Equity-Based Insurance Guarantees Conference

Nov. 11-12, 2019

Chicago, IL

Speeding up Actuarial Seriatim Calculations

Andrey Marchenko

Speeding up Actuarial Seriatim Calculations

Andrey Marchenko, Director, Moody's Analytics

11 Nov 2019, Session 3A

Agenda

1. Large Calculations with Seriatim Records
2. Closed Form Approximation
3. Clustering
4. Per Policy Generate
5. Control Variates
6. Low Discrepancy Sequences
7. Semi Monte Carlo
8. Combinations of Different Methods

1

Large Calculations with Seriatim Records

Large Calculations with Seriatim Records

Problem Setting

- Need to run extremely many lengthy but similar calculations to obtain result
  - Example: calculating many policies projections for multiple scenarios each
- The number of individual projections
  - (Number of policies) * (Number of random scenarios) * (Number of outer loop scenarios) * (Number of pivot points)
- The last two factors are dictated by the purpose, but the former two can be reduced with some techniques at a price
- Omit brute power (hardware) approach – Grid, Cloud, GPU, …

Large Calculations with Seriatim Records

General Considerations

- Trade off between precision and calculations amount
- Find few numbers which really need expensive calculation
  - Choose methods
  - May be hard to implement due to system design
  - Try to combine methods
    - Often methods use the same redundancy and thus are less efficient than alone
    - Use methods exploiting different redundancies
- Use segmentation and different methods for different segments

Large Calculations with Seriatim Records

Basis for Improvement

- Reduction of calculations can usually be done by using:
  - Redundancy of results

    some calculations give close results – reuse it and reduce the total amount of calculations
  - A-priori information on the problem
    - Known distribution - reduce variance induced by Monte Carlo by choosing special representatives
    - Known approximate results - reduce Monte Carlo variance by using Control Variates
    - Known special cases – use segmentation

Large Calculations with Seriatim Records

Results Comparison

» How to compare methods?

– Some speed up, others reduce error/variance

– Hard to compare overheads

» Benchmark – Monte Carlo for all policies with the same set of scenarios

– Error measure is MSE – it takes into account both variance and bias

– For pure Monte Carlo coincides with variance

– Monte Carlo error is scalable by the number of scenarios (Variance ~1/N)

– Overhead is ignored

– Gain in performance – relative reduction in scenarios number for the same MSE

2 Closed Form Approximation

Closed Form Approximation

The Fastest Way

- Can be performed for
  - Comparatively simple models (Black Scholes)
  - Few critical numbers (often deep inside the results calculation)
- Extremely efficient when properly done
- May require massive preliminary research/calculations, but done once only
  - Is specific to settings, so may need recalibrating
- Examples of methods for problems without theoretical closed form solution
  - Least Squares Monte Carlo
  - NN approximations

3

Clustering

Clustering

Main Idea

- Exploits the data redundancy created by the presence of multiple very similar objects giving very close results proportional to some input parameter
- Split the whole set of objects into several non-intersecting subsets called clusters containing objects considered “similar” to one another
- Calculations are performed with a single representative per cluster and then scaled to the size of the cluster
- Reduces calculations amount at the expense of an error added
- Not to confuse with segmentation which does not reduce calculations, but splits them into parallel streams. Different approaches to speed them can be applied.

Clustering

Some Considerations

- Assumes overhead, sometimes considerable
  - Balance number of clusters, overhead, and error
- Challenge to define Location Variables (often assume close parameters yield close results)
  - Continuity shifts similarity from outputs to inputs
  - Can use approximate outputs if cheap to find
- Dimension curse - unavoidable consequence of high dimension geometry
  - Reject non-important inputs
  - Adjust metrics to reduce impact of less important inputs
- Hard to predict and evaluate an error

Clustering

Workflow

1. Define Location Variables, that is map individual objects into some (usually Euclidean) space.
2. Define distance in this space.
3. Perform Clustering (find clusters).
4. Find cluster's Representatives and their weights.
5. Calculate weighted sum, running representatives only.

Policies Clustering

Special Considerations

- Requires preservation of the mean
- Challenge to define Location Variables
  - Many (almost) repeated fields
  - Highly correlated fields
  - Categorical fields may lead to high dimension
  - Fields combinations (year, month)
- Approach: exploit actuarial judgement and product knowledge
  - Reject non-important inputs
  - Adjust metrics to reduce impact of less important inputs
  - Use segmentation

Scenario Clustering

Special Considerations

- » Needs to preserve not means only, but the distribution
  - Different highly non-linear functions to be estimated
- » Dimension curse – almost always, more for ready scenarios than for generated
- » Location variables may be highly non-linear depending on the problem
  - Hard to find
  - Scenarios are often used to calculate very different instrument’s values
- » Can be applied on different levels (random drivers vs. ready scenarios)
- » When applied to drivers, can be made in advance (compare to Low Discrepancy Sequences)

Scenario Clustering

Approaches

- Dimension curse
  - Use PCA to define Location Variables
  - Choose non-linear Location Variables replicating target values
    - Discount factors vs. discount rates
  - Apply AI methods (“Bottleneck architecture”)
- Correct mean and covariance matrix
  - Use weights
  - Create new representatives – exploits linearity
    - Pay attention to special conditions (positivity etc.)

4

Per Policy Generate

Per Policy Generate

Main Idea

- Applicable for calculation of the sum of expectations over random scenarios for a set of policies
- Cannot be used for some problems (e.g. CTE evaluation)
- Compare result's variances for two strategies:
  - Standard Monte Carlo: generate a set of K scenarios and calculate results for each policy using this single set.
  - Per Policy Generate: generate a set of K scenarios <u>for each policy independently</u> of others and calculate results for it using this individual set.
- The second strategy wins if covariance between policy results is high for many different policies – redundancy used

Per Policy Generate

Other Considerations

- Exploits the redundancy close (but not the same) to that used by Clustering – policies similarity leading to high correlations with respect to random scenario
- Maximum relative gain in variance ~(Number of policies)$^{-1}$ when policies are perfectly correlated
- Relative gain does not depend on the Number of scenarios
- Requires generating more scenarios – overhead to be estimated
- Can be modified to set the number of scenarios depending on policy results variance estimated on the fly – extra improvement
- Hard to predict an error, but is possible to evaluate it a-posteriori

5 Control Variates

Control Variates

Variance Reduction Technique

- Exploits high correlation between policy and approximation results which yields correlation between policy results
- Needs an a-priory knowledge of result’s behavior to build an Approximation
  - Calculates for each scenario
    Value (scenario) + *Const* ·(*Approximation mean* – *Approximation Value (scenario)*)
  - Needs fewer scenarios for the same variance
  - Capitalizes on a cheap calculation of *Approximation mean*
- Allows for technical modification by weighting scenarios (Manistre)
- Allows for an error evaluation based on results

6

Low Discrepancy Sequences

Low Discrepancy Sequence

Variance Reduction Technique

- Not a random in any sense, but can replace it for some purposes
- Based on special algorithms (van der Corput, Halton, Sobol,…) suppressing density fluctuations
- Has theoretical background predicting an error ($\sim(\ln N)^{dim}/N$)
- “Uniformity” of this sequence is asymptotically (in N) better, than that for a random sequence
- An error is less compared to a random sequence: $(\ln N)^{dim}/N \ll 1/N^{1/2}$ $(N\to\infty)$
- “Uniformity” deteriorates as dimension (dim) grows; there are ways to deal with it
- Other distributions are produced from the uniform

7

Semi Monte Carlo

Semi Monte Carlo

New Variance Reduction Technique

- A means to mitigate the dimension curse
- Combines numeric integration for few coordinates with Monte Carlo for others
- Provides variance reduction at the expense of some bias introduced by numeric integration
- Requires a-priory knowledge of the model/distribution to be efficient
- Requires individual design for different models/distributions
- Hard to predict/evaluate an error because of the bias

8

Combinations

Combinations of Different Methods

Overview

- Combining different methods rarely can sum gains, but yet improves results
- Is working better if methods combined use different data redundancies or a-priory knowledge
- Harder to implement and optimize parameters
- Harder to predict an error due to factors interference
- Used not as often, hence less experience
- Not all combinations had been tested
- Requires larger overhead

Clustering and Per Policy Generate

- Combines calculations reduction (by clustering) with variance reduction (by per policy generation)
- Based on the same type of data redundancy – policies similarity, but different aspects of it
  - Clustering – mean proportionality to some common parameter
  - Per Policy Generate – high correlations for different policies
- Displays better results than any of its components individually

Clustering and Variance Reduction Technique

Low Discrepancy Sequence, Control Variates, Semi Monte Carlo

- Use policies clustering
- Apply some Variance Reduction Technique to cluster's representatives when running Monte Carlo for them
- Use different data redundancy, so easy to combine
- Can be combined with Per Policy Generate (for representatives)
- Cluster representatives behave differently, hence
  - Hard to predict errors
  - May require significant overhead to adjust method for each representative

Clustering and AI Approximation

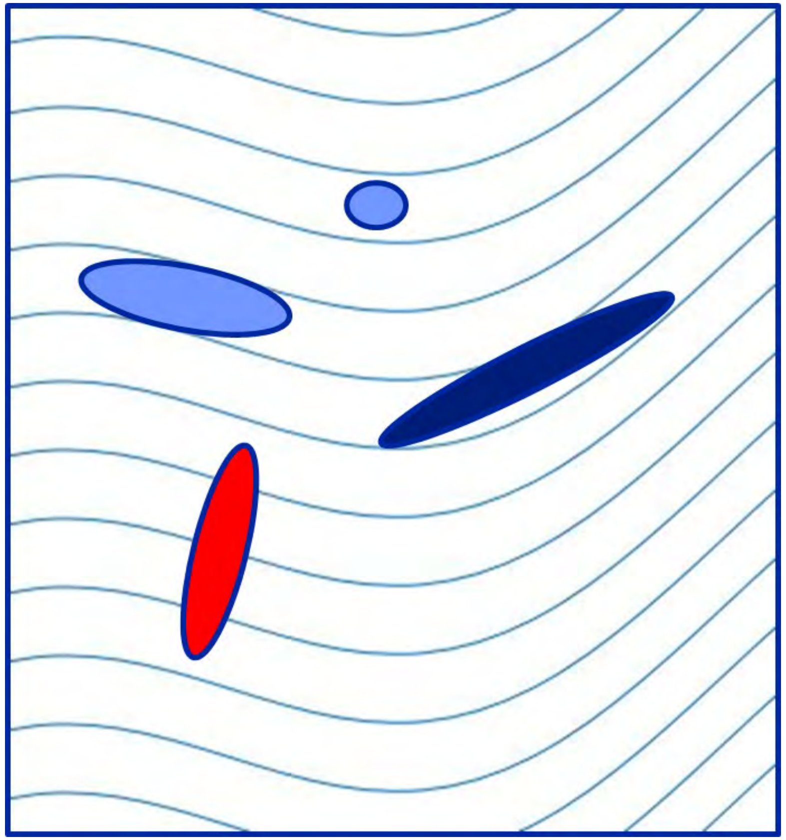
Using Neural Network to Generate Location Variables

- Potentially multiple designs, we consider one only
- Automates Location Variables generation based on NN approximation of results
- Can reduce Location Variables dimension by using “bottleneck” NN architecture
- NN does not use the same redundancy as Clustering
- Requires a proper balance of the training set size and number of clusters
- Very flexible

Clustering and AI Approximation

Main Idea

- We are summing $\vec{\mathrm{Y}} = \vec{\mathrm{F}}(\vec{x})$ for many $\{\overrightarrow{x_k}\}$
- Ideally for a cluster C
$$\vec{x}_1, \vec{x}_2 \in C \Rightarrow \vec{\mathrm{F}}(\vec{x}_1) \approx \vec{\mathrm{F}}(\vec{x}_2)$$
then we can combine them and calculate a single value for $\vec{\mathrm{Y}}$
- This is the case if $\vec{\mathrm{F}}(\vec{\mathrm{x}})$ is smooth and clusters are small.
- A cluster does not need to be small – the results $\vec{\mathrm{F}}(\vec{x})$ should be close.
- It does not depend on the distribution of $\vec{x}$, but on $\vec{\mathrm{F}}(\vec{x})$ only.
- Approximate $\vec{\mathrm{F}}(\vec{\mathrm{x}})$ by NN and use its output as Location Variables



Thank you!

Q&A

Moody's Analytics | AXIS™

Better
Faster
Decisions

Andrey Marchenko
5001 Yonge street, Toronto, ON, M2N 6P6, Canada
Andrey.Marchenko@moodys.com
(1) 416 250-3432