The New Iris Data – Modular Data Generators

Iris Adä
Michael R. Berthold

[Iris.Adae@Uni-Konstanz.de]

ACM SIGKDD 2010, Washington
Outline

• Motivation
  – Why generate data?
  – What do current generators?
  – Why use modular generators?

• Modular Data Generation
  – Creating clusters (Demo)
  – Creating association rules
  – Combining two motifs

• Outlook
Motivation

• Reasons to generate artificial data
  – Teaching
  – Method Development, Validation, Testing
  – Demonstration
Motivation

• **Current tools**

  – Generate data for one method/purpose (cluster, association rules, own algorithm evaluating)

  – Difficult to
    • extend,
    • modify or
    • reuse for other purposes
Modular Data Generators

- Splitting the generation into small modules
Modular Data Generators

• Splitting the generation into small modules

• Each module can then be modified, exchanged and/or extended to meet new requirements

• Integrated in an existing data analysis platform (open source, free, reusable, ...)

[Image of KNIME logo]
Demo : Creating cluster with MDG’s

The New Iris Data – Modular Data Generators
Creating cluster using MDG’s

Empty Table Creator

Creates an empty file. Just lines with RowIDs.

Dialog Options

- **Number**: the number of rows
- **Prefix**: the prefix of the rowkey (before a unique number)
- **Skip**: a random number of rowkeys between 0 and this value will be skipped (if less than 0 no keys will be skipped)
- **Seed**: the random seed

Ports

- **Output Ports**
  - 0 Empty table with rowkeys.
Creating cluster using MDG’s
Creating cluster using MDG's

Random Label Assigner

Assigns the labels based on the probabilities to the rows. Here we use the classnames and the probabilities given in the dialog to assign the new class column. Category with empty name or probability less or equal 0 will be ignored.

Dialog Options

- **Column Name**: the name of the new column
- **seed**: the random seed to get a deterministic result
- **Name of column**: The name of the new column
- **Probability**: the probability of this category

Ports

- **Input Ports**
Creating cluster using MDG’s
Creating cluster using MDG’s

Gauss Distributed Assigner

Assigns a value based on the class column, this value is gauss distributed as defined in the configuration by its mean and max.

Options:
- Dependency Column
  - clusterID
- Column Name: Dim1
- Seed: 1279533108338
- Rounds: min=-5,000, max=5,000

<table>
<thead>
<tr>
<th>Name</th>
<th>Mean</th>
<th>Var</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster4</td>
<td>0</td>
<td>150</td>
</tr>
<tr>
<td>Cluster3</td>
<td>-50</td>
<td>100</td>
</tr>
<tr>
<td>Cluster1</td>
<td>500</td>
<td>200</td>
</tr>
<tr>
<td>Cluster2</td>
<td>1000</td>
<td>50</td>
</tr>
<tr>
<td>Cluster3</td>
<td>-800</td>
<td>200</td>
</tr>
</tbody>
</table>
Creating cluster using MDG’s
Creating cluster using MDG’s

Gauss Distributed Assigner

assigns a value based on the class column, this value is gauss distributed as defined in the configuration by its mean and max.

Dialog Options

Dependency Column
choose a class column

Category Name
the name of the category

seed
the random seed

Bounds
these bounds are used for the whole column.

values
for each value of the given class column, you can define the mean and the variance

Ports
Creating cluster using MDG’s
Creating cluster using MDG’s
Creating cluster using MDG’s
Creating cluster using MDG’s

k-Means

This node outputs the cluster centers for a predefined number of clusters (no dynamic number of clusters). k-means performs a crisp clustering that assigns a data vector to exactly one cluster. The algorithm terminates when the cluster assignments do not change anymore. The node can be configured as follows:

Dialog Options

number of clusters

The number of clusters (cluster centers) to be created.

max number of iterations

The number of iterations after which the algorithm terminates, independent of the accuracy improvement of the cluster centers.

Ports

Input Ports

0 Input to clustering. All numerical values and only these are considered for clustering.
Used Modules

- Creates starting table  
  (No columns, only rows with identifiers)

- Assigns random labels.  
  (based on specified labels / probabilities)

- Assigns random numbers.  
  (based on Gaussian distribution)

- Adds noise to a column  
  (numerical or categorical outliers)
Other Modules

- Assigns conditional random labels (based on values of another column)
- Inserts various kind of rules
Other Modules for Random Numbers

- Independent: one distribution for entire column
- Dependent: distribution for each value of a second column
Association Rules

Increase support of the rule (items)

Verify: Mine Rules!

Generate Baskets

GroupBy

Random Item Inserter

One Rule Inserter

Association Rule Learner

Mapping basketID – productID

Baskets with set of products

Increase confidence of the rule (A=>B)

Verify: Mine Rules!
Combining tables randomly

• E.g. combine the clusters and the association rules
Combining tables randomly

- E.g. combine the clusters and the association rules
Combining tables randomly

• E.g. combine the clusters and the association rules

![Diagram showing model generation and random matcher]

<table>
<thead>
<tr>
<th>Row ID</th>
<th>Date</th>
<th>haircolor</th>
<th>height</th>
<th>shoesize</th>
<th>agency</th>
<th>nrofjobs</th>
<th>BasketID</th>
<th>product</th>
</tr>
</thead>
<tbody>
<tr>
<td>c_1916</td>
<td>07.May.1965</td>
<td>red</td>
<td>133.64</td>
<td>35</td>
<td>redishC</td>
<td>14</td>
<td>c_73BID_5</td>
<td>turkey</td>
</tr>
<tr>
<td>c_1926</td>
<td>01.Aug.1981</td>
<td>219.936</td>
<td>41</td>
<td></td>
<td>darkC</td>
<td>14</td>
<td>c_868BID_6</td>
<td>Coke</td>
</tr>
<tr>
<td>c_1932</td>
<td>23.Apr.1972</td>
<td>blond</td>
<td>185.083</td>
<td>41</td>
<td>darkC</td>
<td>18</td>
<td>c_613BID_21</td>
<td>Sharon Fruit (Persimmon)</td>
</tr>
<tr>
<td>c_1942</td>
<td>09.May.1956</td>
<td>brunette</td>
<td>155.215</td>
<td>38</td>
<td>darkC</td>
<td>13</td>
<td>c_673BID_10</td>
<td>Persimmon</td>
</tr>
<tr>
<td>c_1950</td>
<td>15.Feb.1974</td>
<td>brunette</td>
<td>34.312</td>
<td>37</td>
<td>darkC</td>
<td>12</td>
<td>c_649BID_2</td>
<td>chips</td>
</tr>
<tr>
<td>c_1979</td>
<td>03.Mar.1960</td>
<td>brunette</td>
<td>104.911</td>
<td>36</td>
<td>darkC</td>
<td>17</td>
<td>c_692BID_2</td>
<td>Apple Juice</td>
</tr>
<tr>
<td>c_1994</td>
<td>24.Mar.1951</td>
<td>blond</td>
<td>246.774</td>
<td>46</td>
<td>redishC</td>
<td>21</td>
<td>c_507BID_5</td>
<td>milk lactosefree</td>
</tr>
<tr>
<td>c_2003</td>
<td>03.Aug.1966</td>
<td>red</td>
<td>114.957</td>
<td>38</td>
<td>redishC</td>
<td>21</td>
<td>c_59BID_1</td>
<td>vodka</td>
</tr>
<tr>
<td>c_2022</td>
<td>21.Apr.1975</td>
<td>brunette</td>
<td>100.423</td>
<td>28</td>
<td>redhairC</td>
<td>25</td>
<td>c_331BID_5</td>
<td>Coke</td>
</tr>
</tbody>
</table>
Generate Customer Data

- Generate one row per bought item

Assign a product category randomly

Assign a random real product with matching price and product category

Assign a price category based on gender and occupation
Other things you may want to know but I did not have time to talk about today...

- All generators use (user specifiable) random seeds:
  - Deterministic results
  - Allows to generate different data sets following the same pattern.

- KNIME Workflow variables allow to control global settings

- Batch Execution for automatic data generation
Summary / Outlook

Modular Data Generation in KNIME:
• Visual documentation of complex data generation processes
• Repeatability
• Extendibility

Ongoing Work:
• Add functionality to generate time-dependent data

Available now (nodes and workflows):
http://www.knime.org/datageneration

(demo tonight at the poster)