



THE
POWER
TO KNOW.

**SAS[®] Enterprise Miner[™] and
SAS[®] Text Miner Procedures
Reference for SAS[®] 9.1.3
The RULEGEN Procedure
(Book Excerpt)**

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., SAS® *Enterprise Miner™* and SAS® *Text Miner Procedures: Reference for SAS® 9.1.3*, Cary, NC: SAS Institute Inc.

SAS® Enterprise Miner™ and SAS® Text Miner Procedures: Reference for SAS 9.1.3

Copyright © 2008 by SAS Institute Inc., Cary, NC, USA.

All rights reserved. Produced in the United States of America.

For a Web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

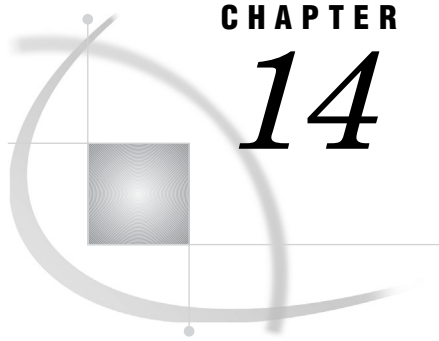
SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st electronic book, October 2008

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at support.sas.com/publishing or call 1-800-727-3228.

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.



CHAPTER

14

The RULEGEN Procedure

Overview: <i>RULEGEN Procedure</i>	323
Syntax: <i>RULEGEN Procedure</i>	323
<i>PROC RULEGEN Statement</i>	323
Details: <i>RULEGEN Procedure</i>	325
<i>Output Processing</i>	325
Example: <i>RULEGEN Procedure</i>	326
<i>Example 1: Performing an Association Discovery</i>	326

Overview: RULEGEN Procedure

PROC RULEGEN uses the output data set created by PROC ASSOC and generates association rules and computes statistics, such as confidence and lift, for the rules. PROC ASSOC identifies item sets that are related. The RULEGEN procedure generates the rules governing their association. PROC RULEGEN output is saved as a SAS data set that can be viewed or browsed by SAS procedures that you can create to reflect your own evaluation criteria.

Syntax: RULEGEN Procedure

```
PROC RULEGEN <option(s)>;
```

PROC RULEGEN Statement

Invokes the RULEGEN procedure.

```
PROC RULEGEN <option(s)>;
```

Required Argument

OUT=<libref.>SAS-data-set

Specifies the output data set to which the rules are written. The output data set has the following variables: CONF, COUNT, EXP_CONF, ITEM1, ITEM2, ..., ITEM_{n+1}, _LHAND, LIFT, _RHAND, RULE, SET_SIZE, SUPPORT.

CONF=COUNT/lhs_count

Contains the percent of confidence.

Definition: *lhs_count* is the number of transactions satisfying the left side of the rule.

COUNT

Contains the number of transactions meeting the rule.

EXP_CONF=rhs_count/total

Contains the percent of expected confidence.

Definition: *rhs_count* is the number of transactions satisfying the right side of the rule.

ITEM1, ITEM2, ..., ITEM_{n+1}

Contains individual items which make up the rule, including the arrow.

_LHAND

Identifies the left side of the rule, where the rule is expressed: _LHAND ==> _RHAND.

LIFT= CONF/EXP_CONF

Contains the lift ratio.

_RHAND

Identifies the right side of the rule, where the rule is expressed: _LHAND ==> _RHAND.

RULE

Contains the text of the rule, for example, A & B ==> C & D.

SET_SIZE

Contains the number of items in the rule.

SUPPORT= COUNT/total

Contains the percent of support, that is, the percent of the total number of transactions that qualify for the rule.

Definition: *total* is the total number of transactions in the data set.

Options

IN=<libref.>SAS-data-set

Specifies the input data source. The input to PROC RULEGEN is the OUT= data set created in PROC ASSOC.

Default: _LAST_

MINCONF=integer

Specifies the minimum confidence level needed in order to generate a rule. This parameter can be adjusted so that only high confidence rules are retained.

Default: 10%

PMML

Specifies that PMML score code be generated. PMML is an XML-based standard for representing data mining results. For more information, see the PMML Support in Enterprise Miner section in the Enterprise Miner 5.3 Java Help.

Details: RULEGEN Procedure

Output Processing

The output data set created by PROC RULEGEN has the following variables:

SET_SIZE

Contains the number of items in the rule.

RULE

Contains the rule text, for example, A & B ==> C & D.

COUNT

Contains the count of transactions meeting the rule.

CONF

Contains the percent of confidence.

EXP_CONF

Contains the percent of expected confidence.

LIFT

Contains the lift ratio.

SUPPORT

Contains the percent of support.

_LHAND

Contains the left side of the rule.

_RHAND

Contains the right side of the rule.

ITEM1, ITEM2, ..., ITEM_n+1

Contains the individual items forming the rule, including the arrow.

Only the rules meeting the minimum confidence value are output. This parameter can be adjusted to retain only the high confidence rules.

The statistical computation is based on Bayes' theorem, stated as probability of event A conditional on event B occurring, and is calculated as the probability of both events A and B occurring divided by the probability of event B.

PROC RULEGEN automatically discovers complex rules with multiple events on either side such as A & B ==> C, implying event C occurred, given that both events A and B occurred.

Consider the rule lhs ==> rhs.

In terms of the output data set variables, the statistics are computed as follows:

- CONF= COUNT/lhs_count
- EXP_CONF=rhs_count/total
- LIFT=CONF/EXP_CONF
- SUPPORT=COUNT/total

where *total* is the number of transactions in the data set.

As you can see, positioning of items on the left or right side does impact statistical calculations, that is, A ==> B and B ==> A are entirely different rules.

Example: RULEGEN Procedure

The following example was executed using the Windows XP Professional operating system and the SAS software release 9.1.3.

Example 1: Performing an Association Discovery

Featured Tasks ASSOCIATION and RULEGEN Procedures

- Specifying the maximum item-set size
- Setting the support level
- Specifying the minimum confidence level

The following example demonstrates how to perform an association discovery using the ASSOCIATION and RULEGEN procedures. The example data set SAMPSIO ASSOCS (stored in the sample library) contains 7,007 separate customer transactions. The variable CUSTOMER is the ID variable that identifies the customers. The variable PRODUCT is the nominal target variable that identifies the items. As a marketing analyst for a grocery chain, you want to identify the top 10 item sets, where the purchase of one item has a high impact on the purchase of another item(s).

Program

Before you can run PROC ASSOC, you must create the the DMDB catalog by using a PROC DMBD step.

```
proc dmdb batch data=sampsio.assocs dmdbcat=catassoc;
  id customer;
  class product(desc);
run;
```

The ASSOCIATION procedure determines the products that are related. The DATA= and DMDB= options identify the input data set and the DMDB catalog, respectively. PROC ASSOC writes the related products to the OUT= data set, which is used as input by the RULEGEN procedure.

```
proc assoc data=sampsio.assocs dmdbcat=catassoc
  out=datassoc(label='Output from Proc Assoc')
```

The ITEMS= option specifies the maximum size of the item set to be considered (default=4). The SUPPORT= option specifies the minimum support level that is required for a rule to be accepted (default =5% of the largest frequency).

```
  items=5 support=20;
```

The CUST statement (alias = CUSTOMER) specifies the ID variable. The TARGET statement specifies the nominal target variable.

```

    cust customer;
    target product;
run;
```

The RULEGEN procedure uses the output from PROC ASSOC to generate the rules. The rules are written to the OUT=data set.

```

proc rulegen in=datassoc
    out=datrule(label='Output from Proc Rulegen')
```

The MINCONF= option specifies the minimum confidence required in order to generate a rule (default =10).

```

    minconf=75;
run;
```

Because neither PROC ASSOC nor RULEGEN generates printed output, the remaining code sorts the data by the LIFT values and then generates a simple list report of the rules that have the top 10 values for LIFT. This is done primarily to limit the amount of output displayed in this example.

```

proc sort data=datrule;
    by descending lift;
run;
proc print data=datrule(obs=5) label;
    var set_size exp_conf conf support lift count
        rule _lhand _rhand;
    title 'Top Ten Rules based on Lift';run;
```

Output

The PROC PRINT list report of the top 10 rules based on the LIFT value. The output data set from PROC RULEGEN contains the following variables:

- SET_SIZE – contains the number of items in the rule.
- EXP_CONF – the expected confidence (right side count/total).
- CONF – the confidence (count / left side).
- SUPPORT – the support level (count/total).
- LIFT – the lift ratio (confidence/expected confidence).
- COUNT – number of transactions meeting the rule.
- RULE – contains the text rule, for example, Right side ==> Left side.
- _LHAND – contains the left side of the rule.
- _RHAND – contains the right side of the rule.
- ITEM1, ITEM2, ..., ITEM_n+1 – contains the individual items forming the rule, including the arrow. For this example, the individual items have been omitted from the list report.

Top Ten Rules based on Lift

Obs	Relations	Expected Confidence (%)	Confidence (%)	Support (%)	Lift	Transaction Count
1	5	12.59	94.74	8.99	7.53	90.00
2	5	10.79	78.99	9.39	7.32	94.00
3	5	11.89	87.04	9.39	7.32	94.00
4	5	12.69	92.78	8.99	7.31	90.00
5	5	13.19	93.75	8.99	7.11	90.00

Obs	Rule
1	sardines & baguette & apples ==> peppers & avocado
2	turkey & coke ==> olives & ice_crea & bourbon
3	olives & ice_crea & bourbon ==> turkey & coke
4	peppers & baguette & apples ==> sardines & avocado
5	peppers & avocado & apples ==> sardines & baguette

Obs	Left Hand of Rule	Right Hand of Rule
1	sardines & baguette & apples	peppers & avocado
2	turkey & coke	olives & ice_crea & bourbon
3	olives & ice_crea & bourbon	turkey & coke
4	peppers & baguette & apples	sardines & avocado
5	peppers & avocado & apples	sardines & baguette

Log

```

89  proc dmdb batch data=sampsio.assocs dmdbcat=catassoc;
90  id customer;
91  class product(desc);
92  run;

```

NOTE: Records processed = 7007 Memory used = 511K.

NOTE: There were 7007 observations read from the data set SAMPSIO.ASSOCS.

NOTE: The data set WORK.DMASSOC has 7007 observations and 2 variables.

NOTE: PROCEDURE DMDB used (Total process time):

```

real time          0.03 seconds
cpu time           0.03 seconds

```

```

93  proc assoc data=sampsio.assocs dmdbcat=catassoc
94  out=datassoc(label='Output from Proc Assoc')
95  items=5 support=20;
96  cust customer;
97  target product;
98  run;

```

```

----- Potential 1 item sets = 20 -----
Counting items, records read:      7007
Number of customers:                1001
Support level for item sets:        20
Maximum count for a set:            600
Sets meeting support level:         20
Megs of memory used:                0.51

```

```

----- Potential 2 item sets = 190 -----
Counting items, records read:      7007
Maximum count for a set:            366
Sets meeting support level:         183
Megs of memory used:                0.51

```

```

----- Potential 3 item sets = 1035 -----
Counting items, records read:      7007
Maximum count for a set:            234
Sets meeting support level:         615
Megs of memory used:                0.51

```

```

----- Potential 4 item sets = 1071 -----
Counting items, records read:      7007
Maximum count for a set:            137
Sets meeting support level:         317
Megs of memory used:                0.51

```

```

----- Potential 5 item sets = 85 -----
Counting items, records read:      7007
Maximum count for a set:            116
Sets meeting support level:         71
Megs of memory used:                0.51

```

NOTE: The data set WORK.DATASSOC has 1207 observations and 7 variables.

NOTE: PROCEDURE ASSOC used (Total process time):

```

    real time          0.01 seconds
    cpu time           0.01 seconds

```

```

103 proc rulegen in=datassoc
104 out=datrule(label='Output from Proc Rulegen')
105 minconf=75;
106 run;

```

```

write set 1
write set 2
write set 3
write set 4
write set 5

```

NOTE: There were 1207 observations read from the data set WORK.DATASSOC.

NOTE: The data set WORK.DATRULE has 939 observations and 15 variables.

NOTE: PROCEDURE RULEGEN used (Total process time):

```

    real time          0.03 seconds
    cpu time           0.01 seconds

```

```
107 proc sort data=datrule;
108 by descending lift;
109 run;
```

NOTE: There were 939 observations read from the data set WORK.DATRULE.

NOTE: The data set WORK.DATRULE has 939 observations and 15 variables.

NOTE: PROCEDURE SORT used (Total process time):

real time	0.01 seconds
cpu time	0.01 seconds

```
110 proc print data=datrule(obs=5) label;
111 var set_size exp_conf conf support lift count
112 rule_lhand_rhand;
113 title 'Top Ten Rules based on Lift';run
```