

Association Rule Mining by Weka Explorer

Zhi Shi, Carl Yang

Weka

- A collection of machine learning algorithms for data mining tasks.
- Contains tools for data pre-processing, classification, regression, clustering, **association rules**, and visualization.
- <http://www.cs.waikato.ac.nz/ml/weka/>



Machine Learning Group at the University of Waikato

[Project](#) [Software](#) [Book](#) [Publications](#) [People](#) [Related](#)

Weka 3: Data Mining Software in Java

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

Found only on the islands of New Zealand, the Weka is a flightless bird with an inquisitive nature. The name is pronounced like **this**, and the bird sounds like **this**.

Weka is open source software issued under the **GNU General Public License**.

Yes, it is possible to apply Weka to **big data**!

Data Mining with Weka is a 5 week MOOC, which was held first in late 2013. Check out the [MOOC site](#) for video lectures and details on how to enrol into this course and a new, advanced Weka course.

Getting started

- [Requirements](#)
- [Download](#)
- [Documentation](#)
- [FAQ](#)
- [Getting Help](#)

Further information

- [Citing Weka](#)
- [Datasets](#)
- [Related Projects](#)
- [Miscellaneous Code](#)
- [Other Literature](#)

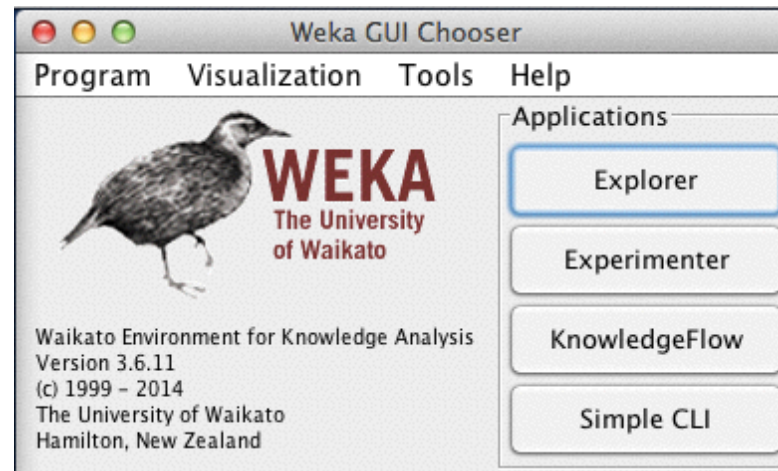
Developers

- [Development](#)
- [History](#)
- [Subversion](#)
- [Contributors](#)



Weka

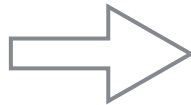
- Install GUI: <http://www.cs.waikato.ac.nz/ml/weka/downloading.html> (Stable book 3rd ed. version)
- We will use Explorer.



Weka

- Load data: import weka-3-6-11/data/supermarket.arff
- ARFF: Attribute domain + Data domain. Represent a transaction database.
- Attribute domain: Attribute name 'department1' followed by value range { t }
- Data domain: A vector that represent presence of one item (t) or absence(?)
- More information: <http://www.cs.waikato.ac.nz/ml/weka/arff.html>.

Trans	Itemsets
T1	Department1, grocery misc,...
T2	baby needs, baking needs,...
...



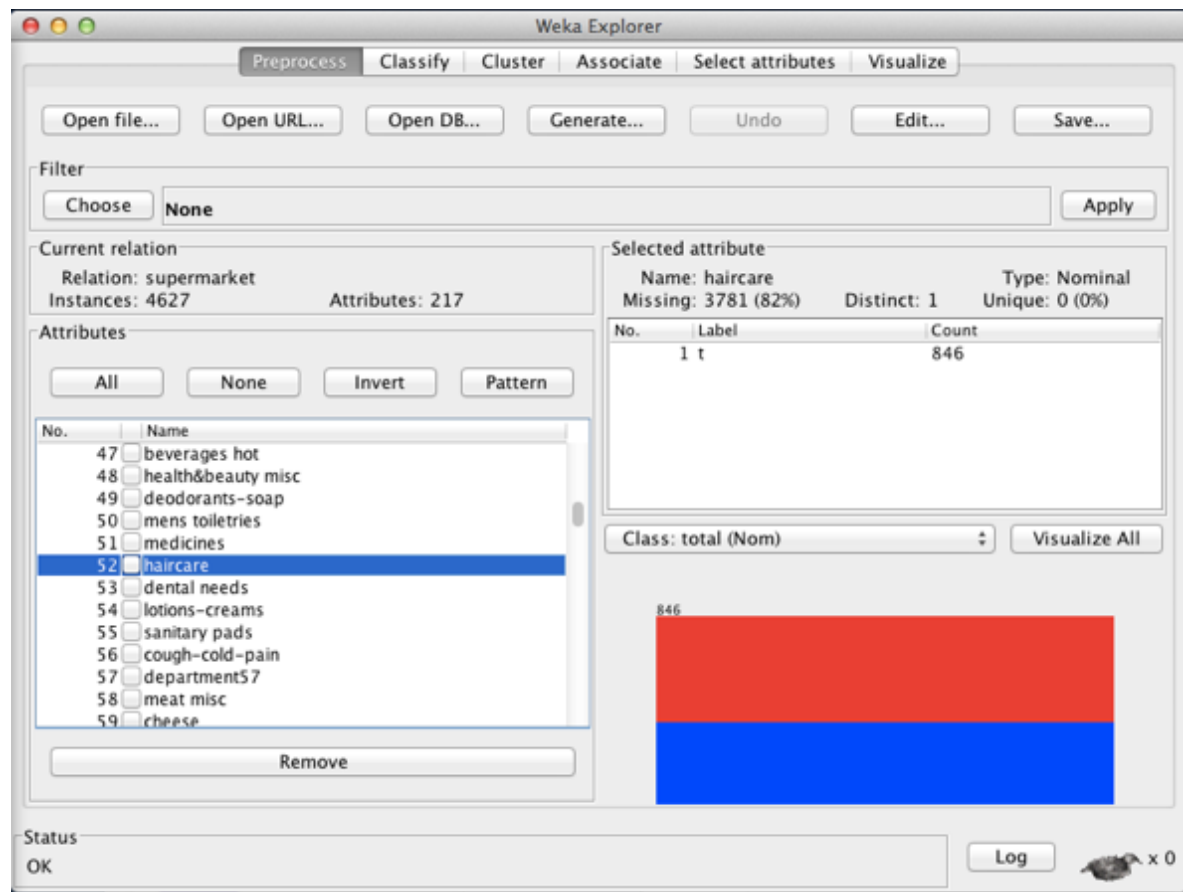
```

1 @relation supermarket
2 @attribute 'department1' { t}
3 @attribute 'department2' { t}
4 @attribute 'department3' { t}
5 @attribute 'department4' { t}
6 @attribute 'department5' { t}
7 @attribute 'department6' { t}
8 @attribute 'department7' { t}
9 @attribute 'department8' { t}
10 @attribute 'department9' { t}
11 @attribute 'grocery misc' { t}
12 @attribute 'department11' { t}
13 @attribute 'baby needs' { t}
14 @attribute 'bread and cake' { t}
15 @attribute 'baking needs' { t}
16 @attribute 'coupons' { t}
17 @attribute 'juice-sat-cord-ms' { t}
18 @attribute 'tea' { t}
19 @attribute 'biscuits' { t}
20 @attribute 'canned fish-meat' { t}
21 @attribute 'canned fruit' { t}

```

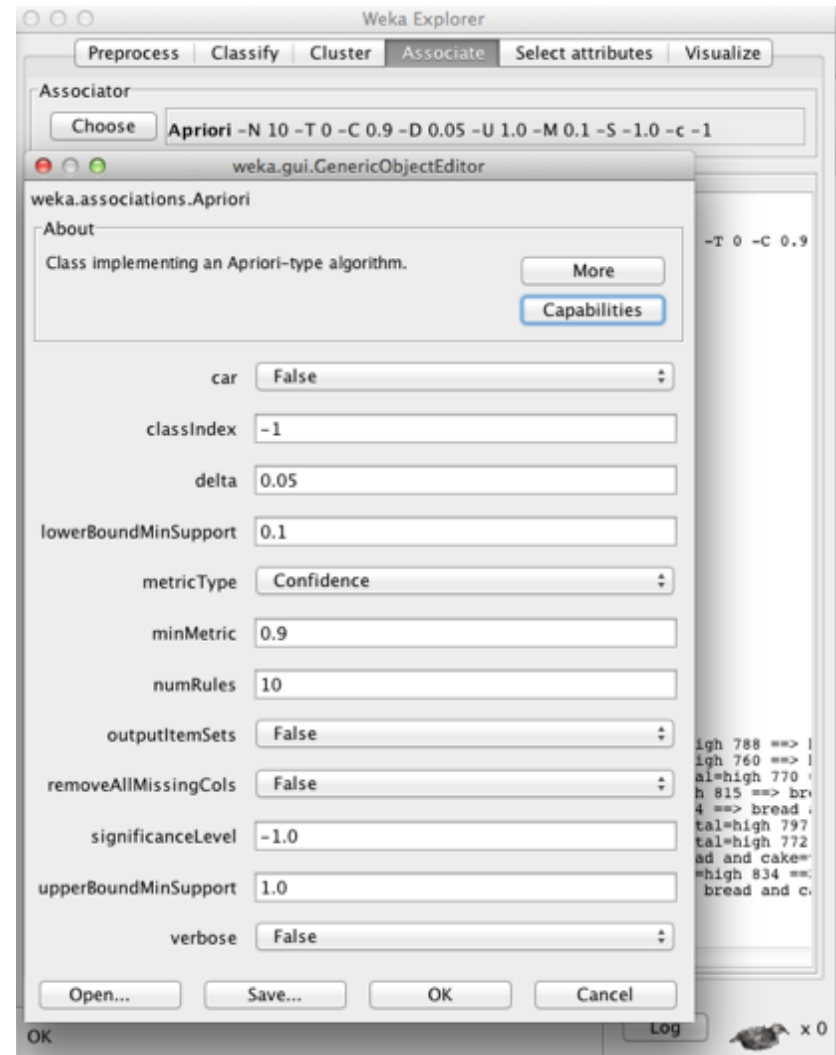
Weka

- Load data



Apriori

- Choose association rule mining: 'Associate'
- Set up parameters for Apriori
 - metricType: confidence
 - minMetric: 0.9 -> min_conf = 0.9
 - numRules: 10



Recap

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk

- **Support(count)**: Occurrence of an itemset X (#X)
 - Support count(Beer) = 3
 - Support count(Diaper) = 4
 - Support count(Beer, Diaper) = 3
- **Support of X** : Fraction of all transactions that contains X ($\#X/\#\text{Transactions}$)
 - Support(Beer) = $3/5 = 0.6$
 - Support(Diaper) = $4/5 = 0.8$
 - Support(Beer, Diaper) = $3/5 = 0.6$
- **Confidence of X, Y**: Conditional probability that a transaction having X also contains Y ($\#XY/\#X$)
 - Confidence(Beer, Diaper) = $3/3 = 1$
 - Confidence(Diaper, Beer) = $3/4 = 0.75$
- **Association Rule**: When buy X, how likely will also buy Y?
 - Represented as $X \rightarrow Y$, support(XY), Confidence(X, Y)
 - Beer \rightarrow Diaper (0.6, 1)
 - Diaper \rightarrow Beer (0.6, 0.75)

Result

The screenshot shows the Weka Explorer application window. The 'Associate' tab is selected in the top menu bar. The 'Associator' section on the left shows the 'Apriori' algorithm with parameters: -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1. The 'Start' button is highlighted. Below the 'Start' button is a 'Result list (right-click for...)' showing a list of runs. The run at 06:47:13 - Apriori is selected and highlighted in blue. The 'Associator output' section on the right displays the results of the Apriori run, including run information, model details, and a list of 10 best rules found.

Weka Explorer

Preprocess | Classify | Cluster | **Associate** | Select attributes | Visualize

Associator

Choose **Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1**

Start Stop

Result list (right-click for...)

- 06:30:59 - Apriori
- 06:32:23 - Apriori
- 06:32:58 - Apriori
- 06:34:42 - Apriori
- 06:39:51 - Apriori
- 06:42:03 - Apriori
- 06:42:07 - Apriori
- 06:42:38 - Apriori
- 06:42:59 - Apriori
- 06:44:06 - Apriori
- 06:44:22 - Apriori
- 06:44:45 - Apriori
- 06:45:00 - Apriori
- 06:47:13 - Apriori**
- 06:47:44 - FPGrowth
- 06:48:13 - FPGrowth

Associator output

```
=== Run information ===
Scheme:      weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Relation:    supermarket
Instances:   4627
Attributes:  217
[list of attributes omitted]
=== Associator model (full training set) ===

Apriori
=====

Minimum support: 0.15 (694 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 17

Generated sets of large itemsets:

Size of set of large itemsets L(1): 44
Size of set of large itemsets L(2): 380
Size of set of large itemsets L(3): 910
Size of set of large itemsets L(4): 633
Size of set of large itemsets L(5): 105
Size of set of large itemsets L(6): 1

Best rules found:

1. biscuits=t frozen foods=t fruit=t total=high 788 ==> bread and cake=t 723    conf:(0.92)
2. baking needs=t biscuits=t fruit=t total=high 760 ==> bread and cake=t 696    conf:(0.92)
3. baking needs=t frozen foods=t fruit=t total=high 770 ==> bread and cake=t 705    conf:(0.92)
4. biscuits=t fruit=t vegetables=t total=high 815 ==> bread and cake=t 746    conf:(0.92)
5. party snack foods=t fruit=t total=high 854 ==> bread and cake=t 779    conf:(0.91)
6. biscuits=t frozen foods=t vegetables=t total=high 797 ==> bread and cake=t 725    conf:(0.91)
7. baking needs=t biscuits=t vegetables=t total=high 772 ==> bread and cake=t 701    conf:(0.91)
8. biscuits=t fruit=t total=high 954 ==> bread and cake=t 866    conf:(0.91)
9. frozen foods=t fruit=t vegetables=t total=high 834 ==> bread and cake=t 757    conf:(0.91)
10. frozen foods=t fruit=t total=high 969 ==> bread and cake=t 877    conf:(0.91)
```


Result

• === Run information ===

- Scheme: weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
- Relation: supermarket
- Instances: 4627
- Attributes: 217
- [list of attributes omitted]
- === Associator model (full training set) ===

parameter setting

- Apriori
- =====

- Minimum support: 0.15 (694 instances)
- Minimum metric <confidence>: 0.9

metric

support count of (biscuits, frozen food, fruit, total=high)

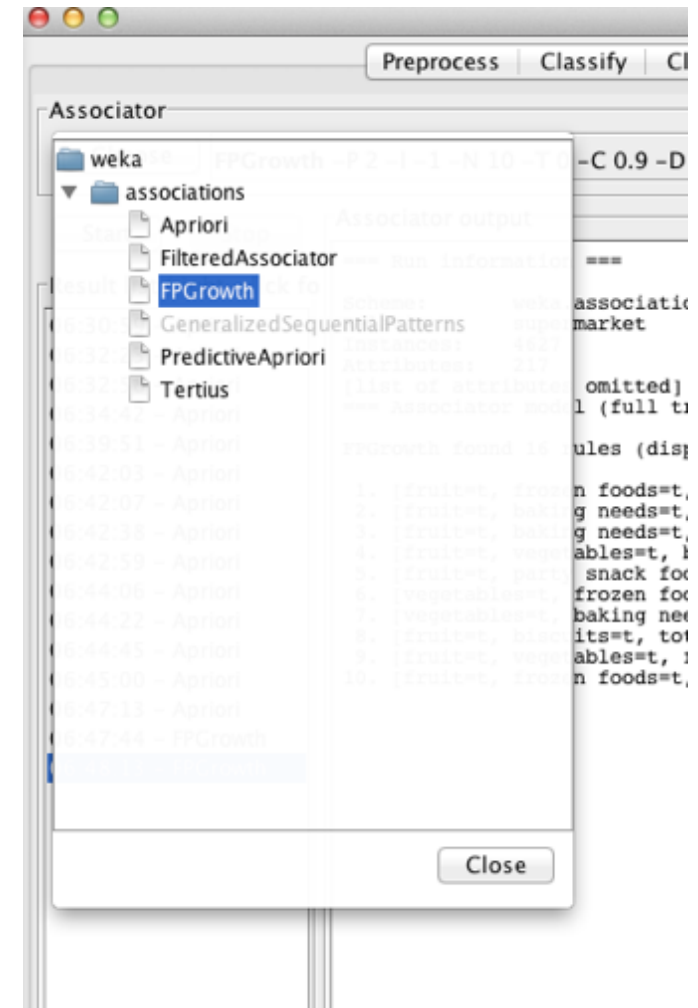
support count of (biscuits, frozen food, fruit, total=high, bread and cake)

• Best rules found:

1. biscuits=t frozen foods=t fruit=t total=high 788 ==> bread and cake=t 723 conf:(0.92)
2. baking needs=t biscuits=t fruit=t total=high 760 ==> bread and cake=t 696 conf:(0.92)
3. baking needs=t frozen foods=t fruit=t total=high 770 ==> bread and cake=t 705 conf:(0.92)
4. biscuits=t fruit=t vegetables=t total=high 815 ==> bread and cake=t 746 conf:(0.92)
5. party snack foods=t fruit=t total=high 854 ==> bread and cake=t 779 conf:(0.91)
6. biscuits=t frozen foods=t vegetables=t total=high 797 ==> bread and cake=t 725 conf:(0.91)
7. baking needs=t biscuits=t vegetables=t total=high 772 ==> bread and cake=t 701 conf:(0.91)
8. biscuits=t fruit=t total=high 954 ==> bread and cake=t 866 conf:(0.91)
9. frozen foods=t fruit=t vegetables=t total=high 834 ==> bread and cake=t 757 conf:(0.91)
10. frozen foods=t fruit=t total=high 969 ==> bread and cake=t 877 conf:(0.91)

FP-growth

- Choose 'FPGrowth'
- Set parameters for FP-growth
 - metricType: confidence
 - minMetric: 0.9 -> min_conf = 0.9
 - numRules: 10



Result

- === Run information ===
- Scheme: weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1
- Relation: supermarket
- Instances: 4627
- Attributes: 217
- [list of attributes omitted]
- === Associator model (full training set) ===
- FPGrowth found 16 rules (displaying top 10)
- 1. [fruit=t, frozen foods=t, biscuits=t, total=high]: 788 ==> [bread and cake=t]: 723 <conf:(0.92)> lift:(1.27) lev:(0.03) conv:(3.35)
- 2. [fruit=t, baking needs=t, biscuits=t, total=high]: 760 ==> [bread and cake=t]: 696 <conf:(0.92)> lift:(1.27) lev:(0.03) conv:(3.28)
- 3. [fruit=t, baking needs=t, frozen foods=t, total=high]: 770 ==> [bread and cake=t]: 705 <conf:(0.92)> lift:(1.27) lev:(0.03) conv:(3.27)
- 4. [fruit=t, vegetables=t, biscuits=t, total=high]: 815 ==> [bread and cake=t]: 746 <conf:(0.92)> lift:(1.27) lev:(0.03) conv:(3.26)
- 5. [fruit=t, party snack foods=t, total=high]: 854 ==> [bread and cake=t]: 779 <conf:(0.91)> lift:(1.27) lev:(0.04) conv:(3.15)
- 6. [vegetables=t, frozen foods=t, biscuits=t, total=high]: 797 ==> [bread and cake=t]: 725 <conf:(0.91)> lift:(1.26) lev:(0.03) conv:(3.06)
- 7. [vegetables=t, baking needs=t, biscuits=t, total=high]: 772 ==> [bread and cake=t]: 701 <conf:(0.91)> lift:(1.26) lev:(0.03) conv:(3.01)
- 8. [fruit=t, biscuits=t, total=high]: 954 ==> [bread and cake=t]: 866 <conf:(0.91)> lift:(1.26) lev:(0.04) conv:(3)
- 9. [fruit=t, vegetables=t, frozen foods=t, total=high]: 834 ==> [bread and cake=t]: 757 <conf:(0.91)> lift:(1.26) lev:(0.03) conv:(3)
- 10. [fruit=t, frozen foods=t, total=high]: 969 ==> [bread and cake=t]: 877 <conf:(0.91)> lift:(1.26) lev:(0.04) conv:(2.92)

Assignment 3

- Give a set of paper titles under 5 topics(Data Mining, Machine Learning, Information Retrieval, Database, Theory), mine “meaningful” frequent words/phrases.
- “data mining” is meaningful. But “based algorithm” is not. But they are all frequent patterns. Frequent phrases does not necessarily mean “meaningful”.
- You need to write Apriori or FPGrowth to mine frequent words/phrases
- Also, it is interesting to see closed & max patterns, association rules.
- Is there any methods that could get better phrases?

PaperID	Title
7600	The Automatic Acquisition of Proof Methods
85825	Frequent pattern discovery with memory

Assignment 3

- Input file: topic0.txt~topic4.txt, vocab.txt
- Each line is the words that most likely belongs to this topic.

```
45 40 41
71 74 73
115 98 117 118 114 116
126 124
134 25 130 132 136 133
161
185
```

Name topic-0.txt

```
0 3 1 2
7 5
25 26 28 23 27 24
30 29 34 31 32
38
43 44 42
47 48 46 49
51 53 52
58 57 26 59
64
```

Name topic-3.txt

```
0      automatic
1      acquisition
2      proof
3      method
4      philosophical
5      formal
6      learning
7      theory
8      low
9      cost
```

Name vocab.txt

Assignment 3

- In the introduction part, we first describe how we generate the input files. You don't need to write code for these steps. But it helps you understand the background.
- Step 1: Implement frequent pattern mining algorithm and find frequent patterns in 5 files
- Step 2: Find max/closed patterns
- Step 3: Find association rules by Weka

```
#SUP: 1226 web
#SUP: 1211 information
#SUP: 1114 retrieval
#SUP: 863 based
#SUP: 757 system
#SUP: 707 search
#SUP: 564 document
#SUP: 490 language
```

FPGrowth found 8 rules (displaying top 8)

```
1. [structure=1]: 209 ==> [decomposition=1]: 194  <conf:(0.92)>
2. [method=1, matrix=1]: 159 ==> [path=1]: 123  <conf:(0.78)>
3. [method=1, path=1]: 159 ==> [matrix=1]: 123  <conf:(0.78)>
4. [path=1]: 336 ==> [matrix=1]: 233  <conf:(0.69)>
5. [massive=1]: 211 ==> [discovery=1]: 141  <conf:(0.67)>
6. [matrix=1]: 416 ==> [path=1]: 233  <conf:(0.56)>
7. [improved=1]: 227 ==> [method=1]: 127  <conf:(0.56)>
8. [matrix=1, path=1]: 233 ==> [method=1]: 123  <conf:(0.56)>
```

Thanks!