Rule-Based Rating Model

Anatoliy Antonov, Vladimir Nikolov, Yanka Yanakieva

Abstract: *Rating Models are intended to evaluate the rating of counterparts within various rating systems. The paper considers a rule-based approach for flexible rating model development based on a set of objective data and subjective criteria grouped in classes. Rating Criteria are aggregated using expressions and rules to obtain the counterpart scoring. The model script (Using CLIPS).*

Keywords: *Artificial intelligence, Fuzzy systems, Expert system, Rating Systems, Credit Risk*

INTRODUCTION

The evaluation of the rating of counterparts within one or many internal or external rating systems used to estimate the credit standing of counterparts against their ability to serve loans and other debt transactions is one of the most important tasks of credit managers. The main problems in determining the counterpart rating using a rating procedure or a rating application consist of providing flexibility for defining rating criteria and aggregation rules. The desired model flexibility is expressed by the following requirements:

- The set of data and criteria having influence on counterpart credit standing is not fixed and depends on the counterpart type – the rating system should be able to derive the rating from different sets of counterpart data using different sets of rules.
- The values for the rating criteria can’t be set in most cases exactly because of missing or inexact information - the rating model should be able to work with non-exact variables exposing confidence distributions within some range.
- The rating criteria are aggregated by a set of expressions and rules that don’t have full confidence in many cases – the rating model should be able to apply rules with certain confidence on non-exact variables to obtain the counterpart scoring.
- The rating model should apply the rating move probabilities from transition matrix of various rating systems while mapping the scoring result to rating levels.

The requirements above lead to flexible rule-based model scripts allowing simple development and configuring of a software rating tool. The model development includes the following steps:

1. Selecting a set of appropriate basic exact or non-exact criteria that describe a counterpart related to his ability to cover his outstanding obligations.
2. Defining a set of higher-level criteria including top-level results (scoring, rating level). The criteria and results form a variable (fact) space.
3. Creating a rule set that order criteria and results within a dependence tree allowing for a reasoning process. Input criteria, results and rules are written down in CLIPS and stored as rating model in a script file.
4. Saving the developed model in a corresponding script file.

1. MODEL SCRIPT

The Rating model script contains all definitions of templates, facts and rules needed to

produce counterpart rating according to the implemented model. The model script is independent from counterpart data and it can be common for a large set of counterparts that have similar behavior and that can be described by the same set of criteria and rules. The model scripts are hold in text files with the extension .clp.
The Rating models exposes a standardized section structure:

```
Fuzzify Function

TEMPLATES for Structure

TEMPLATES for simple Variables

TEMPLATES for fuzzy Variables

FACT Hierarchy

RULE Section
```

The Fuzzify Function is written in CLIPS and is used to convert simple variables to fuzzy variables. This function exists by default and the model developer doesn't change it.
The TEMPLATES for Structure section contains templates for fact structuring providing templates for classes, criteria, questions and results. This section exists by default and the model developer doesn't change it.
The sections containing TEMPLATES for simple Variables and TEMPLATES for fuzzy Variables provide templates for variables referenced by facts from FACT hierarchy section. Ranges and default values for simple variables and fuzzy distributions are defined within this section. The variable type, the variable range and the fuzzy distribution represent properties of the referencing facts and they depend on fact substance and meaning. The model developer should define these properties according to the selected rating variables.
The FACT Hierarchy section contains the basic fact set used to describe input data and results of rating process. The following fixed fact hierarchy is used:

```
creditroot
     creditclass 1
          creditcriteria 1
               creditquestion 1
               creditquestion 2
               ----------------
               creditresult 1
               creditresult 2
               --------------
          creditcriteria 2
          ----------------
     creditclass 2
     -------------
```

The root includes classes, the classes include criteria, the criteria include questions and results. Credit root, credit classes and credit criteria don't contain any variable values. They are used only to structure the GUI and the user interaction. Input and result variables are present in credit questions and in credit results. Rules can apply on any credit question or on credit result variables regardless of fact hierarchy.

The RULE Section includes all rules describing dependencies between facts (questions and results). These dependencies are expressed by the general form:

```
IF      facts variables (questions or results) are present in the fact base
THEN
        calculate value of new or same facts (results) and
        insert these facts into fact base
```

The inserted facts can be included in the IF part of other rules causing firing of these rules that insert new facts and so on until high level results are obtained.

2. FACT DEFINITION

Questions, criteria, classes, ranges and fuzzy distributions are described in the Rating model using templates.

The Rating facts (classes, criteria and questions) build a hierarchical structure. The facts are defined by the statement deffacts and initfacts and loaded into the fact base after parsing the rating model script. The facts refer to the fact templates and set values (strings) for the template slots. These strings remain constant while user interacts within the Risk Rating GUI and during the inference process. The inference engine works only with the referenced variables for questions and results linked by the slots named FuzzyID that connect facts to simple or fuzzy variable templates.

Examples for facts related to root, classes, criteria, questions and results are given below. The name of rating model is hold in root fact slot "name".

3. RULE DEFINITION

The definition of rules is the most important step during model development. Rules can express dependencies between facts within hierarchic fact derivation tree or trees. Rules are activated (fired) upon presence of a set of facts in the fact base. The reasoning process is data driven, result facts are inserted into the fact base by rules and other rules are activated by inserted facts. Value for result facts can be calculated using a large set of logic or arithmetic functions of CLIPS engine. The next sections show example rules that can be used within rating process. A part of these rules works with simple variables containing balance sheet data that are combined by arithmetic expressions to produce indicator values (Kennzahlen). Other rules use indicators to create scores for desired counterpart criteria set. High-level rules produce final scoring results and try to map the scoring results to ratings within a selected rating order.

3.1 Rules for Indicators ("Kennzahlen")

```
     (defrule r-B1
          ?A.1.8  <-   (A.1.8   (value ?A_1_8v  )  )
          ?A.1.10 <-   (A.1.10  (value ?A_1_10v )  )
     =>
          (assert (B.1.RESULT (value (*(/?A_1_10v
?A_1_8v)100.0))))
     )
```

The example rule above fires if two simple variables (A.1.8 and A.1.10) are in the fact base and have valid values. The rule calculates the value for the result fact (B.1.RESULT) and inserts it into the fact base. The rule calculates in fact an indicator value:

```
     Betriebsrendite(in %) = (EBIT / Umsatzerlöse) *  100
```

The arithmetic expression is written using two argument arithmetic functions represented in the form:

(operator first_argument second_argument)

where every argument can be a function or constant.

3.2 Rules for Scores

```
     (defrule r-TR1
          ?B.1.RESULT <-   (B.1.RESULT  (value ?B.1.RESULTv )  )
     =>


          (fuzzify TR.1.RESULT  (* 1.2 ?B.1.RESULTv ) 0.1)
     )
```

The next example rule fires if the simple variable (B.1.RESULT) is in the fact base and has valid value. The rule calculates the value for the fuzzy result variable (TR.1.RESULT) and inserts it into the fact base. The rule transforms an indicator value given in % to a fuzzy score value using the fuzzify function. A simple example transformation is used, i.e.

```
     Betriebsrendite(score between 0 and 10)  =
                    1.2 * Betriebsrendite(in %)
     For example:   7.2 = 1.2 * 6
```

More sophisticated or non-linear transformations can be constructed using, for example, comparison of indicators for counterpart against consolidated indicators for counterpart branch or branches. The implementation of such transformations is quite possible using CLIPS functions in the second rule part (RHS).

3.3 Rule for "Discriminance Function"

The most important rule is the rule that aggregates scoring results for criteria groups into final counterpart score. The rule below applies an example of polynomial function with weights to combine fuzzy scoring results (TR.x.RESULT) into final score (GR.1.RESULT). Local variables such as ?res1 and ?res2 are used to hold intermediate values. The obtained intermediate result in ?res1 is then combined with three other fuzzy variables (B.3.x) to produce the final score. All fuzzy variables are defuzzyfized by MCG before using in expressions. The variable B.3.2 is compared to intermediate result ?res and capes it. This variable represents a killer-criteria, for example it can express the score of country or region of the counterpart that means that the counterpart rating can’t be better than the country or region rating.

The quality of the aggregation rule related to its ability to be an appropriate discrimination function should be checked within the model development process.

4. UNCERTAINTY AND FUZZINESS

Fuzzy CLIPS is an extended version of the CLIPS rule-based shell for representing and manipulating fuzzy facts and rules. In addition to the CLIPS functionality, Fuzzy CLIPS can deal with exact, fuzzy (or inexact), and combined reasoning, allowing fuzzy and normal terms to be freely mixed in the rules and facts of an expert system.

The modifications made to CLIPS contain the capability of handling fuzzy concepts and reasoning. It enables domain experts to express rules using their own fuzzy terms. Fuzzy CLIPS allow any mix of fuzzy and normal terms, numeric-comparison logic controls, and uncertainties in the rules and facts. Fuzzy sets and relations deal with fuzziness in approximate reasoning, while certainty factors for rules and facts manipulate the uncertainty. A Fuzzy CLIPS fact takes the following form:

```
(fact) [CF value]
```

The degree of uncertainty is usually represented by a crisp numerical value on a scale from 0 to 1. A rating model based on the Fuzzy CLIPS features allows work with qualitative estimations and value distributions of the model variables.

5. SIMPLE RATING MODEL EXAMPLE

The rating example called “CRESTA Rating” was inspired by a publication of DG Bank on Internet. Refer to www.cresta-rating.de for details. The example rating model is implemented in the script file Cresta_Rating.clp supplied with the Risk Rating demonstration version.

Rating Criteria Tree

The set of rating criteria, indicators and partial scoring results is constructed within a 5-level hierarchic fact tree (12 balance sheet data, 7 indicators, 7 indicator scores, the final score and the counterpart rating). Example input data and results are given in Table 1:

Table 1

	Balance Sheet Parameter	**Input Data (TDM)**	
1.	Sachanlagen	4.522,10	
2.	Finanzanlagen	161,30	
3.	Immaterielle Anlagewerte	2.954,70	
4.	Bilanzsumme	12.828,00	
5.	Eigenkapital	2.973,60	
6.	Langfristige Finanzschulden	3.805,10	
7.	Kurzfristige Finanzschulden	584,50	
8.	Umsatzerlöse	13.188,60	
9.	Cash-Flow	1.171,40	
10.	Betriebsergebnis (EBIT)	796,90	
11.	Bruttozinsaufwand	228,00	
12.	Jahresergebnis	413,90	
	Indicators	**Calculated Data**	**Partial Scores**
1.	Betriebsrendite	6.04234%	7.25081
2.	Eigenkapitalrendite	13.9192%	8.35149
3.	Umsatzrendite	3.13832%	4.70747
4.	Eigenkapitalquote	23.1805%	6.95416
5.	Deckungsgrad II	88.7497%	4.43748
6.	EBIT/Zinsen	3.49518	6.99035
7.	Cash-Flow zu Finanzschulden	26.68580	5.33716
		Rating	**Final Score**
	Weighting and Aggregation	Moody18(Baa2)	6.58021

Balance Sheet Data

The balance sheet data are grouped in an Input Data class. This data are entered by the user or fetched from PMS database. The balance sheet data are treated as known and represented by simple ranged variables. The corresponding facts include question, why and explanation texts to support the counterpart manager while rating process.

Indicator Calculation

A set of 7 main indicators is calculated from balance sheet data using arithmetic expressions. The indicators are obtained from known data and treated with confidence = 1 (100%). They are represented by simple ranged variables. Refer to www.cresta-rating.de for details about calculation formulas for indicators.

Partial Scoring Results

The indicator data are converted to partial scoring results within range from 0 to 10.
Simple converting by scaling to assumed maximum values is performed.

Final Scoring

The final scoring result is obtained using weighted polynomial function implemented within a rule. The confidence factor of the rule is set to 0.92 (92%) expressing absence of total believe in the discriminance property of the function.

Mapping to Rating

The final scoring result is mapped by a separate rule to external or internal rating system within the counterpart should be rated. The mapping is implemented as follows:

- A master scale containing default probabilities for scoring levels is used to obtain the probability of default for given score.
- The transition matrix of counterpart is loaded, and the probability of default is searched in the last matrix column.
- The corresponding row number or rating level name is set as the new counterpart rating.

This mapping procedure goes through probabilities of default and allows mapping of score results to different transition matrixes. The probabilities of default of the master scale can be obtained using objective historic default statistics.

CONCLUSIONS AND FUTURE WORK

The practical application of many developed rule-based rating models in the Risk Rating module integrated in the Risk Evaluator has led to the following conclusions:

- The rule-based rating models build the kernel of a software rating system
- The rule-based rating model ensures a flexibility of the software system allowing the user to change the variables, to adjust or to change the rating criteria and rules.

REFERENCES

[1] Kasabov K. Nikola, Foundations of neuraral networks, fuzzy systems and knowledge engeneering, 1998, MIT Press, USA.

[2] RiskMetrics - Technical Document, Third Edition, 1995, New York

[3] www.cresta-rating.de

ABOUT THE AUTORS

Dr. Anatoliy Antonov
Eurorisk Systems Ltd.
31, General Kiselov Str., 9002 Varna, Bulgaria
E-mail: antonov at euroriskystems dot com
Dr. Vladimir Nikolov
Eurorisk Systems Ltd.
31, General Kiselov Str., 9002 Varna, Bulgaria
E-mail: nikolov at euroriskystems dot com
YankaYanakieva
Eurorisk Systems Ltd.
31, General Kiselov Str., 9002 Varna, Bulgaria
E-mail: y_yanakieva at euroriskystems dot com