Fuzzy Computing Applications for Anti-Money Laundering and Distributed Storage System Load Monitoring

Yu-To Chen, Johan Mathe
Google, Inc. 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA
{ytchen, johmathe}@google.com

***Abstract*—Fuzzy computing (FC) has made a great impact in capturing human domain knowledge and modeling non-linear mapping of input-output space. In this paper, we describe the design and implementation of FC systems for detection of money laundering behaviors in financial transactions and monitoring of distributed storage system load. Our objective is to demonstrate the power of FC for real-world applications which are characterized by imprecise, uncertain data, and incomplete domain knowledge. For both applications, we designed fuzzy rules based on experts' domain knowledge, depending on money laundering scenarios in transactions or the "health" of a distributed storage system. In addition, we developped a generic fuzzy inference engine and contributed to the open source community.**

I. Introduction

A. Motivation

There is a wide variety of industrial and financial problems which require the analysis of uncertain and incomplete information. To make matters worse, data used for the analysis are often imprecise. These problems present a great opportunity for the application of fuzzy computing technologies.

In a distributed storage system, it is imperative to understand the usage patterns such as memory and CPU, so as to better load balance and avoid bottleneck. This can be accomplished by using a tool which measures the state of the system and reports the overall "healthiness" of the system. Such a tool ought to have a high level of sophistication which incorporates real-time monitoring and decision-making. In a financial institution, we can still find labor-intensive tasks such as review of suspicious account activities based on alerts triggered by a rule-based system with hard-coded cutoffs. Due to the complexity of these tasks, artificial intelligence (AI) and in particular, fuzzy computing has been called upon in support of monitoring of a distributed storage system and detection of money laundering transactions. This paper focuses on the use of fuzzy computing on these two aspects: monitoring and detection. For starters, we will give a brief overview of fuzzy computing in the next section.

B. Fuzzy Computing

The work of Post, Kleene and Lukasiewicz were among the first treatment of imprecision and vagueness in multiple-valued logic systems as oppose to the classical Boolean logic [1], [2]. In 1937, Max Black proposed the use of a consistency profile to represent vague concepts [3]. Most importantly, Zadeh offered a complete theory of fuzzy sets and fuzzy logic in 1965 [4], which enabled us to represent and manipulate ill-defined concepts. In addition, Zadeh defined fuzzy logic's four facets [5], which provided us a language with syntax and semantics for computation. In particular, fuzzy logic allows us to use linguistic variables to model dynamic systems by a set of fuzzy rules. Each rule consists of a set of linguistic variables. These variables take fuzzy values, which are characterized by fuzzy membership functions. In addition, there is a reasoning mechanism, fuzzy inference engine, which operates on the fuzzy rules based on the generalized modus-ponens [6]. A comprehensive review of fuzzy logic and fuzzy computing can be found in [7].

C. Paper Structure

In the next section we will focus on FC and their applications in monitoring and detection. After a brief discussion of the problem of monitoring and detection in Section II, we will illustrate two applications of FC techniques. The first application, described in Section III, consists in the adaptation of fuzzy rules in monitoring of distributed storage systems. The second application, illustrated in Section IV, covers the use of fuzzy logic inference to detect anomalies in money laundering. In the last section V we summarize the advantages of using FC and discuss some potential extensions of these technologies.

II. Fuzzy Computing Applications for Monitoring and Detection

In this paper, we studies two fuzzy computing applications in the areas of monitoring and detection. They are fuzzy load monitoring for a distributed storage systems and fuzzy anti-money laundering for a financial institution. In general, monitoring is the first step to understand any complex real-world system. If certain undesired cases or scenarios were discovered during the monitoring process, the next logic step would be to detect the recurring patterns, and subsequently to try to isolate the issues. Finally, a control strategy could be formalized to manage the problems.

Table I
FUZZY RULES FOR LOAD MONITORING

Memory\CPU	LOW	AVG	HIGH
LOW	HVG	HG	HB
AVG	HG	HG	HB
HIGH	HB	HB	HB

III. FUZZY LOAD MONITORING FOR DISTRIBUTED STORAGE SYSTEMS

A. Problem Description

Google File System [8] is a distributed file system. A GFS cluster consists of a single master and multiple chunkservers (nodes) and is accessed by multiple clients. Different GFS clusters can have very different usage patterns, and these usage patterns are not always correlated with the bytes usage of the cluster. For instance, a cluster used as a backup storage will have a very strong bytes usage with a very low traffic, but a GFS cluster serving live traffic will see a very large throughput without necessarily having an important amount of bytes used.

Even though GFS has been designed to avoid the master being a bottleneck[8] , the CPU load and memory usage of the master still increases as the traffic and the number of chunkservers in the cluster increases.

Our goal is to use a fuzzy inference system that will give us a good idea of the state of the GFS masters for multiple purposes:

- Provisioning: if the GFS master can be upgraded with better hardware if possible
- Re routing users: if some big users can be moved to other clusters

B. Approach

We noticed that during the on-call rotation of a system administrator, there is a usually lot of qualitative thinking involved. Fuzzy logic here mimics this qualitative thinking to allow faster incident response or capacity planning regarding the GFS masters.

After gathering the expert knowledge of various system administrators on the topic, here are some of the following fuzzy sets we found:

- Memory usage high, memory usage low, memory usage average
- CPU usage high, CPU usage low, CPU usage average
- GFS Master Latency is high
- Health is very good (HVG), health is Good (HG), health is bad (HB)

Rules regarding CPU and Memory have been gathered in table I.

Then we use one more rule related to the system latency:

- If RPC latency high then health is bad

C. Prior Work

Most network monitoring systems [9] today work by comparing measurements from hosts (such as latency, CPU usage, error rates etc.) to particular thresholds, and alerting via email or pager, depending on the severity of the event. This approach, when applied to massive distributed systems, requires an extremely deep understanding of the underlying system, and in the case of correlated events, leaves all interpretation to the operator. The fuzzy load inference system is able to give an higher level view of the distributed system.

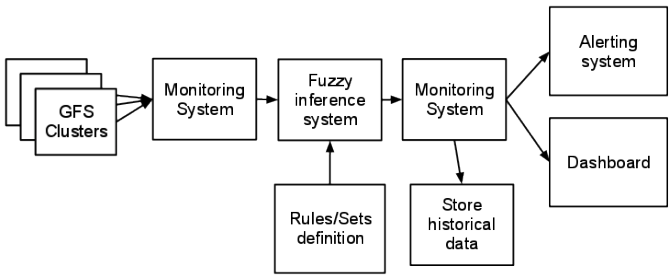

Figure 1. Monitoring system data flow

D. Solution Description

In order to integrate a fuzzy inference system into google's monitoring infrastructure, we had to write a production quality inference system in Python[17], that we called GFuzzy. The rules are defined via protocol buffers – Google's data interchange format [10].

The python inference engine provides different fuzzification and deffuzification functions [17]. We used trianglular fuzzy sets for the fuzzification and the centroid method for deffuzification.

The workflow is the following, as also described in figure 1:

- A monitoring system collects data from multiple GFS clusters
- The fuzzy inference system polls the data from the monitoring system, applies the fuzzy rules, and produces an output
- This output is then itself collected by another monitoring system
- Alerts are sent to system administrators and capacity planners when values go over a certain threshold.

1) Inputs pre-processing: Before converting monitoring data into the fuzzy set, we average the data by using a gaussian window on the last N days of data. This has the effect to smooth data spikes and filter out high frequency noise.

E. Results

The system has been running for one year, showed some initial false negative and positives. Our methodology for finding these was to compare the decisions made by the on-call engineer and the results of the fuzzy inference system. We performed the following improvements in the first iterations:

- The initial implementation only contained RAM and CPU related fuzzy sets. We added the latency set because these are very relevant to the health of the GFS masters.

- We increased the size of the window and switched from a rectangular window to a gaussian window because the rectangular window was too sensible to high frequency noise.
- We tweaked some of our membership functions over time, by looking at our false positive/negative, every time by asking the oncall engineer what were the thresholds that made him decide which data was the root cause of the problem regarding the GFS masters.

F. Conclusions on Fuzzy Load Monitoring

We successfuly designed, implemented and deployed a fuzzy inference engine for monitoring the masters of a distributed file system. Future work involves measuring the nodes (chunkservers [8]) of the file system, and using health of the master and the nodes to automatically determine available capacity and availability of the GFS cluster.

IV. Fuzzy Detection for Anti Money Laundering

A. Problem Description

Anti-money laundering (AML) was implemented in the U.S. by the Bank Secrecy Act of 1970. AML refers to the legal controls that require financial institutions and other regulated entities to prevent or report money laundering activities [11]. According to Wikepedia [12], AML is a term “mainly used in the financial and legal industries to describe the legal controls that require financial institutions and other regulated entities to prevent or report money laundering activities.” In US laws, money laundering includes all financial transaction generating an asset or a value as the result of an illegal act. For example, tax evasion and false accounting. All the financial institutions are required to identify transactions of a suspicious nature and report to the financial intelligence unit in their country [13].

One popular approach [14], [15] is to apply scenarios and risk factors to transactions to detect potentially suspicious activity. Transactional events that meet the rule parameters become alerts. Finally, alerts are subject to additional workflow processes, such as suppression, risk scoring and routing. In essence, it’s a rule-based system and depends heavily on human domain knowledge. As well documented in the literature, a rule-based system suffers from rigid, brittle and inflexible rule conditions. With all the benefits coming with superior human cognitive capability, it inherits the imprecise nature of human linguistic expressions.

B. Approach

Fuzzy logic comes to rescue. It is invented for such purpose as to deal with impression in human’s nature language. For instance, we intuitively understands statements like "the food is delicious" and "the service is excellent." In addition, we can even make up rules such as "if the food is delicious and the service is excellent, then the tip would be handsome." All the linguistic terms such as "delicious", "excellent" and "handsome" are fuzzy in nature. Your deliciousness possibly is not the same as mine. However, fuzzy logic defines degree of "deliciousness", so that it can develop an inference engine based on these linguistic terms later on.

We began the fuzzy AML work by assuming a hypothetical financial corporation, XCorp, whose business involves servicing clients for sending and receiving money via the Internet. A significant body of knowledge in Money Laundering (ML) was amassed via knowledge engineering. In practice, it was done through systematic identification of suspicious transactions by customer service representatives, labor-intensive case studies by analysts, and finally, generalization of ML indicators by both aforementioned parties. The followings showed the main data sources:

- From existing fraud models, which isolate fraudsters who might be ML risk as well
- From customer service representatives who view accounts and are in contact with customers
- From analysts who conduct web searches looking for questionable websites accepting/offering XCorp payments
- From customers or third parties who become aware of suspicious activity, such as when customers report account takeover

A case study was shown as follows to highlight a real-world example:

- Suspect, using a UK postal address developed a pattern of creating XCorp accounts, receiving funds, sending funds to several French accounts, funds were then withdrawn to bank accounts. Receiving and sending accounts were closed after transactions were made. 25 accounts were identified, with a total transaction amount involved of $32K. Sending and recipient accounts shared IPs and machine fingerprints.

From the case study, a number of ML indicators was generalized:

- Multiple accounts controlled by one party
- Lots of account activities within a short time window, such as opening - sending - closing and opening - receiving - withdrawing - closing
- No viable business reasons

C. Prior Work

There is no significant prior work on fuzzy anti-money laundering (AML) in academic journals and proceedings. However, there are a number of commercial software for AML, such as SAS, Actimize, tellemetrix, just to name a few. As discussed, one popular approach is to apply risk factors to transactions to detect potentially suspicious activity via scenarios and case studies. Alerts will be issued if certain rate limiting criteria are met.

D. Solution Description

Our objective was to build a fuzzy inference system for AML. For demonstration, we would describe an AML scenario, then translated it into fuzzy rules. After that, we would outline the fuzzy system’s membership functions. Finally, results of fuzzy inference were summarized.

1) Fuzzy rules : The AML scenario is as follows. An account is suspicious if

- large value received (>= $10,000)
- immediate withdraw (within 1-3 days)

Let's label this scenario as AML1. Therefore, we were aiming to design a fuzzy AML scoring, which takes amount received and a match score (the degree of match for $ received and $ withdrawn) as input, and gives AML1 score as output.

- AML1 <- fuzzy.inference(amt.received, match.score)

The idea was to come up with a score from amount received and match score. For instance:

- AML1 score is very high, if amount received is big and match score is high
- AML1 score is very low, if amount received is small and match score is low

Therefore, the AML1 score is a real number bounded by [0,1],which represents the possibility of the account is a ML violation.

Note that the match score represented the degree of match for $ received and $ withdrawn in a time window. In essence, the score was a function of three parameters: amount received, amount withdrawn and time window. For instance

- Match score is high, if $ withdrawn is within [80%, 120%] of $ received
- Match score is high, if $ withdrawn is within [80%, 120%] of $ received
- Match score is high, if $ withdrawn immediately after $ received
- Match score is zero, if $ withdrawn after 3 days $ received
- Match score is moderate, if $ withdrawn within 1-3 days after $ received

In essence, *match score* is a real number bounded by [0,1]. The higher the score, the closer the match of the two dollar amount within 1-3 days. Let's define a difference measure as $\left|\frac{a-b}{b}\right|$, and label it as "absolute % change," where a and b are the $ withdrawn and received in a time window, respectively. Three levels of absolute % change scores were calculated as there were three different time windows: 1-, 2- and 3-day. In addition, a "match factor" was defined and it took integer values in {1,2,3}, representing three levels of time windows.

- Match.score = 0.9(1-abs.%.change/0.2)(match.factor-1), if abs.%.change <= 0.2
- Match.score = 0, if abs.%.change > 0.2

Essentially, the score got 10% and 19% discount, if $ withdrawn in 2 and 3 days, respectively. Refer to figure 2 for details.

Assume that the AML1 score is a function taking two arguments - amount received and match score. Further, the domain of both arguments is a real number bounded in [0,1]. The idea for fuzzy scoring is to segment the space into disjoint sub-regions with descending average AML1 scores first, then use fuzzy inference to smooth out/make interpolations of scores from region to region. For AML1, nine fuzzy rules

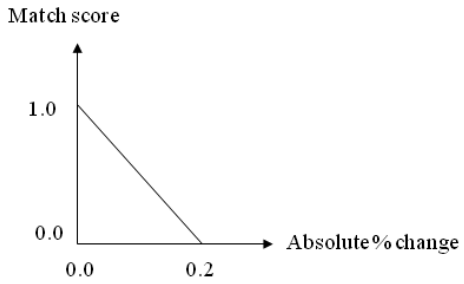

Figure 2. Absolute change percentage vs match score

Table II
AML SCORE FOR EACH OF THE 9 FUZZY RULES

Amount Rcvd\Match score	S	M	B
L	M	H	VH
M	L	M	H
S	VL	L	M

were constructed for the inference, as shown in table II. For instance:

- If amt received is big and match score is high, then AML1 is very high
- If amt received is medium and match score is medium, then AML1 is medium
- If amt received is small and match score is low, then AML1 is very low

2) Fuzzy membership functions: As described in the last section, there were three fuzzy sets: amt received, match score and AML1, where amt received and match score were antecedent, and AML1 was consequence. Their membership functions were defined as shown in figures 3, 4 and 5, respectively.

E. Experimental Results

We tested the fuzzy AML1 scoring on some sampled accounts and their transactions. Of the 710 accounts got scored, the median AML1 was 0.55. 73 accounts whose AML1 score > 0.8 were routed to a AML queue for human reviewing. The queue would be worked in descending order of AML1 scores.

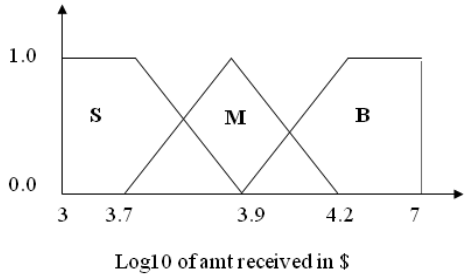

Figure 3. Amount received membership functions

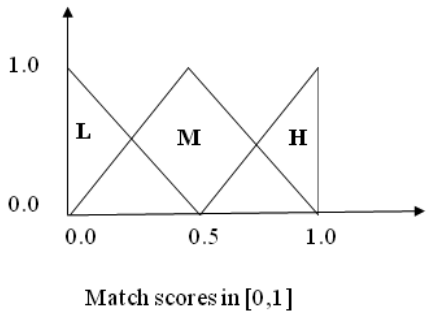

Figure 4. Math score membership functions

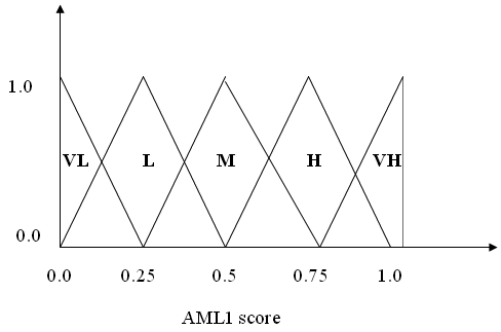

Figure 5. AML1 membership functions

For demonstration, two suspicious accounts were described as follow.

For row 1 in table III, its $ received > $10K (Big) and final match score = 1 (Large), so its AML1 score = 1 (Very Large). In this example, its initial MatchScore =1 since its absolute % change = 0 ($ received = $ withdrawn). In addition, MatchFactor = 1 since all the withdrawals logged in a day after $ received. Therefore there is no discount for the initial MatchScore. As a result, the final MatchScore is the same as the initial one.

For row 2 in the table III, its $ received > 10K (Big) and final match score = 0.83 (Large), so its AML1 score = 0.92 (Large). In this example, its absolute percentage of change change = 1.5% ($ received > $ withdrawn), hence its initial MatchScore = 0.92. However, all the withdrawals logged within 2 days after $ received, hence its MatchFactor = 2. Thus it implied that there was a 10% discount to the initial MatchScore. As a result, the final MatchScore = 0.83 (= 0.92×0.9).

F. Conclusions on Fuzzy Detection

We have presented an approach that uses fuzzy computing to detect money laundering (ML) patterns in complex financial transactions. We showed the process of knowledge engineering for intelligence gathering and understanding of patterns of ML.

Table III
EXPERIMENTAL RESULTS FOR AML

Act	$ recv	#wtxn	$ wtx	match	%change	MScore1	MScore2	AML1Score
1	$35,306	2	$35,306	1	0.0%	1.00	1.00	1.00
8	$25,713	3	$25,326	2	1.5%	0.92	0.83	0.92

After that, we described the design and development of a fuzzy inference system (FIS), which takes as input the transactions, and gives as output the scores of suspicious ML behaviors. The FIS provides an intuitive and robust way to combat ML, while fulfills the obligations for financial institutions set by the authority. Future work will focus on automatic tuning of fuzzy rules and membership functions based the misclassification rates and human agent's feedback.

V. FINAL REMARKS

Fuzzy computing (FC) is having an impact on many industrial and financial operations, from monitoring and predictive modeling to diagnostics and control [16]. It provides us with alternative approaches to traditional knowledge-driven reasoning systems and it overcomes their main flaws in the rigidness of the rule structure. We have demonstrated two successful real-world deployments of FC applications in monitoring and detection. In particular, we described how to monitor the healthiness of a distributed file system and how to detect the suspicious transactions in money laundering. Both systems leverage the tolerance for imprecision, uncertainty and incompletness, which is the hallmark of the problems to be solved. In addition, we developed a generic fuzzy inference engine and contributed to the open source community[17]. In the future, we expect the combination of fuzzy computing with advances in probabilistic reasoning, voice recognition, text processing and computer vision, etc., will further improve and expand our problem-solving capability for a large spectrum of industrial and financial problems.

REFERENCES

[1] N. Rescher, *Many-valued Logic, McGraw-Hill*, New York, NY, 1969.
[2] J. Lukasiewicz, Elementy Logiki Matematycznej *Elements of Mathematical Logic*, Warsaw, Poland: Panstowowe Wydawinctow Naukowe,1929.
[3] M. Black, *Vaguenes: an Exercise in Logical Analysis*, Phil.Sci. vol. 4., pp-427-455, 1937.
[4] L.A. Zadeh, *Fuzzy sets*, Information and Control, vol. 8, pp.338-353, 1965.
[5] L.A. Zadeh, *Foreword*, in Handbook of Fuzzy Computation, E.H. Ruspini, P.P. Bonissone, and W. Pedycz, Eds., Bristol, UK: Institute of Physics, 1998.
[6] Y-M. Pok and J-X. Xu, *Why is Fuzzy Control Robust*, in Proc. Third IEEE Intl. Conf. on Fuzzy Systems (FUZZ-IEEE'94), pp. 1018-1022, Orlando, FL, 1994.
[7] E.H. Ruspini, P.P. Bonissone, and W. Pedycz, *Handbook of Fuzzy Computation*, Bristol, UK: Institute of Physics, 1998.
[8] Ghemawat, S., Gobioff, H., And Leung, S.-T. *The Google file system*, In Proc. of the 19th ACM SOSP (Dec. 2003), pp. 29-43.
[9] Wikipedia, *Network monitoring*, http://en.wikipedia.org/wiki/Network_monitoring.
[10] Google, *Protocol Buffers - Googleś data interchange format*, http://code.google.com/p/protobuf/ .
[11] Paul Allan Schott *Reference Guide to Anti-Money Laundering and Combating the Financing of Terrorism*, World Bank, 2006.
[12] Wikipedia, *Anti Money Laundering*, http://en.wikipedia.org/wiki/Anti-money_laundering
[13] Jackie Harvey, (2005) *An evaluation of money laundering policies*, Journal of Money Laundering Control, Vol. 8 Iss: 4, pp.339 - 345.
[14] SAS *Anti Money Laundering* , http://www.sas.com/industry/financial-services/banking/anti-money-laundering
[15] Kingdon, J., *AI fights money laundering*, Intelligent Systems, IEEE, Vol. 19, Issue 3, May-Jun 2004, pp.87 - 89.
[16] Bonissone et al. *Hybrid Soft Computing Systems: Industrial and Commercial Applications*, Proceedings of the IEEE, 1999.
[17] *GFuzzy*, http://code.google.com/p/gfuzzy/