CS699
Lecture 9
Correlation Analysis
Other Frequent Pattern Mining

Association Rule Mining on Weka

- Data preparation
  - When performing association rule mining on a transactional data using Weka, the dataset must be converted to an appropriate form.
  - Each item becomes an attribute.
  - Each attribute takes on only single value, e.g., {1} or {t}
  - Only items are used (i.e., transaction id's, customer id's, etc. are removed, temporarily or permanently).

Association Rule Mining on Weka

- Data preparation example

CID	Items
C1	beer, bread, chip, egg
C2	beer, bread, chip, egg, popcorn,
C3	bread, chip, egg
C4	beer, bread, chip, egg, milk, popcorn
C5	beer, bread, milk
C6	beer, bread, egg
C7	bread, chip, milk
C8	bread, butter, chip, egg, milk
C9	butter, chip, egg

```
@relation d1-ar-2

@attribute beer {1}
@attribute bread {1}
@attribute butter {1}
@attribute chip {1}
@attribute egg {1}
@attribute milk {1}
@attribute popcorn {1}

@data
1,1,?,1,1,?,?
1,1,?,1,1,?,1
?,1,?,1,1,?,?
1,1,?,1,1,1,1
1,1,?,?,?,1,?
1,1,?,?,1,?,?
?,1,?,1,?,1,?
?,1,1,1,1,1,?
?,?,1,1,1,?,?
```

Association Rule Mining on Weka

- Running Apriori on Weka
  - Starts with min. support of 100% and decreases this in steps of 5% until there are at least 10 rules with the min. confidence of 90% or until the support has reached a lower bound of 10%.
  - These default values can be changed.

Interestingness Measure: Correlations (Lift)

- *play basketball* $\Rightarrow$ *eat cereal* [40%, 66.7%] is misleading
  - The overall % of students eating cereal is 75% > 66.7%.
- *play basketball* $\Rightarrow$ *not eat cereal* [20%, 33.3%] is more accurate, although with lower support and confidence
- Measure of dependent/correlated events: lift

$$lift = \frac{P(A \cup B)}{P(A)P(B)}$$

	Basketball	Not basketball	Sum (row)
Cereal	2000	1750	3750
Not cereal	1000	250	1250
Sum(col.)	3000	2000	5000

$$lift(B, C) = \frac{2000/5000}{3000/5000 * 3750/5000} = 0.89$$

$$lift(B, \neg C) = \frac{1000/5000}{3000/5000 * 1250/5000} = 1.33$$

- lift > 1: positively correlated, lift < 1: negatively correlated, lift = 1: independent

Chi-square Test

	Basketball	Not basketball	Sum (row)
Cereal	2000 (2250)	1750 (1500)	3750
Not cereal	1000 (750)	250 (500)	1250
Sum(col.)	3000	2000	5000

- Chi-square test can be used as a test of independence of two variables
- Given the above contingency table, we want to determine whether there is a correlation between cereal and basketball.
- Perform the chi-square test.
- Null hypothesis: They are independent of each other.

Chi-square Test

	Basketball	Not basketball	Sum (row)
Cereal	2000 (2250)	1750 (1500)	3750
Not cereal	1000 (750)	250 (500)	1250
Sum(col.)	3000	2000	5000

- First, we compute the expected values (shown in the parentheses)

  Example: For (cereal, basketball)

    Expected value = (3750 * 3000) / 5000 = 2250

- Second, compute the chi-square test statistic:

$$\chi^2 = \frac{(2000-2250)^2}{2250} + \frac{(1750-1500)^2}{1500} + \frac{(1000-750)^2}{750} + \frac{(250-500)^2}{500} = 277.78$$

Chi-square Test

- Third, look up the chi-square distribution table.

degrees of freedom = (num_rows – 1) * (num_cols – 1) = 1, and α = 0.05

$$\chi^2_{0.05,1} = 3.84$$

III. Percentage Points of the χ^2 Distribution[a]

ν	.995	.990	.975	.950	.500	.050	.025	.010	.005
					α				
1	0.00 +	0.00 +	0.00 +	0.00 +	0.45	3.84	5.02	6.63	7.88
2	0.01	0.02	0.05	0.10	1.39	5.99	7.38	9.21	10.60
3	0.07	0.11	0.22	0.35	2.37	7.81	9.35	11.34	12.84
4	0.21	0.30	0.48	0.71	3.36	9.49	11.14	13.28	14.86
5	0.41	0.55	0.83	1.15	4.35	11.07	12.38	15.09	16.75
6	0.68	0.87	1.24	1.64	5.35	12.59	14.45	16.81	18.55
7	0.99	1.24	1.69	2.17	6.35	14.07	16.01	18.48	20.28
8	1.34	1.65	2.18	2.73	7.34	15.51	17.53	20.09	21.96
9	1.73	2.09	2.70	3.33	8.34	16.92	19.02	21.67	23.59
10	2.16	2.56	3.25	3.94	9.34	18.31	20.48	23.21	25.19
11	2.60	3.05	3.82	4.57	10.34	19.68	21.92	24.72	26.76

Chi-square Test

- Finally, compare the computed test statistic with the value from the distribution table and make a conclusion.
- In this example, the computed chi-square value is greater than that from the chi-square distribution table (i.e., it is in the rejection region)
- So, we reject the null hypothesis and conclude that there is a correlation between the two.

Null Transactions

- When the number of null transactions is large, these measures may generate misleading results.

	milk	Not milk	Sum (row)
coffee	100	1,000	1,100
Not coffee	1,100	100,000	101,100
Sum(col.)	1,200	101,000	102,200

$$lift(m,c) = \frac{100/102200}{1200/102200 * 1100/102200} = 7.74$$

- The *lift* measure indicates they are positively correlated.
- But, actual data says they are negatively correlated.
- Among 1,100 people who bought coffee, only 100 (or only 9%) bought also milk. This is similar with those who bought milk.

all_confidence and cosine

- Between 0 and 1
- greater than 0.5: positively correlated; smaller than 0.5: negatively correlated

	milk	Not milk	Sum (row)
coffee	100	1,000	1,100
Not coffee	1,100	100,000	101,100
Sum(col.)	1,200	101,000	102,200

- $all\_conf(m,c) = \frac{\sup(m \cup c)}{\max\{\sup(m), \sup(c)\}} = \frac{100}{1200} = 0.08$
- $cosine(m,c) = \frac{sup(m \cup c)}{\sqrt{\sup(m) \times \sup(c)}} = \frac{100}{\sqrt{1200 \times 1100}} = 0.09$
- These measures show they are negatively correlated.
- *all_confidence* and *cosine* measures are *null-invariant*.

all_confidence, cosine: another example

	milk	Not milk	Sum (row)
coffee	1,000	1,000	2,000
Not coffee	1,000	100,000	101,000
Sum(col.)	2,000	101,000	103,000

- $lift(m,c) = \frac{1000/103000}{(\frac{2000}{103000})\times(\frac{2000}{103000})} = 25.75$ (says positively correlated)
- $all\_conf(m,c) = \frac{1000}{2000} = 0.5$ (says independent)
- $cosine(m,c) = \frac{1000}{\sqrt{2000\times2000}} = 0.5$ (says independent)
- Actual data: independent
- Other measures: *max_confidence, Kulczynski* measure

Kulczynski and Imbalance Ratio (IR)

	milk	Not milk	Sum (row)
coffee	1,000	1,000	2,000
Not coffee	1,000	100,000	101,000
Sum(col.)	2,000	101,000	103,000

- $Kulc(A, B) = (P(A|B) + P(B|A))$ / 2, or average of two cond. prob.
- Between 0 and 1; > 0.5: positive; < 0.5: negative; = 0.5: independent
- $Kulc(m, c) = \frac{1}{2}(P(m|c) + P(c|m)) = \frac{1}{2}(\frac{mc}{c} + \frac{mc}{m}) = \frac{1}{2}(\frac{1000}{2000} + \frac{1000}{2000}) = 0.5$ (independent)

$$IR(A,B) = \frac{|\sup(A) - \sup(B)|}{\sup(A) + \sup(B) - \sup(A \cup B)}$$, $0 \leq IR < 1$

$$IR(m,c) = \frac{|m - c|}{m + c - mc} = \frac{|2000 - 2000|}{2000 + 2000 - 1000} = 0$$ (balanced)

Kulczynski and Imbalance Ratio (IR)

- In the table in the next slide, *mc*, *m'c*, *mc'*, and *m'c'* represent the following entries in the contingency table.:

	milk	Not milk	Sum (row)
coffee	*mc*	*m'c*	
Not coffee	*mc'*	*m'c'*	
Sum(col.)			

Comparison

	mc	m'c	mc'	m'c'	lift	all_conf	cosine	Kulc	IR
D1(P)	10000	1000	1000	100000	9.26 (P)	0.91(P)	0.91(P)	0.91(P)	0
D2(P)	10000	1000	1000	100	1.00(I)	0.91(P)	0.91(P)	0.91(P)	0
D3(N)	100	1000	1000	100000	8.44(P)	0.09(N)	0.09(N)	0.09(N)	0
D4(I)	1000	1000	1000	100000	25.75(P)	0.50(I)	0.50(I)	0.50(I)	0
D5(*)	1000	100	10000	100000	9.18(P)	0.09(N)	0.29(N)	0.50(I)	0.83
D6(*)	1000	10	100000	100000	1.97(P)	0.01(N)	0.01(N)	0.50(I)	0.99

- P: positive, N: negative, I: independent, *: contradictory
- Both D1 and D2 have positively correlated data but *lift* shows different values.
- D3 has negatively correlated data but *lift* says positive.
- D4 has independent data but *lift* says positive. This is because *lift* is affected by null transactions.
- *all_conf*, *cosine*, and *Kulczynski* are *null-invariant*.

Comparison (continued)

	mc	m'c	mc'	(mc)'	lift	all_conf	cosine	Kulc	IR
D5(*)	1000	100	10000	100000	9.18(P)	0.09(N)	0.29(N)	0.50(I)	0.83
D6(*)	1000	10	100000	100000	1.97(P)	0.01(N)	0.01(N)	0.50(I)	0.99

- P: positive, N: negative, I: independent, *: contradictory
- D5: $P(c|m)$ = 9.09% (negatively correlated)

  D5: $P(m|c)$ = 90.9% (positively correlated)
- D6: $P(c|m)$ = 0.99% (negatively correlated)

  D6: $P(m|c)$ = 99% (positively correlated)
- *Kulczynski* says independent (makes sense)
- IR indicates D5 and D6 are unbalanced.
- *Kulczynski* along with IR is recommended.

Multi-level Association Rule Mining

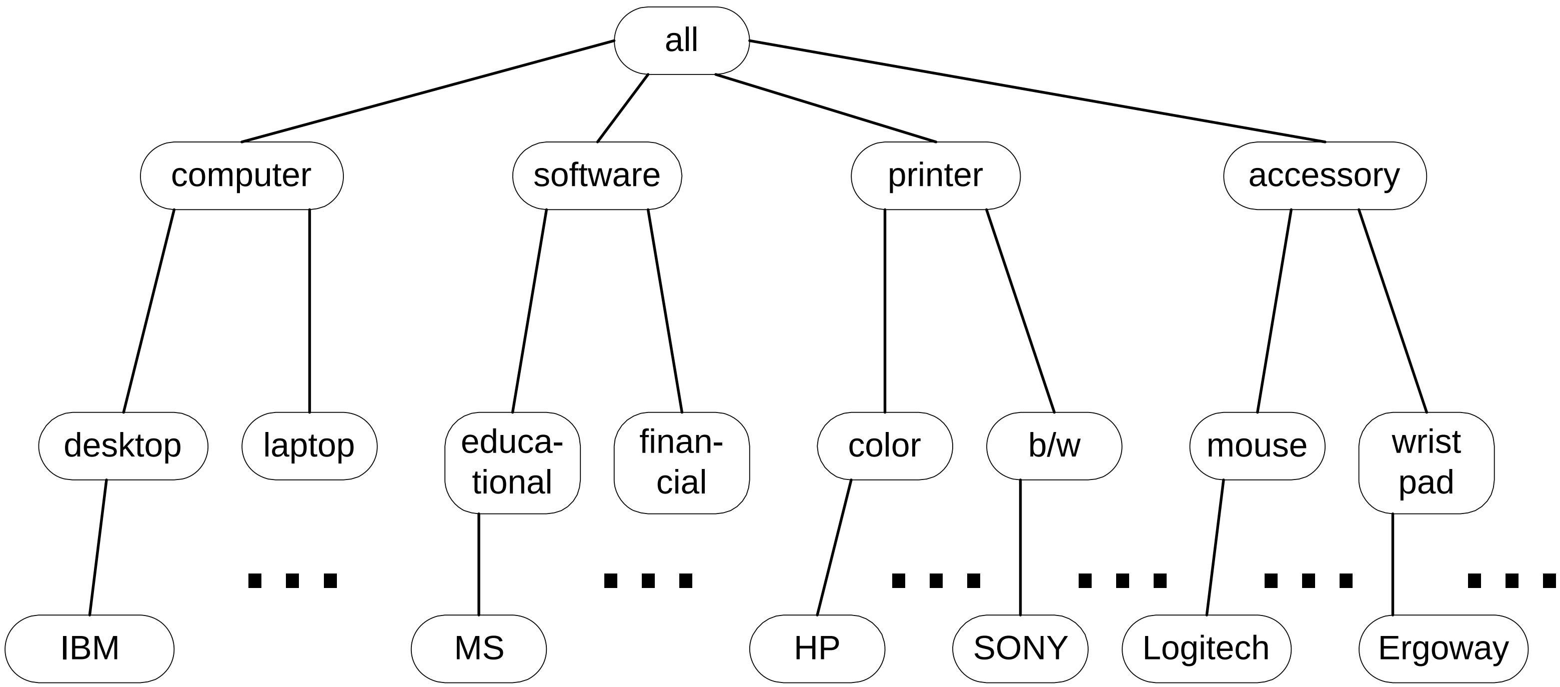
- When there is a concept hierarchy in the database
- Not many strong association rules at low levels
- Different users are interested in association rules at different levels

- Example database

TID	Items
1	IBM desktop computer, Sony b/w printer
2	MS educational SW, MS financial management SW
3	Logitech mouse, Ergoway wrist pad
4	IBM desktop computer, MS financial management SW
5	IBM desktop computer
…	…

Multi-level Association Rule Mining

- Concept hierarchy

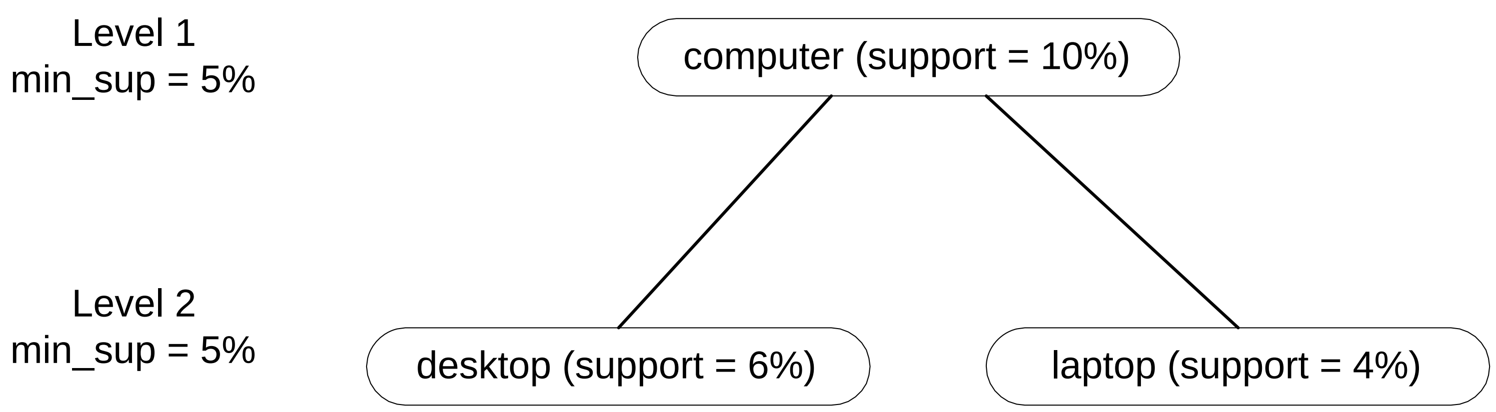


Multi-level Association Rule Mining

- Mining rules
  - In general, top-down approach is employed.
  - Once all frequent itemsets at level 1 are identified, then those at level 2 are found, and so on.
  - For each level, any algorithm to find frequent itemsets can be used.
  - Variations
    - Using uniform support
    - Using reduced minimum support

Multi-level Association Rule Mining

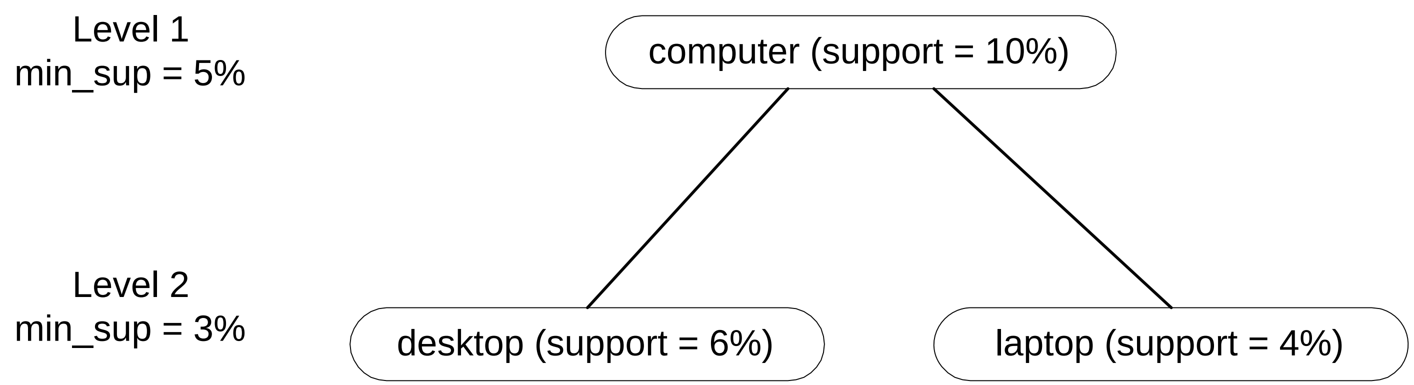
- Using uniform support for all levels
  - Same minimum support is used for all levels.



  - Frequent itemsets: computer, desktop
  - laptop is discarded

Multi-level Association Rule Mining

- Using reduced minimum support at lower levels
  - Each level has its own min. support.
  - Lower levels have smaller min. supports.



  - All (computer, desktop, and laptop) are found as frequent itemsets.

Multi-level Association Rule Mining

- If a node does not satisfy minimum support, its children don’t need to be examined.
- If min. support is set too high, some meaningful rules at low levels may be missed.
- If min. support is set too low, too many uninteresting rules at high levels may be found.

Mining Sequential Patterns

- Given a sequence of elements or events, find a sequential pattern that occurs frequently.
- Applications:
  - Customer shopping sequences
  - Web click streams
  - Program execution sequences
  - Biological sequences
  - Sequence of events in natural or social development

Mining Sequential Patterns

- We will discuss sequential pattern mining from a transactional database.
- The approach described here is based on a sequential pattern mining algorithm called GSP (Generalized Sequential Patterns).
- A *transaction* is a tuple (*sid*, *ts*, *itemset*), where

  sid is a sequence id, which typically is customer id (or *cid*)

  ts is a timestamp

  itemset is a set of items (and items are ordered)
- A *data-sequence* is an ordered list of transactions.
- A (transactional) *database* is a set of data-sequences

Mining Sequential Patterns

- Example database

CID	Time	Items
1	3/25	30
1	3/30	90
2	3/10	10
2	3/15	20,30
2	3/20	40,60,70
3	3/25	30,50,70
4	3/25	30
4	3/30	40,70
4	4/25	90
5	3/12	90

- There are five data-sequences, each corresponding to a customer.
- First data-sequence has two transactions, the second data-sequence has three transactions, …
- Each transaction represents a purchase by a customer of a set of items at a certain time.

Mining Sequential Patterns

- Example database

CID	Time	Items
1	3/25	30
1	3/30	90
2	3/10	10
2	3/15	20,30
2	3/20	40,60,70
3	3/25	30,50,70
4	3/25	30
4	3/30	40,70
4	4/25	90
5	3/12	90

- A sequence is an ordered list of itemsets.
- Sequence examples:
  <{30}, {90}>
  <{30}, {40, 70}>
  <{10}, {40, 60}>

- A *k-sequence* is a sequence consisting of *k* items.
- Above sequences are 2-sequence, 3-sequence, and 3-sequence, respectively

Mining Sequential Patterns

- A sequence A = $<a_1, a_2, \ldots, a_n>$ is a subsequence of another sequence B = $<b_1, b_2, \ldots, b_m>$ if there exist integers $i_1 < i_2 < \ldots < i_n$ such that $a_1 \subseteq b_{i_1}$, $a_1 \subseteq b_{i_2}$, …, $a_1 \subseteq b_{i_n}$
- Example

A = <{3}, {4,5}, {8}>

B = <{7}, {3,8}, {9}, {4,5,6}, {8}>

A is a subsequence of B.

Mining Sequential Patterns

C = <{3}, {15}>

C is a subsequence of D.

D = <{3,5}, {7,8}, {2, 15}>

E = <{3}, {5,8}>

?

E is not a subsequence of F.

F = <{2,3}, {5}, {7,8}>

G = <{3}, {8}>

?

G is not a subsequence of H.

H = <{3,8}>

Mining Sequential Patterns

- Support of a sequence is the number of data sequences that "contain" this sequence.
- A data sequence *d* contains a sequence *s* if *s* is a subsequence of *d*

cid	ts	Itemset
C1	1	cheese
C1	2	butter
C1	15	bread, milk
C2	1	butter, cheese
C2	20	egg, bread
C2	50	milk
C2	50	egg

support of <{cheese}> = 2 (100%)
support of <{egg}> = 1 (50%)
support of <{cheese}, {egg}>
= 1 (50%)
support of <{cheese}, {milk}>
= 2 (100%)
support of <{cheese}, {bread, milk>
= 1 (50%)

Mining Sequential Patterns

- Example database

CID	Time	Items
1	3/25	30
1	3/30	90
2	3/10	10
2	3/15	20,30
2	3/20	40,60,70
3	3/25	30,50,70
4	3/25	10,30
4	3/30	40,70
4	4/25	60,90
5	3/12	90

- Supports of the following sequences are:
  <{30}, {90}>: 2 (or 40%)
  <{30}, {40, 70}>: 2 (or 40%)
  <{10}, {40, 60}>: 1 (or 20%)

- Goal: To discover all sequences with a user-specified minimum support

Mining Sequential Patterns

- Example

CID	Time	Items
1	1	10,30
1	4	80
2	3	10
2	7	30,40
2	20	60,70,80
3	2	30,50
3	8	70,80
4	2	70
4	10	80
5	1	10,20
5	10	30
5	28	70
5	31	80

Mine all frequent sequential patterns with minimum support = 40% (or 2 data-sequences).

L1 (frequent 1-sequences):

<{10}>:3
<{30}>:4
<{70}>:4
<{80}>:5

L2 (frequent 2-sequences)

<{10},{30}>:2
<{10},{70}>:2
<{10},{80}>:3
<{30},{70}>:3
<{30},{80}>:4
<{70},{80}>:2
<{70, 80}>:2

L3 (frequent 3-sequences)

<{10},{30},{70}>:2
<{10},{30},{80}>:2
<{30},{70,80}>:2

Mining Sequential Patterns

- Sequential pattern mining on Weka
  - Weka implemented GSP (with some limitations).
  - Transactional database needs to be converted to an appropriate format as an *arff* file.
  - One transaction per tuple
  - All tuples, so all transactions, have the same number of attributes, and each item is a value of the corresponding attribute.

Mining Sequential Patterns

- Example

```
@relation gsp-books

@attribute day {1, 2, 3}
@attribute 'history' {'revolution', 'civil war'}
@attribute 'biography' {'steinbeck', 'anderson', 'hemingway'}
@attribute 'sports' {'baseball', 'football', 'basketball'}

@data
1,'revolution', 'steinbeck', 'baseball'
1,'civil war', 'anderson', 'football'
2,'revolution', 'anderson', 'baseball'
2,'civil_war', 'anderson', 'football'
3,'revolution', 'hemingway', 'football'
3,'civil war', 'anderson', 'basketball'
```

Mining Sequential Patterns

- Result

  with

  min_sup = 90%

1-sequences

[1] <{revolution}> (3)
[2] <{civil war}> (3)
[3] <{anderson}> (3)
[4] <{football}> (3)

2-sequences

[1] <{revolution}{civil war}> (3)
[2] <{revolution}{anderson}> (3)
[3] <{civil war,anderson}> (3)

3-sequences

[1] <{revolution}{civil war,anderson}> (3)

Mining Sequential Patterns

min_sup = 60%

1-sequences
[1] <{revolution}> (3)
[2] <{civil war}> (3)
[3] <{anderson}> (3)
[4] <{baseball}> (2)
[5] <{football}> (3)

2-sequences
[1] <{revolution}{civil war}> (3)
[2] <{revolution}{anderson}> (3)
[3] <{revolution}{football}> (2)
[4] <{civil war,anderson}> (3)
[5] <{revolution,baseball}> (2)
[6] <{baseball}{civil war}> (2)
[7] <{baseball}{anderson}> (2)
[8] <{baseball}{football}> (2)
[9] <{civil war,football}> (2)
[10] <{anderson,football}> (2)

3-sequences
[1] <{revolution}{civil war,anderson}> (3)
[2] <{revolution}{civil war,football}> (2)
[3] <{revolution}{anderson,football}> (2)
[4] <{civil war,anderson,football}> (2)
[5] <{revolution,baseball}{civil war}> (2)
[6] <{revolution,baseball}{anderson}> (2)
[7] <{revolution,baseball}{football}> (2)
[8] <{baseball}{civil war,anderson}> (2)
[9] <{baseball}{civil war,football}> (2)
[10] <{baseball}{anderson,football}> (2)

4-sequences
[1] <{revolution}{civil war,anderson,football}> (2)
[2] <{revolution,baseball}{civil war,anderson}> (2)
[3] <{revolution,baseball}{civil war,football}> (2)
[4] <{revolution,baseball}{anderson,football}> (2)
[5] <{baseball}{civil war,anderson,football}> (2)

5-sequences
[1] <{revolution,baseball}{civil war,anderson,football}> (2)

Associative Classification

- Each sample (or tuple) is considered as a transaction.
- An (attribute, value) pair is an item.
- Frequent itemsets are mined using an association rule mining algorithm.
- Strong rules are mined from the frequent itemsets, which satisfy the minimum support and minimum confidence thresholds.
- We only use rules which has class attribute in the consequent.
- Rules are organized to form a rule-based classifier.

Associative Classification

- Example dataset

outlook	temperature	humidity	windy	play
sunny	hot	high	F	N
sunny	hot	high	T	N
overcast	hot	high	F	Y
rainy	mild	high	F	Y
rainy	cool	normal	F	Y
rainy	cool	normal	T	N
overcast	cool	normal	T	Y
sunny	mild	high	F	N
sunny	cool	normal	F	Y
rainy	mild	normal	F	Y
sunny	mild	normal	T	Y
overcast	mild	high	T	Y
overcast	hot	normal	F	Y
rainy	mild	high	T	N

Associative Classification

- For association rule mining, each tuple is considered as a transaction and each (attribute, value) pair becomes an itme as follows:

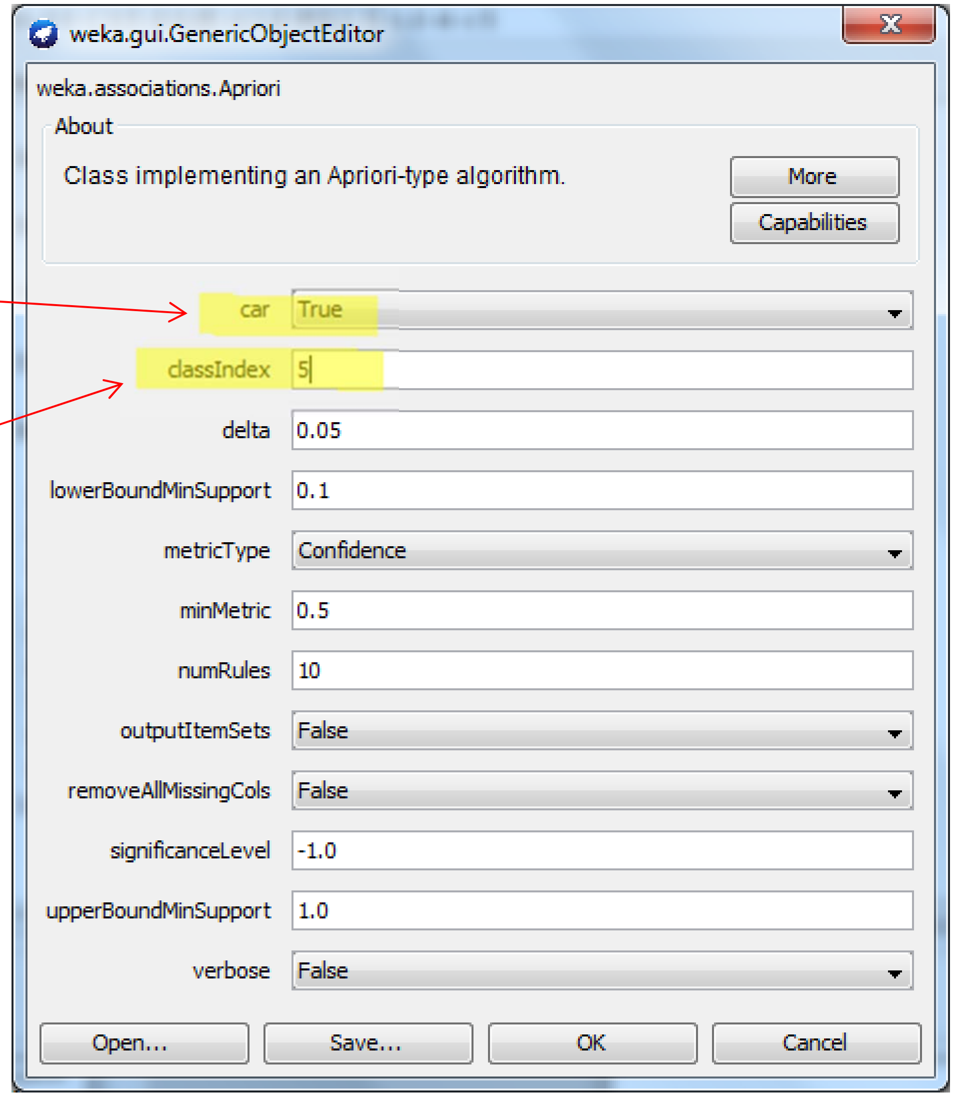
TID	items
1	outlook=sunny, temperature=hot, humidity=high, windy=F, play=N
2	outlook=sunny, temperature=hot, humidity=high, windy=T, play=N
3	outlook=overcast, temperature=hot, humidity=high, windy=F, play=Y
4	outlook=rainy, temperature=mild, humidity=high, windy=F, play=Y
5	outlook=rainy, temperature=cool, humidity=normal, windy=F, play=Y
6	...

Associative Classification

- On Weka,

Set car to True

Specify the index of the class attribute



Associative Classification

- Result

  Top ten rules by confidence

```
Generated sets of large itemsets:

Size of set of large itemsets L(1): 11

Size of set of large itemsets L(2): 4

Best rules found:

 1. outlook=overcast 4 ==> play=yes 4    conf:(1)
 2. humidity=normal windy=FALSE 4 ==> play=yes 4    conf:(1)
 3. outlook=sunny humidity=high 3 ==> play=no 3    conf:(1)
 4. outlook=rainy windy=FALSE 3 ==> play=yes 3    conf:(1)
 5. humidity=normal 7 ==> play=yes 6    conf:(0.86)
 6. windy=FALSE 8 ==> play=yes 6    conf:(0.75)
 7. temperature=cool 4 ==> play=yes 3    conf:(0.75)
 8. temperature=cool humidity=normal 4 ==> play=yes 3    conf:(0.75)
 9. temperature=mild 6 ==> play=yes 4    conf:(0.67)
10. outlook=sunny 5 ==> play=no 3    conf:(0.6)
```

References

- Han, J., Kamber, M., Pei, J., "Data mining: concepts and techniques," 3rd Ed., Morgan Kaufmann, 2012
- http://www.cs.illinois.edu/~hanj/bk3/
- Multilevel association rule mining: Han, J., Kamber, M., Pei, J., "Data mining: concepts and techniques," 3rd Ed., Morgan Kaufmann, 2012, pp. 283 – 287.
- Sequential pattern mining: R. Srikant and R. Agrawal, "Mining sequential patterns: generalization and performance improvements," Proc. 5th Int'l Conf. on Extending Database Technology: Advances in Database Technology, pp. 3 – 17.
- Associative classification: Han, J., Kamber, M., Pei, J., "Data mining: concepts and techniques," 3rd Ed., Morgan Kaufmann, 2012, pp. 416 – 419.