Genetic Evolution of Expert AI Systems in Financial Underwriting

Tom Pawelek (tompaw@tompaw.net), Mississippi State University
Rym Oulad Ali, ENSEA

***Abstract* - In the first part of this paper, we describe an existing sequential/DL based algorithm for loan underwriting. Using the available research sources, we develop a CLIPS based equivalent, using the fuzzy green/yellow/red flags for risk assessment. Afterwards, we run our expert system process on a large number of real-life leads and compare its predictions with the actual performance of resulting loans. In the second part, I compare how well the expert system predictions performed vs the existing algorithms. Afterwards, I propose and implement a genetic evolutionary algorithm to tune the parameters of a simplified expert system.**

Index Terms - Algorithm Performance, Clips, Expert Systems, Fintech.

Disclaimer - Sections "Fuzzy Expert System" and "Implementation" have been co-authored with Rym Oulad Ali from Mississippi State University.

INTRODUCTION

US financial markets are driven by lead generators, which are companies who invest large amounts of capital into the marketing of financial products, such as insurance policies, investments and loans. The actual providers of those services are then offered leads and given a limited amount of time (usually 10-30 seconds) to decide whether to purchase or pass on a given lead.

The risk assessment algorithms need to answer 2 questions: (1) Should a given lead be purchased (based on limited information), and (2) Should a loan be offered to a given individual (based on more information collected during the underwriting). We are going to describe those algorithms and test an expert system approach to those two determinations.

EXISTING SEQUENTIAL FLOW

In our experiments, we are going to focus on low value loans (under \$2,000), since those tend to be evaluated in a fully automated fashion. This covers a lower mid-prime spectrum of financial services with APRs varying from double to triple digits.

Most lenders in this space acquire leads from so-called "ping tree" exchanges, where a limited set of information about a borrower is first presented for a duration of 10-30 seconds (depending on price point and quality). When purchased, the borrower is redirected to lender's website to continue a more thorough, interactive underwriting. This second phase is not time limited (other than by behavioral factors) and results in a funding decision or a decline.

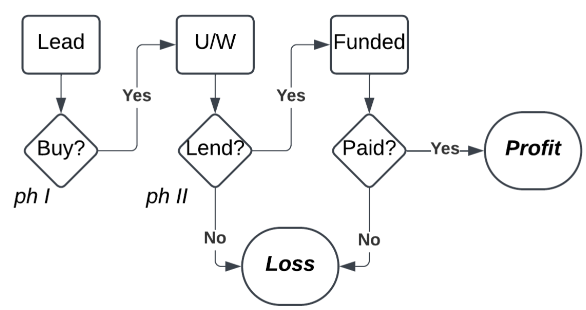

FIGURE I
TYPICAL LENDING FLOW IN LOW/MID PRIME MARKET
(U/W = UNDERWRITING)

We have shown the typical underwriting flow in Figure I. Phase I shows the initial lead purchase decision, and phase II shows the more complex funding decision.

The two detrimental scenarios we are trying to avoid are:

a) Purchasing the lead and not funding the loan
b) Funding the loan and not having it repaid

Of these two, the latter is associated with a larger loss (due to default). Therefore, we would rather decline a loan if the applicant were found not to have sufficient means to repay and limit our losses to the price of lead purchased.

In this paper, we will be focusing on phase II, which occurs after all the additional information has been collected from the borrower. We consider a simplified sequential model as our reference. In this model, the final underwriting outcome is defined as:

$$V = \prod_{i=0}^{R} r_i \qquad (1)$$

Where V is the final decision value, R is the number of sequential steps and r_i are values of individual underwriting steps. Positive decision (fund / proceed) has a value of 1, while negative decision (decline / stop) has a value of 0.
Based on (1), we see that in this simplified model, for a Loan to be funded, every underwriting step needs to have a positive outcome.

UNDERWRITING STEPS

There are three categories of verification steps, which we briefly describe below:

I. Open Value

These are processes which evaluate pseudo-continuous values which, when normalized, belong to a range $[0, +\infty)$. Examples include:

- Declared monthly income
- Verified monthly income
- Bank account age (in months)
- Applicant age (subject to laws & regulations)
- Requested loan amount
- Number of other loans found in bank statement

II. Limited Value

In this type of underwriting steps, our subject value belongs to a predetermined range [x, y]. Examples are:

- Credit score (300-850)
- Propensity to pay score (a-f)

III. Enumerated Value

Finally, we have parameters which values belong to a predetermined set, such as:

- Bankruptcy record (yes / no)
- Criminal record (yes / no)
- Payroll frequency (weekly / biweekly / monthly)
- Email domain (google / microsoft / yahoo / other)

A parameter is considered enumerated if its representation does not carry intrinsic value. For example: propensity to pay has a range of discrete grades from A to F, which correspond to decreasing probability of a given applicant not defaulting on their repayment plan. On the other hand, a payroll frequency does not in itself carry a meaning directly related to someone's ability to repay a loan.

TEST METHODOLOGY

The expert system was developed as an alternative to the existing sequential platform. In order to evaluate its effectiveness, we analyzed a real $13M portfolio of lower mid-prime retail loans. Our dataset contains over 20 thousand accounts originated since 2023. We have limited the scope of our analysis to the accounts which met the following criteria:

- Originated over 180 days ago, to ensure maturity (avg contract length is 22 weeks)
- Contain a recent bank account statement showing payroll deposits, to provide a verified income amount
- Do not have any missing data in their source leads
- Monthly payroll (both declared and verified) is in the range of $1,000 to $10,000

For every loan, we run the lead information through the expert system and compare its approval decision to the actual performance of a given loan. We consider the decision to be correct if either:

- Decision was negative (NO) and the loan resulted in a default,
- Decision was positive (YES) and the loan was profitable.

We defined a fitness function f(X) as sum of all the points for a given expert system X. We also calculated the value f_B as a base fitness performance of the existing sequential platform using the formula:

$$f_B = L - D \qquad (2)$$

Where L is the total number of loans being analyzed and D is the total number of defaulted loans. To ensure a proper representation of defaulted loans in our set, we have scoped a training set with the values of L = 14,893 and D = 7,402, yielding a reference value of f_B = 7,491.
Since a substantial part of the sequential process is supplemented by a manual override of loan officers, those numbers represent the maximum fitness score of a human intuition factor.

FUZZY EXPERT SYSTEM

The system uses a rule-based fuzzy inference approach and was implemented using FuzzyCLIPS integrated with Python via the CLIPSpy library. The system evaluates loan applications based on various applicant attributes and generates one of three decisions: YES, MAYBE, or NO.

Fuzzy expert systems have been shown to be capable of extracting the knowledge of bank customer credit scoring [1], therefore our rule-based flow should be a good candidate for this approach.

IMPLEMENTATION

The choice of a fuzzy expert system is directly related to the uncertainty that is inherent in decision-making for loans. Fuzzy logic enables the system to handle vague concepts like “high income” or “low credit score” by defining fuzzy sets and rules that can assess the input data against these sets.

For the threshold values, we have contacted field experts involved in our scoped portfolio and attempted to extract the knowledge from their long-term experience. This approach of extracting knowledge from a single source is commonly preferred in a narrow scope environment of financial underwriting. [2]

We developed our rule-based fuzzy inference system using FuzzyCLIPS, integrated with Python via the CLIPSpy library. This system assesses loan applications by evaluating key attributes such as credit score, income, and financial history. Based on this analysis, the system outputs one of three recommendations mentioned before.

I. Inputs and Outputs

The system takes the following **inputs**:

- Declared Income: Applicant’s self-reported income.
- Confirmed Income: Income verified via bank statements.
- Income Frequency: Weekly, biweekly, or monthly.
- Bank Account Age: Number of months since the account was opened.
- Applicant Age: Applicant’s age in years.
- Credit Score: A numeric value between 350 and 850.
- Microbilt Rate: A categorical risk rating (A to F).
- Email Domain: Applicant’s email provider (e.g., Google, Microsoft, Yahoo, or others).
- Requested Amount: The amount of the loan requested.
- Number of Other Loans Detected: Concurrent loans found in the applicant’s financial profile.
- Bankruptcy History: A boolean flag indicating whether the applicant has a history of bankruptcy.

The **output** is the loan underwriting decision, expressed as a fuzzy variable with the possible values of YES, MAYBE, or NO.

II. Fuzzy Sets and Membership Functions

The degree of membership functions is decided by a membership function, which maps each input value to a membership degree. The most common membership functions’ shapes are triangular, trapezoidal, and Gaussian functions.

Below are some inputs and the output within their universe of discourse and the membership functions we can use to describe them:

Credit Score
Membership Functions: Triangular

- Low: [350, 400, 500]
- Medium: [500, 600, 650]
- High: [650, 750, 850]

Declared Income
Membership Functions: Trapezoidal

- Low: [0, 0, 2000, 3000]
- Medium: [2000, 3000, 5000, 6000]
- High: [5000, 6000, ∞]

Confirmed Income
Membership Functions: Trapezoidal

- Low: [0, 2000, 3000]
- Medium: [2000, 3000, 5000, 6000]
- High: [5000, 6000, ∞]

Propensity to Pay Score
Membership Functions: Triangular

- A: [0, 0, 1]
- B: [0, 0.25, 1]
- C: [0, 0.5, 1]
- D: [0, 0.75, 1]
- E: [0, 0.9, 1]
- F: [0, 1, 1]

Output
Membership Functions: Triangular

- YES: [0, 0, 1]
- MAYBE: [0, 0.5, 1]
- NO: [0, 1, 1]

III. Rule-Based Decision-Making

The rule-based decision-making process is at the core of the fuzzy expert system. It integrates the fuzzified input variables, evaluates them against predefined decision rules, and generates outputs that align with the desired classification categories (YES, MAYBE, NO). This system uses the following approach:

Input Evaluation: Inputs, such as credit score, income, and financial history, are fuzzified into linguistic variables (e.g., Low, Medium, High).
For example, a credit score of 600 would belong partially to the "Medium" category with a membership degree of 0.8 and "High" with 0.2.

Decision Rules: Rules are written to encode domain knowledge into the decision-making process.

Inference Process: Rules are evaluated in parallel using a Mamdani-style inference engine. Each rule's outcome contributes to the aggregate decision-making process.

Defuzzification: The fuzzy output from the applicable rules is aggregated. A crisp output is generated by selecting the decision category with the highest membership value.

TEST OUTCOMES

Our initial runs of the expert system yielded a fitness score of 5,150, which represents a decline of 30% compared to our reference base value.

Further improvements to the rule set allowed us to increase this result up to a maximum value of 7,222, which stands at 96.4% of the human-assisted process.

To determine the business validity of this approach, we must analyze the potential losses of capital associated with increased defaults (or, to a smaller extent – loss of opportunity). Once factored in, the results for our expert-based system were:

- Capital loss: $212k (ca 2% of face value)
- Labor savings over 2 years: $120k

Overall, replacing loan officers with our basic expert system would result in a net loss of around $100k.

PERFORMANCE IMPROVEMENTS

As we worked on improving our expert system performance, we have noticed a point of diminishing (or even inverse) returns, where increased complexity of CLIPS rules resulted in minimal or even negative changes to our fitness function score.

Ultimately, we found ourselves working with an extremely basic rule set (for instance: declining past bankruptcies) combined with a list of fuzzy rules negating a loan for given sets of individual circumstances. This shows resemblance to our human-assisted protocol and might suggest that another approach is required if we want to improve our fitness results even further.

One proposed solution was to simplify the "fuzziness" of our expert system to only one parameter (income), keeping other factors discrete and then perform evolutionary training of this system. If we consider a neural network representation of an expert system, this would put our project on par with the most promising combination systems of fuzzy logic and neuro-computing. [3] This conclusion has been independently verified by multiple researchers in the field, including *Sreekantha and Kulkarni* [4].

The following chapters describe evolutionary experiments on a non-fuzzy expert system.

EVOLUTIONARY ALGORITHM & CHROMOSOME ENCODING

To use evolution as a way to improve my crisp expert system's parameters, I have implemented the following procedure, which mostly reused my input data from our previous experiments.

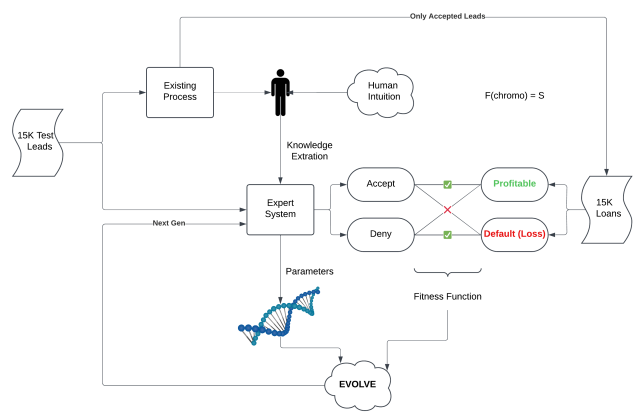

FIGURE II
USING GENETIC ALGORITHMS TO EVOLVE EXPERT SYSTEMS

The fundamental step in this approach is to encode a set of parameters of a given Expert System into a binary chromosome. Compared to its biological counterpart, with 4 possible DNA nucleotides for each bit (ACGT), binary encoding allows for only 2 values (0 or 1).

The most computational heavy step of this process is the fitness function evaluation for a given chromosome. In order to calculate our fitness value, we need to build an expert system using the chromosome-encoded parameters and then evaluate its performance using our previously defined fitness function f(X), where X is a CLIPS system encoded by a given chromosome. Our test sample size L ~ 15,000 loans.

Consequently, my goal was to maximize the efficiency of binary encoding by giving each chromosome bit an impactful role on the resulting expert system. In other words: the resolution of information had to be adjusted by what I perceived as appropriate for each parameter.

In practice, I normalized my values into two groups:

- Boolean switches: 1 bit for each Yes/No pair
- Numerical values: X bits for N values ($X = \log_2 N$)

To maximize information use, numerical parameters were stratified into "power of 2" values. Resulting chromosome is shown in the table below:

Parameter	Bits
Income difference (declared vs actual)	3
Minimum income	3
Minimum credit score	4
Minimum bank account age	3
Minimum propensity to pay score	3
Minimum age	3
Maximum age	3
Number of other loans	3
Requested amount	2
Check income difference	1
Check minimum income	1
Check credit score	1
Check bank account age	1
Check propensity to pay score	1
Check age	1
Check number of other loans	1
Check requested amount	1
Check income frequency	1
Allow weekly income	1
Allow bi-weekly income	1
Allow monthly income	1
Allow weekly as bi-weekly income	1
Allow semi-monthly income	1
Check for past bankruptcies	1
TOTAL	**42**

In most cases, a given business factor was represented by a pair of Boolean + numerical values. For example: a 1-bit switch determining if our expert system should verify credit score and a 4-bit quantifier of the credit score value.

The parameters presented here were chosen for experimental purposes. US lenders have to abide by numerous regulations such as FHA (Fair Housing Act) and ECOA (Equal Credit Opportunity Act), which might restrict certain factors from being directly considered in the underwriting process.

The process itself is a classical evolutionary algorithm, with the following distinct phases:

- Fitness evaluation: calculating f(X) for each chromosome
- Genetic roulette wheel: picking candidates for the next generation based on randomized selection with each chromosome chance proportionate to normalized f(X) values
- Genetic crossover and mutation within the selected candidate pool

Factoring in resource constrains and the amount of data to process, I chose the following experiment parameters:

- Population size: 40 (equal to CPU cores)
- Generations: 200
- Crossover Rate: 0.7
- Mutation Rate: 0.025

Each fitness evaluation required a complete run of the expert system, resulting in over 120 million evaluations in order to complete the process.

EXPERIMENT RESULTS

Post-evolution, I observed the value of f_{MAX} = 8,956 with a converged $f_{CONVERGE}$ = 8,378. Those represent accordingly an increase of 19.55% and 11.84% over the base fitness performance of f_B. Net US Dollar gain is ~$650,000, which stands for 5% of the portfolio face value.

The evolution of expert system performance is plotted below:

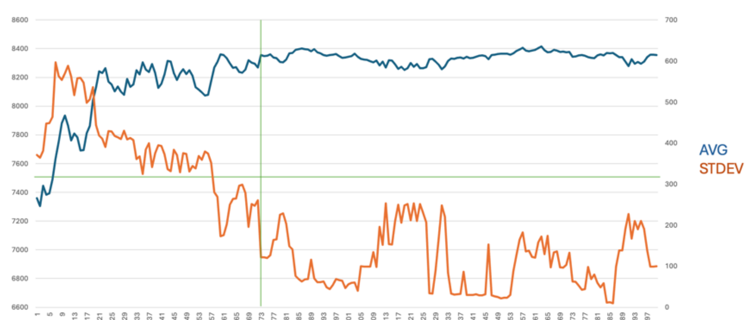

FIGURE III

200 GENERATIONS OF EXPERT SYSTEM PERFORMANCE

AVG – AVERAGE FITNESS ACROSS ALL POPULATION

STDEV – STANDARD DEVIATION

GREEN HORIZONTAL – REF. VALUE, VERTICAL – PLATEAU

Fitness score plateau was reached around generation 73 with f_{MAX} achieved in generation 99. Standard deviation minimum was the lowest between generations 183-185.

FINAL OBSERVATION & CONCLUSION

Genetically trained expert system was able to outperform human intuition very quickly. Potential concerns of local maxima or overfitted model were addressed by minimizing the standard deviation and re-using the evolved expert systems on other portfolios, including leads that have never been funded. Those tests yielded similar relative performance.

The classic genetic algorithm allows us to find models which perform very well, despite being different from human intuition and what would seem to be the most reasonable set of underwriting parameters.

Notably, by using a simplified crisp expert system, its role has effectively been reduced to a decision tree. In my follow-up experiments, I will attempt to replace some numerical variables resolution with fuzzy parameters. At the same time, it would be beneficial to keep the chromosome as short as possible, to maximize the chance of discovering maxima within our experimental populations.

In a long-term study, the population size itself is significantly increased, exceeding 10,000 chromosomes. Due to its computational intensity, the trial will take 6-12 months to complete.

References

[1] Ali Bazmara, Soheila Sardar Donighi, Bank Customer Credit Scoring by Using Fuzzy Expert System, International Journal of Intelligent Systems and Applications(IJISA), vol.6, no.11, pp.29-35, 2014

[2] Jay Liebowitz, The Handbook of Applied Expert Systems, CRC Press, 2019

[3] Rashmi Malhotra, D.K. Malhotra, Fuzzy Systems and Neuro-Computing in Credit Approval, The Journal of Lending & Credit Risk Management, 1999

[4] D. K. Sreekantha, R. V. Kulkarni, Expert system design for credit risk evaluation using neuro-fuzzy logic, 2010

Author Information

Tom Pawelek, PhD Student, Computer Science, Department of Computer Science and Engineering, Mississippi State University. Fintech CTO with experience in US lending and receivables space, head of R&D department focused on process automation and the use of generalized AI algorithms in risk assessment.

Rym Oulad Ali, Computer Science student, École Nationale Supérieure de l'Électronique et de ses Applications, Paris, France.