN, given MinPts=4 (incl. own point) and Eps=3 (incl. points having distance 3). Use Manhattan distance.

X	Y
4	8
4	9
4	10
4	13
4	14
5	3
5	7
5	14
6	15
6	16
6	19
7	11
7	16
7	17
7	18
7	19

> Manhattan distance: $|x_1 - x_2| + |y_1 - y_2|$

Creating distance matrix:

	(4,8)	(4,9)	(4,10)	(4,13)	(4,14)	(5,3)	(5,7)	(5,14)	(6,15)	(6,16)	(6,19)	(7,11)	(7,16)	(7,17)	(7,18)	(7,19)
(4,8)	0	1	2	5	6	6	2	7	9	10	13	6	11	12	13	14
(4,9)	1	0	1	4	5	7	3	6	8	9	12	5	10	11	12	13
(4,10)	2	1	0	3	4	8	4	5	7	8	11	4	9	10	11	12
(4,13)	5	4	3	0	1	11	7	2	4	5	8	5	6	7	8	9
(4,14)	6	5	4	1	0	12	8	1	3	4	7	6	5	6	7	8
(5,3)	6	7	8	11	12	0	4	11	13	14	17	10	15	16	17	18
(5,7)	2	3	4	7	8	4	0	7	9	10	13	6	11	12	13	14
(5,14)	7	6	5	2	1	11	7	0	2	3	6	5	4	5	6	7
(6,15)	9	8	7	4	3	13	9	2	0	1	4	5	2	3	4	5
(6,16)	10	9	8	5	4	14	10	3	1	0	3	6	1	2	3	4
(6,19)	13	12	11	8	7	17	13	6	4	3	0	9	4	3	2	1
(7,11)	6	5	4	5	6	10	6	5	5	6	9	0	5	6	7	8
(7,16)	11	10	9	6	5	15	11	4	2	1	4	5	0	1	2	3
(7,17)	12	11	10	7	6	16	12	5	3	2	3	6	1	0	1	2
(7,18)	13	12	11	8	7	17	13	6	4	3	2	7	2	1	0	1
(7,19)	14	13	12	9	8	18	14	7	5	4	1	8	3	2	1	0

Now we find each point having more then 4 close points. Starting with the first point `(4, 8)` we find the neighbouring cores. Then we go to these neighbours and do the same until all nodes are found. We find that:
[point(4, 8, PointType.CORE),
point(4, 10, PointType.CORE),
point(4, 9, PointType.CORE),
point(4, 13, PointType.CORE),

point(5, 14, PointType.CORE),
point(4, 14, PointType.CORE),
point(6, 16, PointType.CORE),
point(6, 15, PointType.CORE),
point(7, 18, PointType.CORE),
point(7, 17, PointType.CORE),
point(7, 16, PointType.CORE),
point(6, 19, PointType.CORE),
point(7, 19, PointType.CORE),
point(5, 7, PointType.BORDER)]

Oppgave 5 - Classification

a. Explain cross validation and what this technique is used for.

   Cross validation is splitting the data into k subsets. Run training on k-1 subsets and test on the remaining one. Do this k times each time changing the test subset

b.

   ▼ Task

   A car insurance company has for existing customers stored information that include customer number, age (L/M/H, e.g., 18-25/26-70/71-100), car type, driving length per year (4000/8000/20000/Unlimited), bonus (Low/Medium/High) and if they have had damage on their car that has been covered by the insurance. When new customers ask for a price offer, the company want to set the price to normal or high depending on whether they believe the customer is going to get a damage on the car or not, i.e., they want to predict the attribute ”Damage”.

Cust.nr.	Age	Car type	Lenght per year	Bonus	*Damage*
1	L	Ferrari	8000	Low	*Yes*
2	M	BMW	8000	High	*No*
3	H	Lexus	Unlimited	High	*Yes*
4	L	Audi	8000	High	*No*
5	H	Opel	8000	Low	*Yes*
6	M	Toyota	8000	Low	*No*
7	M	Honda	8000	High	*No*
8	M	Nissan	8000	High	*No*
9	M	Audi	Unlimited	High	*No*
10	M	BMW	8000	Low	*Yes*
11	H	Toyota	Unlimited	High	*No*
12	L	Nissan	4000	Low	*Yes*
13	L	Opel	Unlimited	High	*Yes*
14	M	Audi	8000	High	*No*
15	M	Opel	8000	High	*No*
16	M	Toyota	4000	Low	*No*

Assume that we will use decision tree as the classification method. We will use the above dataset as our training data. We use the Gini index as measure for impurity, and the following two equations might be of help for solving the problem:

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

$$GAIN_{split} = GINI(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} GINI(i) \right)$$

Task: The goal of the classification is to be able to predict “Damage”. Compute the GAINsplit for splitting by attribute (1) ”Age” and (2) ”Bonus”. Which of these splits would you chose to start building your decision tree? Justify your answer

Gini:

Yes: 6, No: 10 → High impurity

$GINI = 1 - (\frac{6}{16})^2 - (\frac{10}{16})^2 = 0.47$

Age:

	Yes	No
L	3	1

Bonus

	Yes	No
High	2	8

M	1	8	Low	4	2
H	2	1			

$GINI_L = 1 - (\frac{3}{4})^2 - (\frac{1}{4})^2 = 0.38$

$GINI_M = 1 - (\frac{1}{9})^2 - (\frac{8}{9})^2 = 0.20$

$GINI_H = 1 - (\frac{2}{3})^2 - (\frac{1}{3})^2 = 0.44$

$GINI_{age} = \frac{4}{16} * 0.38 + \frac{9}{16} * 0.20 + \frac{3}{16} * 0.44 = 0.29$

$GAIN_{age} = 0.47 - 0.29 = 0.18$

I would start splitting by `age` since this gives the best `GAIN`

$GINI_{high} = 1 - (\frac{2}{10})^2 - (\frac{8}{10})^2 = 0.32$

$GINI_{low} = 1 - (\frac{4}{6})^2 - (\frac{2}{6})^2 = 0.44$

$GINI_{bonus} = \frac{10}{16} GINI_{high} + \frac{6}{16} GINI_{low} = \frac{10}{16} * 0.32 + \frac{6}{16} * 0.44 = 0.37$

$GAIN_{bonus} = 0.47 - 0.37 = 0.10$

Oppgave 6 - Association rule

a. .

▼

Assume the market basket data below. Use the apriori-algorithm to find all frequent itemsets with minimum support of 50 % (i.e., minimum support count is 4). Use the $F_{k-1} \times F_{k-1}$ method for candidate generation.

TransactionID	Item
T1	ABCDFGH
T2	DKM
T3	FK
T4	ACGH
T5	ACDDGH
T6	BM
T7	DFKM
T8	ABCDGH

Unique chars:

A	4
B	3
C	4
D	5
F	3
G	4

H	4
K	3
M	3
A, C	4
A, D	3
A, G	4
A, H	4
C, D	3
C, G	4
C, H	4
D, G	3
D, H	3
G, H	4
A, C, G	4
A, C, H	4
A, G, H	4
C, G, H	4
A, C, G, H	4

b. .

▼ Task

Assume the market basket data below. You are now going to use the FP-growth-algorithm in order to find all frequent itemsets with minimum support of 60 % (i.e., minimum support count is 3).

1) Construct a FP tree based on the dataset.

2) Find frequent itemsets using the FP-growth-algorithm. Use table notation with the following columns in order to show the result:

- Item
- ”Conditional pattern base”
- ”Conditional FP-tree”
- Frequent itemsets

TransactionID	Item
T1	f, a, c, d, g, i, m, p
T2	a, b, c, f, l, m, o
T3	b, f, h, j, o
T4	b, c, k, s, p
T5	a, f, c, e, l, p, m, n

[(('c',), 4), (('f',), 4), (('a',), 3), (('b',), 3), (('m',), 3), (('p',), 3)]

[[('c',), ('f',), ('a',), ('m',), ('p',)],
[('c',), ('f',), ('a',), ('b',), ('m',)],
[('f',), ('b',)],
[('c',), ('b',), ('p',)],
[('c',), ('f',), ('a',), ('m',), ('p',)]]

root

|--c(4)

|--f(3)

|--a(3)

|--m(2)

|--p(2)

|--b(1)

|--m(1)

|--b(1)

|--p(1)

|--f(1)

|--b(1)

Item	Conditional subdatabase	Conditional FP-tree	Frequnt itemset
p	{(cfam:2), (cb:1)}	{(c:3)}|p	p, cp
m	{(cfa:2), (cfab:1)}	{(cfa:3)}|m	m, cm, fm, am, cfm, cam, fam, cfam
b	{(cfa:1), (f:1), (c:1)}	Ø	b
a	{(cf:3)}	{(cf:3)}|a	a, ca, fa, cfa

f	{(c:3)}	{(c:3)}|f	f, cf
c	Ø	Ø	c