A Metric to Evaluate Robustness of a Rule-Based System

Sunil Kumar Kopparapu
TCS Innovation Labs - Mumbai,
Tata Consultancy Services Limited
Yantra Park, Thane(West), Maharastra, Mumbai 400601, INDIA.
Email: SunilKumar.Kopparapu@TCS.Com

***Abstract*—AI based expert systems are often rule based and are used in several commercial applications. Rules are often arrived at based on experts knowledge. The process of deriving rules is often subjective, error prone and inexact. For a given set of true inputs, the performance of a rule based system is usually measured by the correctness of the rules, derived from the experts knowledge, to produce the desired output. In almost all the rule based systems the inputs to the expert system themselves could be erroneous having been generated from another process. In this paper we assume that the the rule base created from the expert knowledge is not erroneous. We formulate a metric to measure the robustness of a rule based system in terms of erroneous inputs to the expert system. We first formulate the metric for measuring the robustness of a rule based system and then show through an example its utility to measure the robustness.**

I. Introduction

Typically expert systems are often rule based and work on assimilation of the expert knowledge in the form of *if-then-else* kind of rules. Rule-based systems are used extensively in applications such as mortgages, credit card authorization, fraud detection, e-commerce, and personalization [1]. The process of deriving rules from the experts knowledge is subjective, error prone and inexact giving raise to different rule set for the same experts knowledge leading to use of fuzzy rules in expert systems [2]. There are essentially two methods of rule-based system [3] namely forward-chaining and backward-chaining. In forward-chaining the input information is provided and the system traverses through the rule-base to identify the answer or the output, while in backward-chaining the goal or the answer is provided and the system needs to traverse the rule-base to identify the inputs to achieve the goal. Which method to use is usually determined by the nature of problem at hand.

We can visualize an expert system to be an *input-output* system, where a set of inputs leads to a specific set of outputs which are derived by traversing through the rule-base. This process is deterministic in the sense that the same set of inputs lead to the same set of outputs as long as the rule-base is unchanged. However, in practise, the set of inputs to the system are often provided by yet another process which could be erroneous. In this scenario the output produced by the system using the same rule base could result in a different outcome which is consistent with the erroneous inputs.

The underlying assumption in most rule based systems is the correctness of the inputs to the expert system to produce the desired outcome. When the inputs are not erroneous, the rule based system performs as per the rule base supporting the expert system. Most often the inputs themselves are far from correct, especially when the inputs are obtained from another process, this introduces errors in the input. The effect of this is an outcome which might or might not be outcome for non-erroneous inputs. The robustness of an expert system, or the rule-base supporting the expert system can be measured by the ability of the rule-base to work with erroneous inputs.

In literature there are some studies on evaluation of rulebased system albeit in different contexts. For example, Chander et. al. [4] talk about a method to evaluate rule-based system when rules and incrementally added to the system; In [5] Alun et. al. talk of validation the rule-based systems when the data is incomplete. In this paper, we formulate a metric to measure the robustness of a rule-based system; this is the primary contribution of the paper. We further use this metric to show how we can measure the robustness of a rule-based system. The rest of the paper is organized as follows: In Section II we introduce the metric to measure the robustness of a rule-based system, in Section III we show experimentally how this metric can be used to measure the robustness of a a rule-based system and conclude in Section IV.

II. Metric for Evaluating a Rule-Base

In a very simplistic view, a rule-based system uses a set of *if-then-else* kind of rules, mapped from human expert knowledge, to map a set of inputs to the desired output. For the purpose of analysis let a rule-base $\mathcal{R}$ be defined as a function which takes an input $\mathcal{I}$ of dimension R, say $[i^1, i^2, \cdots i^R]$, and determines $\mathcal{O}^S$, namely,

$$\mathcal{R} : \mathcal{I}^R \rightarrow \mathcal{O}^S$$

The output O^S is a vector of dimension S (all possible outcomes) and is represented as $[o^1, o^2, \cdots o^S]$. Depending on the expert system the output could be either

- binary output, namely, $o^k = 1$ and $\forall l \neq k$, $o^l = 0$ a binary vector where only one outcome is signaled or
- scalar valued output, namely, there is a scalar value associated with all the S outcomes with a constraint that $\sum_{i=1}^{S} o^i = 1$, the output is often the o^k with the largest value.

In our discussion we will assume that the expert system produces an output O^S where each o^i has a scalar value.

Note 1: Note that the binary output case is a special case of the scalar values output.

Typically, the dimension of output, S, is small compared to the input dimension R. We formalize a mechanism to measure the performance of a rule-base[1], $\mathcal{R}$.

Definition 1: A rule-base $\mathcal{R}$ is said to be robust when an error ϵ in the input, namely an error in $\mathcal{I}^R$, results in no significant change in the outcome $\mathcal{O}^S$.

Note 2: As discussed in the previous section, in practical systems the inputs to the expert system invariably come from another process making errors in input highly probable.

Let us assume that there is an error ϵ in the input $\mathcal{I}^R$. which results in an erroneous input $\mathcal{I}_\epsilon^R$. The rule-base $\mathcal{R}$ acts on the erroneous input to produce the outcome O_ϵ^S namely,

$$\mathcal{R} : I_\epsilon^R \rightarrow O_\epsilon^S$$

The resulting error in the outcome due to the ϵ error in the input is given by D_ϵ, the typical Euclidean measure, namely,

$$D_\epsilon \triangleq d(O_\epsilon^S, O^S) = \sqrt{\frac{1}{S}\sum_{i=1}^{S}(o_\epsilon^i - o^i)^2} \qquad (1)$$

[1]We will use rule-base and expert system interchangeably in this paper

Clearly, $D_\epsilon \in [0, 1]$, the smaller the value of D_ϵ the better the robustness of $\mathcal{R}$ to the error ϵ in the input $\mathcal{I}^R$. Further there is a need that not only D_ϵ be small but also the final outcome because of the error in the input not change, meaning if the selection criteria is to pick the maximum in $\{o^i\}_{i=0}^S$, say o^k, then the erroneous input should also give the same o^k under the selection criteria. Subsequently, there are three possible outcomes due to error in the input, namely,

- C_0 When an error in the input does not cause any change in the final outcome.
- C_1 When an error in the input causes the output to change, namely, if o^k is the output produced without any error then an erroneous input produces an outcome o^l such that $l \neq k$.
- C_ϕ When an error in the input fails to produce any output because there are no rules to handle this erroneous set of inputs. In this case the input is considered unanswerable by the $\mathcal{R}$.

Note 3: Each of these cases of outcomes, namely, C_0, C_1 and C_ϕ has a D_ϵ, defined in (1) associated with it.

Clearly C_0 with small values of D_ϵ is a desirable outcome and an ideal system should have minimal or no C_1 and C_ϕ outcomes at all.

A. Evaluating Rule-Bases

We formalize a metric to measure the performance of two rule-bases, say, $\mathcal{R}_i$ and $\mathcal{R}_j$ for the different cases (C_0, C_1 and C_ϕ). In the case of C_0, an error in the input results in the rule-base $\mathcal{R}_i$ producing an output, say, $\mathcal{O}_{i,\epsilon}^S$.

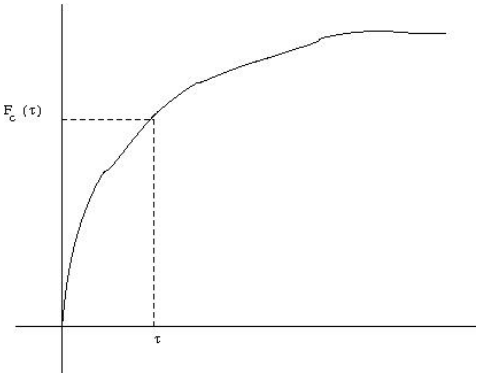


Fig. 1. Cumulative distribution F_C curve for inputs in C_0

Note 4: In the case of C_0 an error in input does not change the outcome in terms of the final decision, however, the output $O_{i,\epsilon}^S$ is different from O_i^S and produces an error D_ϵ.

Note that the input $\mathcal{I}$ is of dimension R and is denoted by $[i^1, i^2, \cdots i^R]$. If the input i^k can take v_k possible different values then one can construct a set $\mathcal{A}$ of inputs with cardinality $\mathcal{N}$ where

$$\mathcal{N} = \prod_{k=1}^{R} v_k$$

$\mathcal{A}$ consist of all the different possible enumerations of the inputs. A subset of $\mathcal{A}$, say $\mathcal{V}$, are valid, in the sense the inputs that the expert system is expected to handle. Let the cardinality of $\mathcal{V}$ be $\mathcal{M}$. For all the $\mathcal{M}$ inputs in $\mathcal{V}$ generate a set of erroneous inputs, let this be denoted by the set $\mathcal{V}_\epsilon$.

Note 5: Note that the erroneous input $\mathcal{I}_\epsilon$ corresponding to an input $\mathcal{I} \in \mathcal{V}$ belongs to $\mathcal{A}$.

Now collect all the possible inputs ($\in \mathcal{V}_\epsilon$) that fall into category C_0 into a a set $\mathcal{C}$, and compute for all $\mathcal{I}^R \in \mathcal{C}$ the error D_ϵ in the output using (1) due to the error (ϵ) in the input. Let us say there are a set of k inputs that fall in the category C_0. Then corresponding to each of the k inputs there is an output produced by the rule-base, this is the desired output[2]. For each of the k outputs calculate the distance D_ϵ as mentioned in (1). This results in k scalar values, say $\{D_\epsilon^i\}_{i=1}^k$. An histogram of the k distances $\{D_\epsilon^i\}_{i=1}^k$ gives the distribution of the error in the output for inputs in the set $\mathcal{C}$. Construct a cumulative density function $F_\mathcal{C}$ as shown in Figure 1. The x-axis is a measure of D_ϵ (error in determining the output) and the y-axis gives the the total number of inputs in the set $\mathcal{C}$ which are within some error distance. So for any point τ on the x-axis, we get a value on the curve $F_\mathcal{C}(\tau)$ which gives the total number of k inputs which are within a distance of τ, namely all erroneous inputs $\mathcal{I}_\epsilon \in \mathcal{C}$ with error D_ϵ such that $D_\epsilon \leq \tau$.

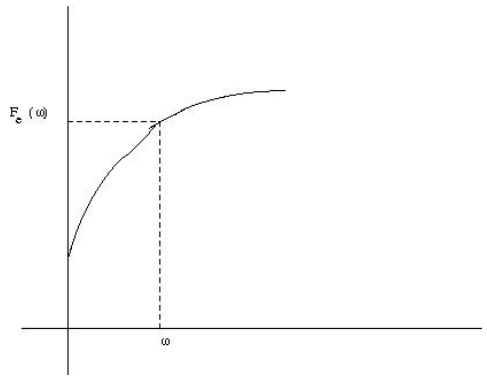


Fig. 2. Cumulative distribution $F_\mathcal{E}$ curve for inputs in C_1

[2]because each of these k inputs are in the category C_0

Similarly collect all the possible inputs ($\in \mathcal{V}_\epsilon$) that fall into category C_1, namely, when an error in input $\mathcal{I}^R$ results in an output such that there is a change in the determined outcome. Here the rule-base is susceptible to error. We collate all the inputs which fall in C_1 into the set $\mathcal{E}$. Let us say there are l inputs in the set $\mathcal{E}$. As explained earlier for the inputs in the set $\mathcal{C}$ we compute the cumulative distribution, $F_\mathcal{E}$. So for any point ω on the x-axis, we get a value on the curve $F_\mathcal{E}(\omega)$ (see Figure 2) which gives the number of inputs $\mathcal{I}in\mathcal{E}$ which are within a distance of ω, namely, $D_\epsilon \leq \omega$ from the erroneous output due to an error in the input.

Note 6: A good rule-base is one which has a higher value of $F_\mathcal{E}(\omega)$ and $F_\mathcal{C}(\tau)$ simultaneously for small values of ω and τ respectively.

Definition 2: A rule-base $\mathcal{R}_i$ is said to be more robust that the rule-base $\mathcal{R}_j$ *if and only if* for a given τ and ω

$$F_\mathcal{E}^i(\omega) > F_\mathcal{E}^j(\omega)$$

and

$$F_\mathcal{C}^i(\tau) > F_\mathcal{C}^j(\tau)$$

are true simultaneously.

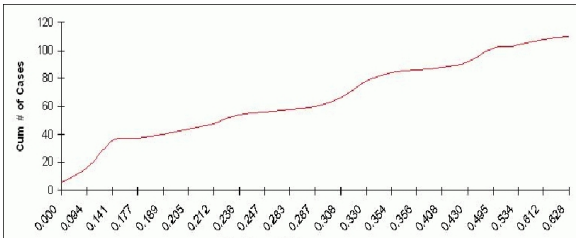
III. Experimental Results

For the purpose of testing the performance of the proposed metric to determine the robustness of a rule-base system we constructed two separate rule-base, one was a hand crafted rule-base ($\mathcal{R}_h$) while the other was a system generated rule-base[3] ($\mathcal{R}_s$) for the same problem.

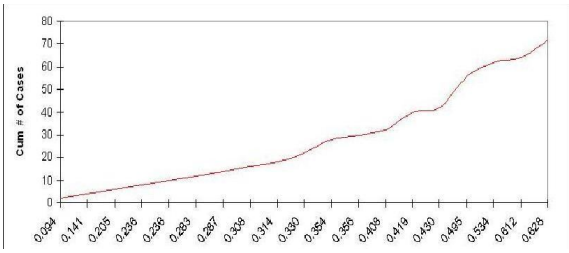
We constructed an interface to the rule-based system which would take a natural English input query as the input and would determine the intent of the query by positioning the query as being belonging to one of the $S = 4$ classes, namely, a yellow pages query *(Where is McDonalds in Manhattan?)*, a movie search query *(Where can I see Ice Age 3?)*, a music query *(I want to listen to Abba)* or a travel related query *(How can I reach Piccadilly from London?)*. Namely, $\mathcal{O}^S = [yellowpages, movie, music, travel]$. The input query in natural language was first broken into tokens using an n-gram analysis and each of the token was then tagged to one of the $R = 7$ tags, namely, $\mathcal{I}^R = [actorname, address, moviename, singername, composername, keywords, businessname]$. In both the cases $\mathcal{R}_h$ and $\mathcal{R}_s$, the query in natural English was broken into terms and each term was assigned a tag from $\mathcal{I}^R$.

[3]We will not describe the rule-base generation mechanism adopted because it is not central to the theme of the paper

So the input to the $\mathcal{R}_h$ and $\mathcal{R}_s$ was an input $\mathcal{I}^R$ of $R = 7$ dimension binary vector. For example, if a tag was present it was marked as 1 else it was marked as 0; so a $\mathcal{I}^R = [1, 0, 1, 0, 0, 0, 0]$ had the tag $actorname$ and $moviename$ present while the other 5 tags were absent. The rule-base took this input and produced an output $\mathcal{O}^S = [o_1, o_2, o_3, o_4]$ if o_2 was the maximum then the input would be identified as being a $movie$ query. An error (ϵ) in input is introduced by creating



(a) $F_{\mathcal{C}}$ for system generated rule-base.



(b) $F_{\mathcal{E}}$ for system generated rule-base.

Fig. 3. Cumulative distribution ($F_{\mathcal{C}}$, $F_{\mathcal{E}}$) for R_s.

all $\mathcal{I}_\epsilon^R$ which are at a Hamming distance of 1 from $\mathcal{I}_R$. Namely, for $\mathcal{I}^R = [1, 0, 1, 0, 0, 0, 0]$, $\mathcal{I}_\epsilon^R$ generated was $[0, 0, 1, 0, 0, 0, 0]$, $[1, 1, 1, 0, 0, 0, 0]$, $[1, 0, 0, 0, 0, 0, 0]$, $[1, 0, 1, 1, 0, 0, 0]$, $[1, 0, 1, 0, 1, 0, 0]$, $[1, 0, 1, 0, 0, 1, 0]$, $[1, 0, 1, 0, 0, 0, 1]$.

Note 7: For the purpose of analysis we have considered an error in only one element at a time (Hamming distance 1); though this can be generalized to have more than one error in the tagging process

Now each of the erroneous inputs was passed to the rule-base system and D_ϵ was computed and was marked as belonging to C_0 or C_1 or C_ϕ. The cumulative plots $F_{\mathcal{C}}$ (all $\mathcal{I}_\epsilon^R \in C_0$) and $F_{\mathcal{E}}$ (all $\mathcal{I}_\epsilon^R \in C_1$) are shown in Figures 3(a) and 3(b) respectively. While it is difficult to see due to the resolution of plot (Figure 3(a)) for $\mathcal{R}_s$ only $k = 6$ erroneous inputs with errors fell in category C_0 with a $D_\epsilon = 0$ compared to $k = 106$ with $D_\epsilon = 0$ for the hand crafted rule-base system $\mathcal{R}_h$. The number of erroneous inputs which fell in category C_1 for $\mathcal{R}_h$ with $D_\epsilon = \frac{1}{\sqrt{2}}$ was 92 compared to 32 for the system generated rule-base system $\mathcal{R}_s$. Clearly the hand crafted rule-base ($\mathcal{R}_h$) is more robust than the system generated rule-base ($\mathcal{R}_s$) in handling errors in the input. This is intuitive and as expected.

Note 8: Note that the hand generated rule-base has $\mathcal{O}^S$ which are binary and hence if an erroneous input is in category C_0 then $D_\epsilon = 0$ and if the erroneous input is in category C_1 then $D_\epsilon = \frac{1}{\sqrt{2}}$. Hence the choice of $D_\epsilon = \frac{1}{\sqrt{2}}$ above.

IV. Conclusions

In this paper we looked at an expert or a rule based system as a function which maps an input to an output based on the rule-base supporting the system and emphasized that the input to the expert system is not always correct because the input it self could have been generated from another process. We first formulated a metric to measure the robustness of a rule-based system to errors in the input and showed that the metric involves the calculation of two cumulative functions, namely, $F_{\mathcal{C}}$ and $F_{\mathcal{E}}$ to determine the robustness of a rule-base system. We demonstrated the utility of measuring the robustness of a rule-base system by constructing a rule-based system to identify the intent of a query and showed that the system based on hand crafted rules is more robust than the rule-based system supported by rule-base which was system generated.

Acknowledgments

We would like to thank Dr Arijit De of the TCS Innovation Labs - Mumbai for implementing the rule based system for both hand crafted rule base and system generated rule base and conducting the experiments.

References

[1] "Rule based systems," World Wide Web electronic publication, Accessed in Apr 2010. [Online]. Available: http://www.ramalila.net/Adventures/AI/rule_based_systems.html

[2] D. A. Kaur and K. Kaur, "Fuzzy expert systems based on membership functions and fuzzy rules," *Artificial Intelligence and Computational Intelligence, International Conference on*, vol. 3, pp. 513–517, 2009.

[3] J. Freeman-Hargis, "Methods of rule-based systems," World Wide Web electronic publication, Accessed in Apr 2010. [Online]. Available: http://ai-depot.com/Tutorial/RuleBased-Methods.html

[4] P. G. Chander, R. Shinghal, and T. Radhakrishnan, "Incremental and integrated evaluation of rule-based systems," in *IEA/AIE*, ser. Lecture Notes in Computer Science, I. F. Imam, Y. Kodratoff, A. El-Dessouki, and M. Ali, Eds., vol. 1611. Springer, 1999, pp. 276–285.

[5] A. D. Preece, C. Grossner, and T. Radhakrishnan, "Validating rule-based systems that operate with incomplete data," in *EUROVAV*, M. Ayel and M.-C. Rousset, Eds. ADERIAS-LIA, Universite de Savoie, 1995, pp. 77–90.