ROOT: A data mining tool from CERN
What can actuaries do with it?

Ravi Kumar, Senior Manager, Deloitte Consulting LLP
Lucas Lau, Senior Consultant, Deloitte Consulting LLP

Southern California Casualty Actuarial
Club Fall Meeting
Pasadena, California
December 3, 2010

Contents
• Background
• ROOT
• Case Study
• Conclusion

Background
• P&C insurance companies are increasingly becoming Data Driven
organizations
  • To understand their customers better
  • To improve operational efficiencies
  • To stay profitable
  • To stay ahead of their competitors

• So, data volumes are growing at an annual rate of 60% or even more.
Actuaries are finding it increasingly difficult to manage this data
explosion that is happening around them

• ROOT, a data mining tool from CERN, offers a scalable solution that
can help actuaries deal with large volumes of data

Demand for internal data
Actuaries are using more detailed internal data in their day to day work

Demand for external data
Actuaries are constantly looking for new business insights that can be derived from external data
Data, data everywhere

- Data needs are thus transforming everyday analytical work
- Actuaries are increasingly forced to take more responsibilities in how the data is prepared for their analysis
- Commercially available data management tools do not match their hype
- ROOT can provide a scalable solution to the actuaries in taking control of their data analytic needs

What is ROOT?

- Root is an efficient Data Storage framework specifically designed to handle "Big Data"
- Root is a Data Visualization Tool with extensive data visualization capabilities
- Root is a Modeling Tool with extensive set of data fitting, modeling and analysis methods including data simulation
- Root is a complete C++ programming environment
- Root comes with full memory management, networking, parallel processing capabilities for increased scalability

Root as a Data Storage Framework

- Root stores data in a compressed columnar structure
- Root can store complex objects, structures, arrays, images, in addition to simple data types
- Design is optimized for fast analysis capabilities
- For example:
  - In this example, Root takes only 18% space compared to the corresponding ASCII file

Root as a Data Visualization Tool...

- Root provides convenient graphic tools to visualize and explore even very large datasets
- Histograms 1-, 2- and 3-Dimensional
  - With interactive capabilities
  - With Dynamic re-binning capabilities
- Curve Fitting
- Graphs
- Pie Charts
- QQ Plots
- Parallel Coordinates

Root as a Modeling Tool

- Multiple Regression
- Maximum Likelihood Fitting
  - General purpose function minimizers
  - Can fit any user defined function to the data
- Neural Networks
- Decision and Regression Trees
- Boosting and Bagging
- Principal Component Analysis
- Rules based predictive learning
- Support vector machines
- Several other Classification/Regression Algorithms
More about Root …

- Root is available as a free open source software from http://root.cern.ch/drupal/
- Root is supported on many operating systems like Windows, Unix, Linux, Mac OS etc.
- Root is used in many industries
  - High Energy Physics community
  - Wall Street trading companies (Merrill Lynch, Renaissance Corp)
  - Flight planning systems (MITRE)
  - Defense (USAF, DoD)
  - Oil, geology (Halliburton)
  - Medical Imaging (Philips Medical)

Case Study: Price Optimization

What is Price Optimization?

Price Optimization is the process of calculating optimal set of prices taking into consideration demand elasticity, customer preferences and cost of delivering products or services.

- The definition of optimal price depends on business goals.
- Examples of business goals are
  - Maximize revenue
  - Maximize profit
  - Maximize customer retention
  - Maximize growth
  - Minimize least-squared errors

Case Study: Price Optimization

What are you trying to optimize?

1. Define the Objective Function
   \[ \text{Profit} = \sum \text{Quantity} \times (\text{Price} - \text{Cost}) \]
2. Determine the Constraints
   - Floor/ceiling prices
   - Inter-product relationship rules (equal, greater than) based on attributes
3. Decide on strategy, trade-offs
   Revenue versus profit versus market share versus ...
4. Use Calculus methods (Lagrange multipliers, Linear Programming, etc) to solve for price for multiple strategic scenarios

Putting Optimization Into Practice

How do we optimize prices?

1. Profitability function formalization:
   Most institutions can accurately measure performance from an operational standpoint. However, they cannot accurately estimate the profitability of an individual transaction at the time of origination. This requires robust pro forma models that estimate expected income, costs, and risks over the lifetime of a transaction.
   Accurate transaction-level estimation of profitability is a pre-requisite for a successful pricing initiative. When implementing optimization, a single measure must be selected at a given point in time.

2. Incorporation of constraints to the benefit function:
   Once benefit function is estimated, restrictions are incorporated (based on internal policies, standard requirements, etc.) that may condition pricing policy.

3. Benefit optimization:
   The ‘constrained’ objective function (e.g. profit, revenue, attrition rate, etc.) is optimized through simulation of multiple scenarios. Generating profit functions for a segment, market, product, or a combination thereof.
Data Exploratory in Price Elasticity

- Definition of Elasticity: the ratio of percentage change in quantity sold to percentage change in price

\[ Elasticity = \frac{\frac{\partial Q_{Sold}}{\partial P}}{P} \]

- In order to have an accurate estimate of elasticity, we need to examine the outliers of both numerator and denominator

Case Study: Price Optimization

1. Let’s look at the traditional way to do it in SAS: `proc univariate`.
2. How do they distribute graphically?

Here is the graphical representation of distribution

- X-Axis: % change in Quantity Sold
- Y-Axis: Frequency
- X-Axis: % change in price
- Y-Axis: Frequency

The absolute count on Y-axis does not tell us about the frequency of extreme cases because they are too small relative to the majority. We need to do some transformation to Y-axis: SAS? or ROOT?

After the log transformation, the frequency of extreme values pops

Just a right click on your mouse, the Y-axis has been transformed to log scale. This save the hassle of programming the log transformation and running the program in SAS!

Modeling in Root

- Use TMinuitMinimizer to minimize the negative of the objective function in ROOT
- Root is a complete C++ programming environment, so that you can customize your objective function and make your optimization engine as proprietary as possible
- Root is an efficient tool in running optimization. We compared our internal optimization engines between R and Root. Root outperformed R significantly in terms of running time.
### Conclusion

- Root is specifically designed to handle, store, and analyze large amounts of data very efficiently.
- Root comes with many:
  - Statistical tools
  - Data mining tools
  - Data visualization tools
  - Connectivity tools
- Root is suitable for any environment requiring serious and scalable data analysis solution.

### Bibliography

- [http://root.cern.ch/drupal/](http://root.cern.ch/drupal/) - Main ROOT Website