Formalizing Actuarial Mathematics

Yosuke ITO

Sompo Himawari Life Insurance Inc.

November 21st, 2021

Disclaimer

- The contents presented here are solely the speaker's opinions and do not reflect the views of Company.
- There are some inaccuracies in explaining actuarial mathematics due to the priority on intuitive understanding.

Contents

Self Introduction

Professional Experience

- Sompo Himawari Life Insurance Inc., December 2020 – Present.
  - Aggregates the business results of life insurance products.
- Meiji Yasuda Life Insurance Company, April 2014 – November 2020.
  - Revised the reinsurance contracts.
  - Determined the prices of life insurance products.
  - Attended the approval negotiations with Financial Services Agency.
  - Qualified as an actuary (Fellow of the Institute of Actuaries of Japan).
  - Aggregated the business results of group life insurance.
  - Calculated retirement benefit obligations of client enterprises.
  - Validated the financial soundness of Employees' Pension Plans.

Education

- Nagoya University
  - Master of Mathematical Sciences, March 2014.
- The University of Tokyo
  - Bachelor of Science, Mathematics Course, March 2012.

Contents

What Is Actuarial Mathematics?

- Actuarial mathematics is a "mathematical, statistical, financial and economic theory to solve real business problems, typically involving risk, uncertainty and the financial impact of undesirable events". [3]
- It is related to
  - calculus,
  - probability theory,
  - financial theory.
- The traditional actuarial roles are considered as
  - determining the prices of insurance products,
  - estimating the liability of a company associating with the insurance contracts.
- Recently, the risk management skill of actuaries is required in a wider range of businesses.

Formalizing Actuarial Mathematics

- The most traditional area of actuarial mathematics is life insurance mathematics.
- It deals with
  - how to determine the prices of life insurance products,
  - the estimation of loss reserves – “the amount an insurer would need to pay for future claims on insurance policies it underwrites”. [1]
- I formalized the basic part of life insurance mathematics in Coq.
  - GitHub: `Yosuke-Ito-345/Actuary` `https://github.com/Yosuke-Ito-345/Actuary`
  - How to install: `opam install coq-actuary` (thanks to Karl Palmskog)
- I delivered a presentation of this work in the annual conference of the Institute of Actuaries of Japan in November 5th, 2021. (The proceeding will be published in 2022.)

Contents

Pricing a Pure Endowment I

Pure Endowment: "a type of life insurance policy in which an insurance company agrees to pay the insured a certain amount of money if the insured is still alive at the end of a specific time period" [1]

Assumption

- *amount insured: \$10000*
- *entry age: 30 years old*
- *policy period: 10 years*
- *probability that the insured person will survive for 10 years: 90%*
- *annual interest rate: 2%*

Question

How do you determine the price of this insurance?

Pricing a Pure Endowment II

Assumption

- *amount insured: \$10000*
- *entry age: 30 years old*
- *policy period: 10 years*
- *probability that the insured person will survive for 10 years: 90%*
- *annual interest rate: 2%*

- The expected payment after 10 years is

$$\$10000 \times 90\% = \$9000.$$

Question

Do you really need \$9000 now?

Pricing a Pure Endowment III

Assumption

- *amount insured: \$10000*
- *entry age: 30 years old*
- *policy period: 10 years*
- *probability that the insured person will survive for 10 years: 90%*
- *annual interest rate: 2%*

- If the insurance company earns 2% investment yield annually, the required amount for this insurance can be discounted:

$$\frac{\$9000}{(1+2\%)^{10}} \approx \$7383.$$

Pricing a Pure Endowment IV

Definition

The present value of a pure endowment on a person aged x payable at the end of n years is written as $A_{x:\frac{1}{n|}}$ per unit insurance amount:

$$A_{x:\frac{1}{n|}} := {}_np_x v^n.$$

Here,

- ${}_np_x$ is the probability that the insured person aged x will survive for n years,
- $v := 1/(1+i)$, where i is the interest rate.

- In the example above, $A_{x:\frac{1}{n|}} \approx 0.7383$.

Pricing a Whole Life Annuity I

Whole Life Annuity: “a financial product sold by insurance companies; it gives out monthly, quarterly, semi-annual, or annual payments to a person for as long as they live, beginning at a stated age” [2]

Assumption

- *amount insured: \$1000*
- *frequency of payment: yearly*
- *entry age: 60 years old*
- *annual mortality rates:*

$$\begin{cases} q_x = 0.1 & \text{if } x < 99 \\ q_x = 1 & \text{if } x = 99 \end{cases}$$

- *annual interest rate: 2%*

- The present value of the expected payment after k years is

$$\$1000 \times A_{60:\overset{\,1}{\overline{k|}}} = \$1000 \times {}_k p_{60} v^k.$$

Pricing a Whole Life Annuity II

- The present value of this annuity is

$$
\begin{aligned}
\sum_{k=0}^{39} \$1000 \times A_{60:\frac{1}{k|}} &= \sum_{k=0}^{39} \$1000 \times {}_k p_{60} v^k \\
&= \sum_{k=0}^{39} \$1000 \times \left\{ \prod_{j=0}^{k-1} (1 - q_{60+j}) \right\} v^k \\
&= \sum_{k=0}^{39} \$1000 \times 0.9^k \cdot \left(\frac{1}{1 + 2\%} \right)^k \\
&= \$1000 \times \frac{1 - (0.9/1.02)^{40}}{1 - 0.9/1.02} \\
&\approx \$8443.1.
\end{aligned}
$$

Pricing a Whole Life Annuity III

Definition

The present value of a life annuity on a person aged x payable at the beginning of each year so long as the person survives for up to a total of n years is written as $\ddot{a}_{x:\overline{n}|}$ per unit annual payment:

$$\ddot{a}_{x:\overline{n}|} := \sum_{k=0}^{n-1} {}_k p_x v^k.$$

When the annuity is whole-life ($n = \infty$), $\ddot{a}_{x:\overline{n}|}$ is also written as $\ddot{a}_x$.

- In the example above, $\ddot{a}_{60} \approx 8.4431$.

Pricing a Term Life Insurance I

Term Life Insurance: "a type of life insurance that guarantees payment of a stated death benefit if the covered person dies during a specified term" [2]

Assumption

- *amount insured: \$10000*
- *entry age: 30 years old*
- *policy period: 10 years*
- *annual mortality rate: 0.01*
- *annual interest rate: 2%*

- The present value of the expected payment after k years is

$$\$10000 \times {}_{k-1}p_{30} \cdot q_{30+(k-1)} \cdot v^k.$$

Here, the death benefit is supposed to be paid at the end of the year of death.

Pricing a Term Life Insurance II

- The present value of this insurance is

$$
\begin{aligned}
&\sum_{k=1}^{10} \$10000 \times {}_{k-1}p_{30} \cdot q_{30+(k-1)} \cdot v^k \\
&= \sum_{k=1}^{10} \$10000 \times 0.99^{k-1} \cdot 0.01 \cdot \left(\frac{1}{1+2\%}\right)^k \\
&= \$10000 \times 0.01 \cdot \frac{1-(0.99/1.02)^{10}}{1-0.99/1.02} \cdot \frac{1}{1.02} \\
&\approx \$860.
\end{aligned}
$$

Pricing a Term Life Insurance III

Definition

The present value of a term life insurance on a person aged x payable at the end of the year of death within n years is written as $A^1_{x:\overline{n}|}$ per unit insurance amount:

$$A^1_{x:\overline{n}|} := \sum_{k=1}^{n} {}_{k-1}p_x \cdot q_{x+(k-1)} \cdot v^k.$$

- In the example above, $A^1_{x:\overline{n}|} \approx 0.0860$.

Actuarial Notations and Formulas

- These kind of symbols are called "actuarial notations" and commonly used in various countries.
  - INTERNATIONAL ACTUARIAL NOTATION
    `https://www.casact.org/sites/default/files/database/proceed_proceed49_49123.pdf`
- In life insurance mathematics, the relations between the actuarial symbols are well examined:

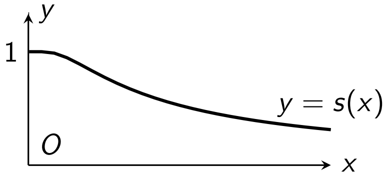
$$A^{1}_{x:\overline{n}|} = 1 - iv\,\ddot{a}_{x:\overline{n}|} - A_{x:\overset{1}{\overline{n}|}}.$$

- Actuaries use these symbols efficiently to calculate prices of products, reserves of the company, etc.

Survival Function

Definition

Let T be a random lifetime variable, and define $s(x) := P(T > x)$ for an age x. The function s is called the "survival distributive function".



Example

$$
\begin{aligned}
{}_np_x &= P(T > x + n \mid T > x) \\
q_x &= P(T \leq x + 1 \mid T > x) \\
\mathring{e}_0 &:= E(T) = \int_0^\infty s(x)\,dx \quad \text{(average life span)}
\end{aligned}
$$

Life Table

- In practice, actuaries use "life tables" to calculate probabilities.

x	l_x	d_x
0	100000	238
1	99762	143
2	99619	120
$\vdots$	$\vdots$	$\vdots$

Example

$$
\begin{aligned}
{}_n p_x &= l_{x+n}/l_x \\
q_x &= d_x/l_x \\
\mathring{e}_0 \approx e_0 &:= \sum_{x=1}^{\infty} l_x/l_0
\end{aligned}
$$

Contents

Overview of the Actuary Package

coq-actuary

- GitHub: `Yosuke-Ito-345/Actuary`
  `https://github.com/Yosuke-Ito-345/Actuary`
- How to install: `opam install coq-actuary`

Release of Version 2.1 (November 1st, 2021)

filenames	SLOC	contents
`Basics.v`	1000	basic lemmas of mathematics
`Interest.v`	794	present and future values of fixed annuities
`LifeTable.v`	827	life tables and their properties
`Premium.v`	1863	life annuities, insurances, and their prices
`Reserve.v`	727	reserves of life insurances
`all_Actuary.v`	5	all the libraries above
`Examples.v`	187	some applications of this package

Formalizing Life Table

```
(* life table *)
Record life : Type := Life {
  l_fun :> R -> R;
  l_0_pos : 0 < l_fun 0;
  l_neg_nil : forall u:R, u <= 0 -> l_fun u = l_fun 0;
  l_infty_0 : is_lim l_fun p_infty 0;
  l_decr : decreasing l_fun
}.
```

```
Definition ages_dead (l:life) : Ensemble nat := fun x:nat => \l[l]_x = 0.
Definition l_finite (l:life) := exists x:nat, (ages_dead l x).
```

```
Section DifferentiableLifeTable.
(* Suppose l is continuously differentiable. *)
Hypothesis l_C1 : forall u:R, ex_derive l u /\ continuous (Derive l) u.
```

Implementing Actuarial Notations

```
(* present value of a pure endowment life insurance *)
Definition ins_pure_endow_life (i:R) (l:life) (u:R) (n:R) :=
  \v[i]^n * \p[l]_{n&u}.
Notation "\A[ i , l ]_{ u : n `1}" :=
  (ins_pure_endow_life i l u n) (at level 9, u at level 9).
```

```
Section Premium.

Variable i:R.
Hypothesis i_pos : 0 < i.
Variable l:life.
Hypothesis l_fin : (l_finite l).

Notation "\v" := (\v[i]) (at level 9).
Notation "\p_{ t & u }" := (\p[l]_{t&u}) (at level 9).
Notation "\A_{ u : n `1}" := (\A[i,l]_{u:n`1}) (at level 9, u at level 9).
```

- MathComp and Coquelicot are required.
- Classical logic is assumed.
- The axiom of choice is partly used.

Application of the Actuary Package

Theorem

If the annual interest rates i and i' satisfies $i \leq i'$, then we have $\ddot{a}_{x:\overline{n}|}(i) \geq \ddot{a}_{x:\overline{n}|}(i')$.

```
Lemma ann_due_decr_i : forall (i i' : R) (x n : nat),
  0 < i -> 0 < i' -> x < \ω -> i <= i' -> \a''[i']_{x:n} <= \a''[i]_{x:n}.
```

```
Proof.
  move => i i' x n Hipos Hi'pos Hx Hleii'.
  have Hvpos : 0 < \v[i] by apply /v_pos /Hipos.
  have Hv'pos : 0 < \v[i'] by apply /v_pos /Hi'pos.
  rewrite !ann_due_annual.
  apply Rsum_le_compat => k /andP; case => /leP Hmk /ltP Hkn.
  apply Rmult_le_compat_r; [by apply (p_nonneg _ l_fin) |].
  case: (zerop k) => [Hk0 | Hkpos].
  - rewrite Hk0 !Rpower_O //; lra.
  - case: (Rle_lt_or_eq_dec i i') => // [Hlt | Heq].
    + rewrite /Rpower.
      apply /Rlt_le /exp_increasing.
      apply Rmult_lt_compat_l; [rewrite (_ : 0 = INR 0%N) //; apply lt_INR => // |].
      apply ln_increasing => //.
      rewrite /v_pres.
      apply Rinv_1_lt_contravar; lra.
    + rewrite Heq; lra.
Qed.
```

Future Applications I

Error Detection of Actuarial Documents

- Tasks
  - formalizing the remaining part of life insurance mathematics
  - generalizing lemmas in the Actuary package
  - automation of reasoning
- Problems
  - the too strong assumption of differentiability in Coquelicot (no singular point permitted)
  - insufficient formalization of the improper integral

Future Applications II

Verification of Programs Used in Actuarial Business

- What We Can Do Now
  - extracting functions defined in Coq to programs written in OCaml, Haskell and Scheme
  - verifying the existing source codes by
    1. writing a model of the program, and
    2. formally proving that the model satisfies the required properties
  - mechanically checking the exact source C programs by Frama-C
  - avoiding miscompilation of C programs by CompCert
  - ...

Future Applications III

Verification of Programs Used in Actuarial Business

- Tasks
  - developing a mechanically checking tools like Frama-C for actuarial softwares
  - developing a formally verified compiler like CompCert for actuarial softwares
- Problems
  - lack of experts in formal verification well-versed in actuarial mathematics
  - limited users compared to common programming languages
  - expensive cost for development

Acknowledgments

- I thank Reynald Affeldt for giving me a lot of information about the current researches on the formalization of mathematics.
- I also thank Karl Palmskog for adding the Actuary package to the Coq OPAM repository and arranging for easy installation.

References

[1] Insuranceopedia.
Dictionary.
https://www.insuranceopedia.com/dictionary, 2021.

[2] Investopedia.
Dictionary.
https://www.investopedia.com/financial-term-dictionary-4769738, 2021.

[3] University of Leeds.
Actuarial mathematics BSc.
https://courses.leeds.ac.uk/f702/actuarial-mathematics-bsc, 2021.

Discussion

1. Should I avoid the classical logic and the axiom of choice?
2. What is the best way to apply proof assistants to actuarial businesses?
3. How widespread are proof assistants among programmers?
4. How do you choose the appropriate proof assistant?