Knowledge Discovery and Data Mining

Unit # 12

Association Rules in Clementine

- Apriori
- GRI
- CARMA
- Sequence

Apriori

- The Apriori node discovers association rules in the data. Association rules are statements of the form
  - if antecedent(s) then consequent(s)
- For example, "if a customer purchases a razor and after shave, then that customer will purchase shaving cream with 80% confidence."
- Apriori extracts a set of rules from the data, pulling out the rules with the highest information content.
- Apriori offers five different methods of selecting rules and uses a sophisticated indexing scheme to efficiently process large datasets.

Apriori (Cont'd)

- Requirements. To create an Apriori ruleset, you need one or more In fields and one or more Out fields. Input and output fields (those with direction In, Out, or Both) must be symbolic. Fields with direction None are ignored. Fields types must be fully instantiated before executing the node. Data can be in tabular or transactional format. See Tabular versus Transactional Data for more information.
- Strengths. For large problems, Apriori is generally faster to train than GRI. It also has no arbitrary limit on the number of rules that can be retained and can handle rules with up to 32 preconditions. Apriori offers five different training methods, allowing more flexibility in matching the data mining method to the problem at hand.

GRI

- The Generalized Rule Induction (GRI) node discovers association rules in the data. Association rules are statements in the form
  - if antecedent(s) then consequent(s)
- For example, if a customer purchases a razor and aftershave lotion, then you can be 80% confident that the customer will also purchase shaving cream.
  - if razor and aftershave lotion then shaving cream
- GRI extracts a set of rules from the data, pulling out the rules with the highest information content.
- Information content is measured using an index that takes both the generality (support) and accuracy (confidence) of rules into account.

GRI (Cont'd)

- Requirements. To create GRI association rules, you need one or more In fields and one or more Out fields. Output fields (those with direction Out or Both) must be symbolic. Fields with direction None are ignored. Fields types must be fully instantiated before executing the node. In contrast to Apriori and CARMA, which read both tabular and transactional data, GRI requires data be in tabular format. See Tabular versus Transactional Data for more information.
- Strengths. Association rules are usually fairly easy to interpret, in contrast to other methods such as neural networks. Rules in a set can overlap such that some records may trigger more than one rule. This allows the ruleset to make rules more general than is possible with a decision tree. The GRI node can also handle multiple output fields. In contrast to Apriori, GRI can handle numeric as well as symbolic input fields.

CARMA

- The CARMA node uses an association rules discovery algorithm to discover association rules in the data. Association rules are statements in the form
  - if antecedent(s) then consequent(s)
- For example, if a Web customer purchases a wireless card and a high-end wireless router, the customer is also likely to purchase a wireless music server if offered.
- The CARMA model extracts a set of rules from the data without requiring you to specify In (predictor) or Out (target) fields.
- This means that the rules generated can be used for a wider variety of applications.

CARMA (Cont'd)

- For example, you can use rules generated by this node to find a list of products or services (antecedents) whose consequent is the item that you want to promote this holiday season.
- Using Clementine, you can determine which clients have purchased the antecedent products and construct a marketing campaign designed to promote the consequent product.

CARMA (Cont'd)

- Requirements. In contrast to GRI and Apriori, the CARMA node does not require In fields or Out fields. This is integral to the way the algorithm works and is equivalent to building an Apriori model with all fields set to Both. You can constrain which items appear only as antecedents or consequents by filtering the model after it is built. For example, you can use the model browser to find a list of products or services (antecedents) whose consequent is the item that you want to promote this holiday season.
- To create a CARMA ruleset, you need to specify an ID field and one or more content fields. The ID field can have any direction or type. Fields with direction None are ignored. Fields types must be fully instantiated before executing the node. Like Apriori, data may be in tabular or transactional format. See Tabular versus Transactional Data for more information.

CARMA (Cont'd)

- Strengths. The CARMA node is based on the CARMA association rules algorithm. In contrast to Apriori and GRI, the CARMA node offers build settings for rule support (support for both antecedent and consequent) rather than antecedent support. CARMA also allows rules with multiple consequents. Like Apriori, models generated by a CARMA node can be inserted into a data stream to create predictions. See Overview of Generated Models for more information.

Sequence

- The Sequence node discovers patterns in sequential or time-oriented data, in the format bread ⇨ cheese.
- The elements of a sequence are item sets that constitute a single transaction.
- For example, if a person goes to the store and purchases bread and milk and then a few days later returns to the store and purchases some cheese, that person's buying activity can be represented as two item sets. The first item set contains bread and milk, and the second one contains cheese.
- A sequence is a list of item sets that tend to occur in a predictable order.
- The sequence node detects frequent sequences and creates a generated model node that can be used to make predictions.

Sequence (Cont'd)

- Requirements. To create a Sequence ruleset, you need to specify an ID field, an optional time field, and one or more content fields. Note that these settings must be made on the Fields tab of the Modeling node; they cannot be read from an upstream Type node. The ID field can have any direction or type. If you specify a time field, it can have any direction but must be numeric, date, time, or timestamp. If you do not specify a time field, the Sequence node will use an implied timestamp, in effect using row numbers as time values. Content fields can have any type and direction, but all content fields must be of the same type. If they are numeric, they must be integer ranges (not real ranges).

Sequence (Cont'd)

- Strengths. The Sequence node is based on the CARMA association rules algorithm, which uses an efficient two-pass method for finding sequences. In addition, the generated model node created by a Sequence node can be inserted into a data stream to create predictions. The generated model node can also generate SuperNodes for detecting and counting specific sequences and for making predictions based on specific sequences.