Note

Ben-PJ.Xu

April 19, 2017

Contents

1 What is a variable in C++? :cpp:

- a variable provides us with named memory storage that we can write to, retrieve, and manipulate
- a variable always denotes an object
- an object (or a variable) is always addressable
- there are two values associated with an object (or a variable):
  1. data value or rvalue or read value (literal constant can also serve as an rvalue
  2. address value or lvalue or location value

2 How to reflect OG under CIRC EV? :EV:

2.1 Example 1

Given the following scenarios:

t	scen 1	scen 2	scen 3
1	15	15	30

where scen 2 is the best estimate scenario, the $TVOG = E(CF) - CF^{BE} = 20 - 15 = 5$. When projection (on BEA) is conducted for EV, then the distributable earning -15($= -5 + 5 - 15$) and the EV = -15. In this case, OG is not reflected. So the EV should be $-15 - 5 = -20$.

2.2 Example 2

Given the following information:

1. a participating policy, whose reserve equals to 100
2. the total asset equals to 130
3. guaranteed interest rate equals to 2%
4. there are 3 interest rate scenarios:

scen 1	0%
scen 2	3%
scen 3	6%

   and "scen 2" is the best estimate scenario
5. RDR equals to 0%
6. p/h participating ratio equals to 100%
7. TVOG equals to

$$\begin{aligned} \text{TVOG} &= E(\text{CF}) - \text{CF}^{\text{BE}} \\ &= \frac{1}{3} \times (103 + 103 + 106) - 103 \\ &= 1 \end{aligned}$$

Calculate EV which equals NAV + ViF:

$$\begin{aligned} \text{NAV} &= 130 - 100 - 1 = 29 \\ \text{ViF} &= \text{CF} + \text{NII} - \Delta\text{RES} \\ &= -103 + 3 - (-101) \\ &= 1 \end{aligned}$$

But actually the EV should be equal to 29.

3 Classification problem in machine learning :ml:

Why I am confused about the classification problem is that y is discrete-valued, however a continuous function is used as hypothesis (**sigmoid function**).

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

where

$$g(z) = \frac{1}{1 + e^{-z}}$$

A property of the derivative of the sigmoid function:

$$\begin{aligned}
g'(z) &= \frac{d}{dz}\frac{1}{1 + e^{-z}} \\
&= \frac{1}{(1 + e^{-z})^2}(e^{-z}) \\
&= \frac{1}{(1 + e^{-z})} \cdot \left(1 - \frac{1}{(1 + e^{-z})}\right) \\
&= g(z)\,(1 - g(z))
\end{aligned}$$

How to fit θ?

1. endow our classification model with a set of probabilistic assumptions
2. fit the parameters via maximum likelihood

Assuming that

$$\begin{aligned}
P(y = 1|x;\theta) &= h_\theta(x) \\
P(y = 0|x;\theta) &= 1 - h_\theta(x)
\end{aligned}$$

more compactly,

$$p(y|x;\theta) = (h_\theta(x))^y (1 - h_\theta(x))^{1-y}$$

Likelihood of the parameters is as following:

$$\begin{aligned}
L(\theta) &= p(\vec{y}|X;\theta) \\
&= \prod_{i=1}^{m} p(y^{(i)}|x^{(i)};\theta) \\
&= \prod_{i=1}^{m} (h_\theta(x^{(i)}))^{y^{(i)}} (1 - h_\theta(x^{(i)}))^{1-y^{(i)}}
\end{aligned}$$

it will be easier to maximize the log likelihood:

$$\begin{aligned}\ell(\theta) &= \log L(\theta) \\ &= \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1-y^{(i)}) \log(1-h(x^{(i)}))\end{aligned}$$

how to maximize the log likelihood? **Gradient Ascent**:

$$\theta := \theta + \alpha \nabla_\theta \ell(\theta)$$

or **stochastic gradient ascent rule**:

$$\theta_j := \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$$

which means every time we are given a ith trainning sample, we update jth component of vector θ as last value of jth component of vector θ minus the product of α, $y^{(i)}$ minus h of ith x vector (using the last θ vector), and the jth feature in ith x.

4 Applying Newton's Method to fit θ given a likelihood function $\ell(\theta)$:ml:

To maximize $\ell(\theta)$, we should find the θ make $\ell'(\theta)$ equals to zero. Then Newton's method has the following update rule:

$$\theta^{(t+1)} := \theta^{(t)} - \frac{\ell'(\theta^{(t)})}{\ell''(\theta^{(t)})}$$

Matrix form:

$$\theta := \theta - H^{-1} \nabla_\theta \ell(\theta)$$

where H is called **Hessian**, whose entries are given by

$$H_{ij} = \frac{\partial^2 \ell(\theta)}{\partial \theta_i \partial \theta_j}$$

5 More generalised approach for a machine learning alogorithm :ml:

1. assuming the probability distribution for y

2. relate feature vector x to the probability/distribution function underlying y

3. write the expression for ℓ in term of θ (parameters underlying the probability/distribution function)

4. find the θ by maximizing the $\ell(\theta)$

6 Why extended formula? :prophet:

1. looping statement

2. use of local and global variables

3. use of user defined function

4. retention of values

7 How to use extended formula? :prophet:

1. use just like a normal variable, however more complex facilities are allowed (i.e. local variables, looping ...) — no "PUBLIC" declaration, no "PARAMETER" declaration, assign the calculated result to "VAR_NAME"

2. use as an in-memory table to improve efficiency, the following gives an example

   ```
   public INT_RATE(0) as number
   private YR as number
   private T_YR as number
   T_YR := 50
   for YR := 1 to T_YR
     INT_RATE(YR) := READ_GENERIC_TABLE(...)
   next
   ```

3. parameterised extended formulas actually provide a way of abstraction of calculation process. However, it cannot be called directly. Instead an instantiation of the parameterised formula is created within another formula. The variable associated with that formula can then be used to access variables with the parameterised formula.

```
#PRAM_EX_FORM("PARAMETER")
```

4. parameterised extended formula is macro I guess, when called, it should just perform a substitution of text.

8 Use of "CONSTRAINT" qualifier :prophet:

```
private VAR as text CONSTRAINT {"FOO", "BAR", "QUX"}
private VAR_ARR(10) as text CONSTRAINT {"FOO", "BAR", "QUX"}
```

9 Use of "PERSISTENT" :prophet:

Using "PERSISTENT" t-dependent extended variable will not require recalculation of this variable for certain period already calculated.

Another method could be used to achieve the same effect is to use a normal variable to retrieve the values calculated in the extended formula and referring the normal variable.

10 Use of enumeration :prophet:

An enumeration defines

1. a type
2. possible values (intergral constants)
3. the size of possible values

When used in the following way:

```
private MY_ARRAY(BOND_TYPE) as number
```

the size of possible values is used.
When used in the following way:

```
private SEX as SEX_TYPE
```

the enumeration actually define a "SEX_TYPE" type.

11 Performance issues :prophet:

1. do not use “READ_GENERIC_TABLE” with t-dependent index

2. do not use function “exp” or “″

12 Tips on Prophet :prophet:

1. distinguish between input variable and core variable
2. see which definition is actually used
3. see which indicators are attached with the product
4. when run an accumulation, previously produced results will also be included
5. the scope of variable shown when using result diagram view (standard output, AA00, . . .)
6. “VAR_NAME” in Prophet is just as macro
7. it seems there is no way to access the variable indirectly
8. rebasing loop should be outside calculation loop, this means do rebasing first, then do calculation loop
9. string variable could not be used as an intermediate variable in calculation loop — use an extended formula, assign the string variable to a scope of this extended formula
10. when referring the scope of a t-dependent extended formula, it should be written as “EX_FORM(t).FIELD”

13 Building a model in Prophet from scratch :prophet:

13.1 must have a t-dependent variable

13.2 mandatory varible

- PROJ_TERM_Y
- DUR_M (this variable should be not dependent on t)

14 Combining rebasing loops with calculation loops :prophet:

Rebasing loops should be within each calculation loops, I guess C code would be like below:

```
int DUR_M;
int PROJ_TERM_Y;
int NUM_CLCLOOPS;
int NEW_REBASE_M(int REBASE_MTH);
for (int CALC_LOOP=1; CALC_LOOP<=NUM_CLCLOOPS; CALC_LOOP++) {
    for (int REBASE_MTH=DUR_M; REBASE_MTH!=PROJ_TERM_Y*12; NEW_REBASE_M(REBASE_MTH)) {
        // calculating variables
    }
}
```

Each calculation loop would stand for each projection on a certain assumption, however, each run I think should only contain one main assumption. So why are calculation loops needed? — “projection(s) in projection” problem. The former “projection(s)” may refer to the required projection(s) to calculate reserves on various assumption(s) different from the experience assumption, which is the assumption on which the latter projection is performed.

Each rebasing loop in a certain calculation loop would be used to calculate a certain reserve (depending on the calculation loop) for a future period (indicated by the “REBASE_MTH” variable in the rebasing loop). So when the experience assumption is used in a certain calculation loop, rebasing loops are not required. Because no reserve on a different assumption is needed.

The following is my model design: For each run, the following items should be specified:

- main assumption
- a loop info table which indicates the reserves need to be calculated and the rebasing setting

The first calculation loop is always reserved for the main assumption, used to project the experience status variables (account value, number of inforce policies . . .). The rebasing setting for the first calculation loop would always be set as “NA” which indicate no rebasing loops are needed.

Then, each calculation loop is iterated and with the related rebasing setting reserves on different assumption are projected.

Finally, if the experience cash flows are still needed, one more projection on the experience are needed produce the required cash flows. (This is flaw in this design, because the experienc projection is performed twice.)

15 Git basics :git:

status:

- untracked
- staged
- commited

16 Gray's talk on Prophet :prophet:

- code and table
- factor out commonalities (parameterised modelling) by code, differences as parameters in table
- benefit array
- decompostion of t-dependent benefit
- introduce assumption band, assign band to product or channel ..., assign assumption to band
- 'PPP_BAND"? Readability?
- parameterisation of table names
- column vector stands for vector of policies of various status; row vector stands for probabilities of insured evenets; matrix stands for sum assured payable conditional on status and insured events; matrix stands for transition matrix
- calculate each possible TAR choices? By an array variable
- product level dimension, override the dimension defined at workspace level
- "LIMIT_C_LPS" variable?

17 New Feature in Prophet 9 :Prophet:

module contains a set of user-defined function which encapsulates certain calculation logic

flexible tables when setting up a flexible table, meta-info about how to read it is also given, then no need for any codes to control how to read table

Split by Model Point Batches not only by Product

Books a sub-liabrary that allow a more flexible and neat way to organise code

PP API Enhancements embedded calculation logic built in Prophet with you other application, e.g. Excel Workbook . . .

18 EV review - Project Jade :EV:work:

- 84% on a TEV basis (or 78% on an MCEV basis) of PV of DE relates to cashflows in the first 10 years of projection as at 31 Dec 2015
- 47m of profits was expected to monetise from ViF
- this pattern gives the insurer the flexibility
- primarily fee charges and relatively small expenses for unit-linked policies
- 10% reduction in lapses would lead to a 11.2m, increase in MCEV, 2% (downside impact?)
- 10% reduction in expense would lead to 26.6m, increase in MCEV, 5%
- Value of Sterling, exchange rate
- new business margin decreases
- postive movement comes from prudence in the technical provision, exchange rate, offset by the decreasing of interest rate and non-recurring development cost
- negative non-linked reserve is the sterling reserve, to the extent of the cash value

- Hong Kong regulatory regime applies
- Current regulatory environment
- currency hedge only for non-linked part
- deferred acq. cost . . . would be removed from NAW
- tax currently are free
- calculation of CNHR and Risk Margin under Solvency II
- FCoC is zero? No tax but also no investment income, intangible asset should be removed from NAW.
- economic variance as a main source for the postive movement for MCEV
- two main busines: Regular Premium business and Portfolio Bonds
- products within the same business seem has varying level of profits

19 Angela :work:

- key questions instead of report
- actuarial, product
- ţÿžčž¡
- żń
- ÿ¿specingč£
-
- 5ň8?
- cash flow cross multiplication
-
- ÿż£ğ
- branchčňţńšżěű
- Bond ¿ Single Premium

- 2015 čň15čňSAR16čň
- ô
- Solvency regimečň¿čňGPVčňstatic factor
- Solvency II čňavailable capital ž threhold űijś
- ĺűĺšű
- žćśĺś NPVčňGPV
- Solvency II ÿ Testing
- Valuation Solvency I basis TEV
- Observation, summary, question
- Valuation report: assumption is hiden
    - business overview
        - product landscapečňtargetţč£Business Plan
        - channel
        - sales
    - solvency status, summary of reserves: components, mannual adjustment ôč£
        - analysis of reserve movement
        - Bridging current reserve and Solvency II reserve
        - Avaiable capital, any conversion between solvency regime
    - EV
- Risk: FCR report
- Other management accounts
- Reinsurance