Data mining algorithms: Association rules

Motivation and terminology

1. Data mining perspective
    - Market basket analysis: looking for associations between items in the shopping cart.
    - Rule form: Body => Head [support, confidence]
    - Example: buys(x, “diapers”) => buys(x, “beers”) [0.5%, 60%]
2. Machine Learning approach: treat every possible combination of attribute values as a separate class, learn rules using the rest of attributes as input and then evaluate them for support and confidence. Problem: computationally intractable (too many classes and consequently, too many rules).
3. Basic terminology:
    1. Tuples are *transactions*, attribute-value pairs are *items.*
    2. *Association rule*: {A,B,C,D,...} => {E,F,G,...}, where A,B,C,D,E,F,G,... are items.
    3. *Confidence* (accuracy) of A => B : P(B|A) = (# of transactions containing both A and B) / (# of transactions containing A).
    4. *Support* (coverage) of A => B : P(A,B) = (# of transactions containing both A and B) / (total # of transactions)
    5. We looking for rules that exceed pre-defined support (*minimum support*) and have high confidence.

Example

1. Load the weather data in Weka (click on **Preprocess** and then on **Open file...** weather.nominal.arff). The data are shown below in tabular form.

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes

rainy	mild	high	true	no

Click on **Associate** and then on **Start.** You get the following 10 association rules in **Associator output** window:

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

Basic idea: item sets

1. Item set: sets of all items in a rule (in both LHS and RHS).
2. Item sets for weather data: 12 one-item sets (3 values for outlook + 3 for temperature + 2 for humidity + 2 for windy + 2 for play), 47 two-item sets, 39 three-item sets, 6 four-item sets and 0 five-item sets (with minimum support of two).

One-item sets	Two-item sets	Three-item sets	Four-item sets
Outlook = Sunny (5) Temperature = Cool (4) ...	Outlook = Sunny Temperature = Mild (2) Outlook = Sunny Humidity = High (3) ...	Outlook = Sunny Temperature = Hot Humidity = High (2) Outlook = Sunny Humidity = High Windy = False (2) ...	Outlook = Sunny Temperature = Hot Humidity = High Play = No (2) Outlook = Rainy Temperature = Mild Windy = False Play = Yes (2) ...

3. Generating rules from item sets. Once all item sets with minimum support have been generated, we can turn them into rules.

- Item set: {Humidity = Normal, Windy = False, Play = Yes} (support 4).
- Rules: for a **n-item set there are (2^n -1) possible rules**, chose the ones with highest confidence. For example:
  `If Humidity = Normal and Windy = False then Play = Yes (4/4)`
  `If Humidity = Normal and Play = Yes then Windy = False (4/6)`

```
If Windy = False and Play = Yes then Humidity = Normal (4/6)
If Humidity = Normal then Windy = False and Play = Yes (4/7)
If Windy = False then Humidity = Normal and Play = Yes (4/8)
If Play = Yes then Humidity = Normal and Windy = False (4/9)
If True then Humidity = Normal and Windy = False and Play = Yes
(4/12)
```

Generating item sets efficiently

1. Frequent item sets: item sets with the desired minimal support.
2. Observation: if {A,B} is a frequent item set, then both A and B are frequent item sets too. The inverse, however is not true (find a counter-example).
3. Basic idea (Apriori algorithm):
    - Find *all* n-item sets. Example (n=2): L2 = { {A,B}, {A,D}, {C,D}, {B,D} }
    - Generate (n+1)-item sets by merging n-item sets. L3 = { {A,B,C}, {A,C,D}, {A,B,D}, {B,C,D} }.
    - Test the newly generated (n+1)-items sets for minimum support.
        - Eliminate {A,B,C}, {A,C,D} and {B,C,D} because they contain non-frequent 2-item sets (which ones?).
        - Test the remaining item sets for minimal support by counting their occurrences in data.
    - Increment n and continue until no more frequent item sets can be generated.
    - Test step uses a *hash table* with all n-item sets: remove an item from the (n+1)-item set and check if it is in the hash table.

Generating rules efficiently

1. Brute-force method (for small item sets):
    - Generate all possible subsets of an item sets, excluding the empty set $(2^n - 1)$ and use them as rule consequents (the remaining items form the antecedents).
    - Compute the confidence: divide the support of the item set by the support of the antecedent (get it from the hash table).
    - Select rules with high confidence (using a threshold).
2. Better way: iterative rule generation within minimal accuracy.
    - Observation: if an n-consequent rule holds then all corresponding (n-1)-consequent rules hold as well.
    - Algorithm: generate n-consequent candidate rules from (n-1)-consequent rules (similar to the algorithm for the item sets).
3. Weka's approach (default settings for Apriori): generate best 10 rules. Begin with a minimum support 100% and decrease this in steps of 5%. Stop when generate 10 rules or the support falls below 10%. The minimum confidence is 90%.

Advanced association rules

1. *Multi-level* association rules: using concept hierarchies.
    - Example: no frequent item sets.

A	B	C	D	...
1	0	1	0	...
0	1	0	1	...
...	...	...	...	...

- Assume now that A and B are children of A&B, and C and D are children of C&D in concept hierarchies. Assume also that A&B and C&D aggregate the values for their children. Then {A&B, C&D} will be a frequent item set with support 2.

2. Approaches to mining multi-level association rules:
   - Using uniform support: same minimum support for all levels in the hierarchies: top-down strategy.
   - Using reduced minimum support at lower levels: various approaches to define the minimum support at lower levels.
3. Interpretation of association rules:
   - *Single-dimensional* rules: single predicate. Example: buys(x, “diapers”) => buys(x, “beers”). Create a table with as many columns as possible values for the predicate. Consider these values as binary attributes (0,1) and when creating the item sets ignore the 0's.

diapers	beers	milk	bread	...
1	1	0	1	...
1	1	1	0	...
...	...	...	...	...

   - *Multidimensional* association rules: multiple predicates. Example: age(x, 20) and buys(x, computer) => buys(x, computer_games). Mixed-type attributes. Problem: the algorithms discussed so far cannot handle numeric attributes.

age	computer	computer_games	...
20	1	1	...
35	1	0	...
...	...	...	...

4. Handling numeric attributes.
   - Static discretization: discretization based on predefined ranges.
   - Discretization based on the distribution of data: binning. Problem: grouping together very distant values.
   - Distance-based association rules:
     - cluster values by distance to generate clusters (intervals or groups of nominal values).
     - search for frequent cluster sets.

- Approximate Association Rule Mining. Read the paper by Nayak and Cook.

Correlation analysis

1. High support and high confidence rules are not necessarily interesting. Example:
   - Assume that A occurs in 60% of the transactions, B - in 75% and both A and B - in 40%.
   - Then the association A => B has support 40% and confidence 66%.
   - However, P(B)=75%, higher than P(B|A)=66%.
   - In fact, A and B are negatively correlated, corr(A,B)=0.4/(0.6*0.75)=0.89<1
2. Support-confidence framework: an estimate of the *conditional probability* of B given A. We need a measure for the certainty of the implication A => B, that is, whether A implies B and to what extend.
3. Correlation between occurrences of A and B:
   - corr(A,B) = P(A,B)/(P(A)P(B))
   - corr(A,B)<1 => A and B are negatively correlated.
   - corr(A,B)>1 => A and B are positively correlated.
   - corr(A,B)=1 => A and B are independent.
4. Contingency table:

	outlook=sunny	outlook<>sunny	Row total
play=yes	2	7	9
play=no	3	2	5
Column total	5	9	14

   - if outlook=sunny then play=yes [support=14%, confidence=40%].
   - corr(outlook=sunny,play=yes) = (2/14)/[(5/14)*(9/14)] = 0.62 < 1 => negaive correlation.
   - if outlook=sunny then play=no [support=21%, confidence=60%].
   - corr(outlook=sunny,play=no) = (3/14)/[(5/14)*(5/14)] = 1.68 > 1 => positive correlation.