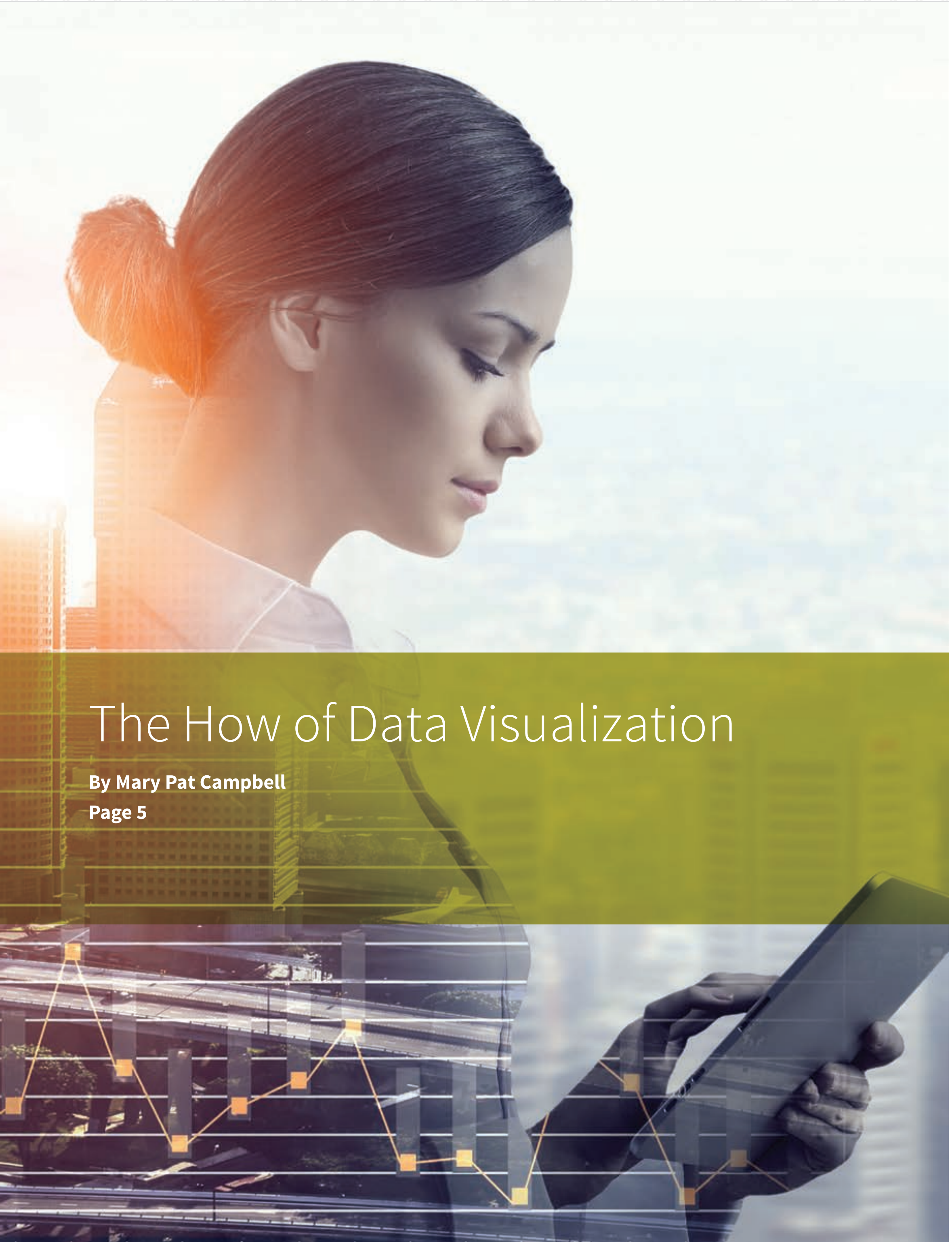
SOCIETY OF ACTUARIES
TECHNOLOGY SECTION

CompAct

ISSUE 60 • APRIL 2018



CompAct

Issue 60 • April 2018

Published twice a year by the Technology Section of the Society of Actuaries.

475 N. Martingale Road, Suite 600
Schaumburg, Ill 60173-2226
Phone: 847.706.3500 Fax: 847.706.3599
www.soa.org

This newsletter is free to section members. Current issues are available on the SOA website (*www.soa.org*).

To join the section, SOA members and non-members can locate a membership form on the Technology Section webpage at *http://www.soa.org/sections/technology/technology-landing/*.

Publication Schedule

Publication Month: October 2018
Articles Due: July 12, 2018

Letter From the Editor

By Ravi Bhagat

I am honored to introduce the first edition of *CompAct* in 2018! You will start to see a natural evolution in our content that mirrors the objectives of the Technology Section, and as my first action as editor, I'd like to reflect on the overall goals of the newsletter. The newsletter's goal is to provide a basis for developing technology proficiency through educating our readers, and to promote technology discovery through the exploration of innovative and disruptive topics. Of course, both of these objectives are also served by casual introductions and guiding our readers to other outlets by which they can further explore the topics. To this end, our newsletter includes articles and content written by a wide range of individuals with diverse backgrounds. To broaden the perspective provided by the diverse group of authors, we did not limit ourselves to traditional actuarial topics (if such a thing exists). Ultimately, this enriches the value of the newsletter and increases its relevance. We leave it to the reader to draw upon the content, connect it to their personal and professional endeavors, and leverage it to carry them (and the industries they serve) into the future.

On a personal note, I would strongly encourage readers to submit topics of interest for future publication, articles for consideration and/or general feedback related to the newsletter. The quality, breadth and relevance of the content is largely dependent on our members and we can collectively redefine this newsletter to be an outlet where one can immerse themselves in technology related subject matter. For thoughts, questions and feedback, I can be reached at *ravibhagat@kpmg.com*.

In this issue of *CompAct*, we have six articles. One is a continuation of a recurring series, and we have five new contributions.

THE HOW OF DATA VISUALIZATION

In the fifth and final installment of a continuing series on data visualization, Mary Pat Campbell explores several interesting aspects within Microsoft Excel. The main vehicle in this instance to support the visualization techniques is Microsoft Excel, but as the author notes, there is analogous functionality within other platforms.

ACORD

To provide greater awareness and spark reader curiosity, we included an introductory article on ACORD, a global standards-setting body for the insurance industry. The cost-effective and high-quality flow of data and information across the insurance value chain is paramount in today's insurance industry. ACORD is one such non-profit, industry-owned organization that is encouraging and facilitating new standards.

SMALL COMPANY, MODERN DATA STRATEGY

Wholesale transformation initiatives are considered the norm for Fortune 500 and global insurers, but small- and medium-sized insurers are defining highly targeted initiatives. Ying Zhao and Win Georg contributed a practical article that outlines considerations for small- to medium-sized companies as they look to capitalize on technology innovation.

INTRODUCTION TO DISTRIBUTED COMPUTING

Introducing the theme of Big Data and related computational challenges, Jason Alteri delves into the topic of distributed computing. As demands on computational analysis and the need for faster turnaround times continue to increase, this article provides a great introduction into distributed computing technologies.

CLOUD COMPUTING

Continuing on the theme of computational challenges, Joe Long and Dan McCurley provide an insightful piece that details a three month machine learning exploration project and the considerations of cloud computing enablement. With increasing demands for high-quality data, the author notes that the cloud is an ecosystem of resources that can be leveraged to explore ideas and complete tasks that were once limited due to computational constraints.

VA VALUATION USING GPU

Lastly, as part of the Technology Section's recent essay writing contest, Huina Chen and Henry Bequet penned an exploratory article on GPU and Deep Learning advancements within the variable annuity valuation space. Using variable annuity valuation provides a great use-case that can be relatable to other functional areas that have demands for robust and timely financial analytics. ■

Ravi Bhagat, FSA, MAAA, is an actuarial director with KPMG, LLP. He can be reached at *ravibhagat@kpmg.com.*

Chairperson's Corner

By Mark Africa

Welcome to the April issue of *CompAct!* I am grateful and honored to serve as chair of the Technology Section for the current cycle. I would like to briefly review section recognition, the vision and mission of our section for 2018 and the value of feedback and volunteerism.

Firstly, a thank you to our outgoing section members for their contributions. In particular, I'd like to thank our outgoing chair Paul Ramirez for his leadership and direction on behalf of the section. Paul did a great job leading by example and most definitely leaves large shoes to be filled. I also welcome our new section members and friends of our section. I firmly believe that we can never have too many friends! We are grateful for the contributions of article authors, webcast and meeting presenters. I also want to thank Sean Hayward who will serve as vice chair and performed admirably leading our second annual face-to-face meeting earlier this year at the SOA headquarters, thank you Sean.

Secondly, we will highlight some mission themes that we have formulated this year. We are a relatively small section, but technology covers a vast array of ever-changing and emerging topics: artificial intelligence, cloud computing and standardized data models just to name a few. Our membership is talented and highly skilled in a variety of topics, but we recognize that we cannot be an authoritative resource on all technology related topics. Instead, our aim is to be a source actuaries can turn to when wanting to learn what topics they need to pursue further and where they can go to learn more about those topics. In some cases, we will be the provider of that information they can use to learn more, but again, we can't be experts in everything technology related, it is simply too broad a topic. We are continuing our work to enhance Apps for Actuaries within the SOA website and also plan to continue work with ongoing research regarding InsurTech; more to come on that topic. We will have a continued focus to include vendor/consultant feedback as well as carriers in Technology Section activities. Vendors play a huge role in our technology services and products services and their input remains vital to our mutual success.

Lastly, I want to discuss volunteerism. When Paula Hodges suggested that I participate in the Technology Section three years ago, I admit that I was somewhat skeptical. But, I viewed it as an opportunity to give back to a profession that has been so rewarding to me. We all have day jobs and personal lives, but I can tell from my experience, volunteering to serve on an SOA section council is very rewarding in ways that I had no anticipated. The opportunity to network and learn from and exposure to others within our industry is very interesting and dynamic, thank you Paula for calling on me. This is your section and your SOA. If we are doing something that you like, tell us, if we do something that you do not like, tell us that as well, we appreciate all of your feedback. If you really want to impact change and have a voice, run for a section council or volunteer as a presenter, author or friend of a section, we welcome all of you.

If you are interested in participating in the activities of our section, please take advantage of one of these options. For more information, please contact me or Jane Lesch, our SOA section specialist.

Thank you for your continued support! ■

Mark Africa, ASA, MAAA, is an IT actuary at AIG. He can be reached at *mark.africa@aig.com*.

The How of Data Visualization

By Mary Pat Campbell

This is a fifth and final part of a continuing series on data visualization (aka dataviz):

- The Why of Data Visualization—Questions to ask When Visualizing Numerical Information (March 2016)
- The Who of Data Visualization—Major figures and Books in Advocating Data Visualization Best Practices (May 2016)
- The Where of Data Visualization—Websites to Polish Your Data Visualization Game (December 2016)
- The What of Data Visualization—Software to Implement Data Visualization (October 2017)
- The How of Data Visualization—Specific Data Visualization Techniques to Consider in Actuarial Practice

(The when of data visualization being NOW, of course.)

For this article, I'm going to touch on a few data visualization techniques for you to consider using in your own actuarial work, that you may not be as familiar with. I have mentioned a few of these techniques in prior articles, and in this case, I will be supplying a spreadsheet with all these implemented in Excel. These all have analogues in Python and R, as well as other common dataviz systems.

A quick note on software uses: the important thing is to consider the specific technique, why you would want to use it, and what it is best in accomplishing. Some of these methods are easier to implement in speci fic software systems currently. However, software, even technical software, is moving towards perpetual beta (i.e., always in development), you'll find many useful techniques spreading more broadly. The more useful the technique, and the more widely used, the more likely you'll find it easily implementable in multiple places.

In the following examples, I will be using data from the Public Plans Database, specifically looking at funded ratios by state.[1] The spreadsheet with all the finished examples can be found at my dropbox here: *https://www.dropbox.com/s/s9zfib1bvdgjcqq/How%20of%20dataviz%20-funded%20ratio%20viz.xlsx?dl=0*

CONDITIONAL FORMATTING

I demonstrated a use of conditional formatting in the article "The What of Data Visualization," and I saw Bob Crompton showed a use of conditional formatting in his article "Data Visualization for Model Controls" (*Predictive Analytics and Futurism* newsletter, June 2017). Conditional formatting has been available for use in Excel even before Excel 2007, but options have been added over time and certain built-in rules are in Excel itself.

Conditional formatting is unique to spreadsheet and table setups, because the concept is that your data—whether numeric or categorical—will be highlighted or colored in some way reflecting the contents of each cell. In Excel, one has been able to do conditional formatting based on a value being greater than or less than certain amounts, within a certain range, etc. One has been able to conditionally format a cell not only based on the value within that cell, but also based on a formula or value in a different cell. It has become quite flexible in its current implementation.[2]

Let me demonstrate some of the more complicated, built-in conditional formatting rules.

In Table 1, I have public pension plans and their funded ratios listed over time by funded ratio. I have selected the largest plans, by assets in 2015. I have data for 2001–2015, but I will show only the last five years in the following table.

Table 1
Our Original Data of Funded Ratios

GASB Funded Ratio by Fiscal Year	2011	2012	2013	2014	2015
California Teachers	69.3%	67.2%	66.9%	68.5%	68.5%
California PERF	82.6%	83.1%	75.2%	76.3%	73.1%
NY State & Local ERS	90.2%	87.2%	88.5%	92.0%	93.8%
Florida RS	86.9%	86.4%	85.4%	86.6%	86.5%
Texas Teachers	82.7%	81.9%	80.8%	80.2%	80.2%
New York State Teachers	96.7%	89.8%	87.5%	92.9%	94.2%
Wisconsin Retirement System	99.9%	99.9%	99.9%	100.0%	100.0%
Ohio PERS	77.4%	80.9%	82.4%	83.8%	85.0%
Ohio Teachers	58.8%	56.0%	66.3%	69.3%	69.3%
North Carolina Teachers and State Employees	94.0%	94.2%	94.8%	95.6%	92.5%

This is difficult to do much in the way of comparison. It's just a bunch of numbers, with a similar number of digits. Nothing stands out visually.

If I highlight the cells containing the funded ratio amounts, I can select "Conditional Formatting" from the Home ribbon, which gives me a variety of choices, as seen in Figure 1. There are rules

Figure 1
Screenshot of the Conditional Formatting Dropdown Menu in Excel

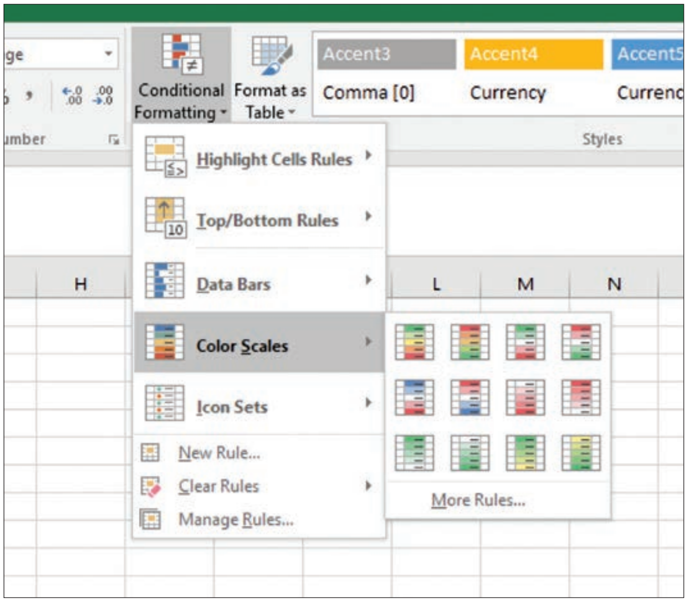


Table 2
Conditionally Formatted Table, Using Excel Default Choices in Three-color Scale

GASB Funded Ratio by Fiscal Year	2011	2012	2013	2014	2015
California Teachers	69.3%	67.2%	66.9%	68.5%	68.5%
California PERF	82.6%	83.1%	75.2%	76.3%	73.1%
NY State & Local ERS	90.2%	87.2%	88.5%	92.0%	93.8%
Florida RS	86.9%	86.4%	85.4%	86.6%	86.5%
Texas Teachers	82.7%	81.9%	80.8%	80.2%	80.2%
New York State Teachers	96.7%	89.8%	87.5%	92.9%	94.2%
Wisconsin Retirement System	99.9%	99.9%	99.9%	100.0%	100.0%
Ohio PERS	77.4%	80.9%	82.4%	83.8%	85.0%
Ohio Teachers	58.8%	56.0%	66.3%	69.3%	69.3%
North Carolina Teachers and State Employees	94.0%	94.2%	94.8%	95.6%	92.5%

Figure 2
Conditional Formatting is Addictive … Beware of too Many Elements!

2011	2012	2013	2014	2015
69.3%	67.2%	66.9%	68.5%	68.5%
82.6%	83.1%	75.2%	76.3%	73.1%
90.2%	87.2%	88.5%	92.0%	93.8%
86.9%	86.4%	85.4%	86.6%	86.5%
82.7%	81.9%	80.8%	80.2%	80.2%
96.7%	89.8%	87.5%	92.9%	94.2%
99.9%	99.9%	99.9%	100.0%	100.0%
77.4%	80.9%	82.4%	83.8%	85.0%
58.8%	56.0%	66.3%	69.3%	69.3%
94.0%	94.2%	94.8%	95.6%	92.5%

to highlight top values (or lowest values), or highlight specific cells, but if I want to do a visualization comparing multiple quantities, I find the color scales the most useful of the choices.

This, after all, is a matter of the states, so I feel like using the Red/White/Blue color scheme, as seen in Table 2. Now we see that the California Teachers plan and Ohio Teachers plan have relatively low funded ratios and Wisconsin and New York State Teachers have relatively high ratios.

Conditional formatting is great for creating dashboards within your Excel spreadsheets in general. It can help you get a high-level view of data you're actively working with, and makes for a relatively simple interface if you want to create reporting dashboards.

Be careful: the conditional formatting rules can "stack." After one rule is implemented, if one highlights the same data … well, check out Figure 2. A little conditional formatting can go a long way.

SPARKLINES

Sparklines were popularized by Edward Tufte, who wrote the following:

> A sparkline is a small, intense, simple, word-sized graphic with typographic resolution.
>
> Sparklines mean that graphics are no longer cartoonish special occasions with captions and boxes, but rather sparkline graphics can be everywhere a word or number can be: embedded in a sentence, table, headline, map, spreadsheet, graphic. Data graphics should have the resolution of typography.[3]

The concept is to be able to mix text and other elements with data visualization. However, text generally works by having discrete forms from a small set, thus ensuring "readability" even when type size gets small. When graphs are shrunk to the height of text lines, certain small gradations are difficult to distinguish, so most sparkline implementations come with some features to emphasize high or low points. Sparklines tend to be most used with time series, as the natural tracking of quantitative entities over time works well with small lines, especially if the overall trend is of interest and not small differences.

Sparklines are also implemented in Excel, starting with Excel 2010.[4] In their Excel implementation, a single sparkline "lives" within a regular Excel cell, allowing one to put a mini-graph next to data, or making it easy to construct dashboards native to Excel. There are three types of Excel sparklines: line, column and win/loss. I will show examples of each, and why one may wish to use them.

Given that sparklines are intended to live side-by-side with text, the other major package/language they're implemented in is LaTeX. A pdfLaTeX implementation can be seen here: *https://ctan.org/pkg/sparklines*.[5] The sparklines package allows the mini-graphics to be embedded in a final pdf document created via LaTeX.

Let's go back to our table, and put back all the years 2001–2015. Say we want to compare the general trend for the funded ratios for all the plans, but don't want to have 10 lines on a single graph.

Sparklines can be found on the Insert ribbon in Excel, in its own grouping. Place the cursor in the cell in which you want to place the sparkline, choose which type you want, and select the data you want graphed. This can be seen in Figure 3.

Figure 3
Setting the Sparkline for the First Cell

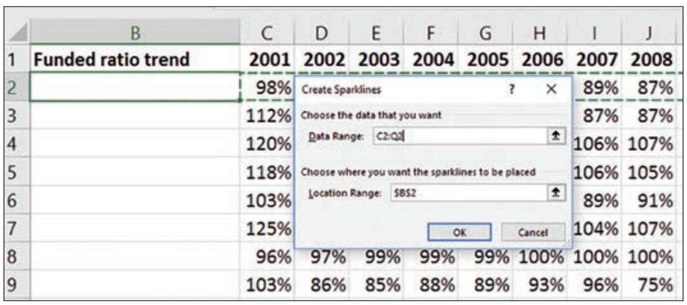


Figure 4
Sparkline Design Menu can be Seen in the Ribbon

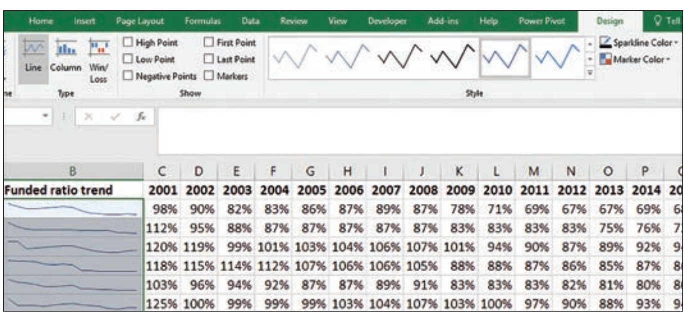
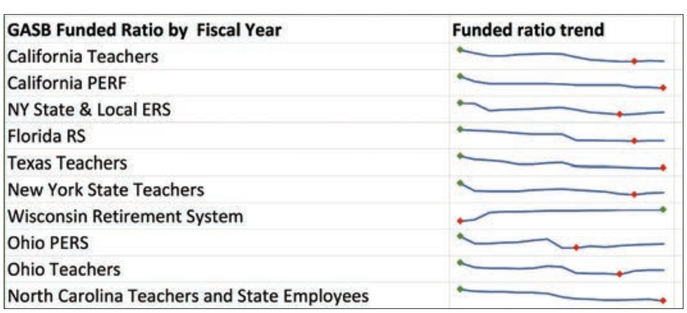


Figure 5
A Final Table With Sparklines Within the Spreadsheet



The real magic comes when you copy one sparkline over a range. Just as with normal Excel formulas, this is treated in a relative manner, and the copied set are all linked to the original as a grouping. See Figure 4 to see the sparkline design menu, where you can edit features.

One can make several choices, such as forcing all the sparklines into the same vertical axis, highlighting the high or low point, and a few other choices. The point is to make small graphs with few features that are easily readable in a small space. One cannot provide many options at that point.

In my final version in Figure 5, I have hidden my data, marked the high and low points, and let the max and min for each graph differ by plan. The resulting visualization gives general trends but doesn't allow for comparison of actual amounts between plans. As this is a spreadsheet-native graphic, this is also useful for Excel-based dashboards.

SLOPE GRAPHS

Slope graphs are a bit odd, in that they are a type of graph you already know: line graphs. Except there are only two points on each line: the begin and the end. The standard style for a slope graph is that the name of the data series is at the begin and/or end of each line, instead of a legend separate from the data itself.

The point of slope graphs is to give a very high-level feel for changes in different quantities, but it's also a way to compare rankings of many items in a before-and-after sense. I've seen slope graphs used to show how survival rates for different types of cancer have changed for decades ago versus recently. Because each graph element is a straight line segment, it makes it relatively easy for people to have multiple lines on a single graph in order to compare them.

This is not a built-in graph type for Excel, but one can easily create these via the built-in line graphs, adding data labels, and coloring lines differently by slope (if one wants to make that distinction.) A few ways of implementing these in Excel can be seen in the endnotes.[6]

For the funded ratios from 2001 to 2015 for our top 10 pensions, see the trend in Figure 6 on the next page.

TILE GRID MAPS

Here's my favorite new dataviz technique I've been using: tile grid maps. I had originally thought about writing about

Figure 6
On, Wisconsin!

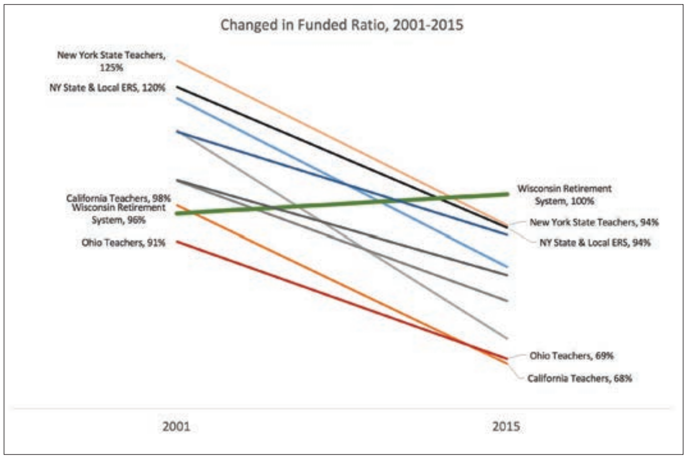
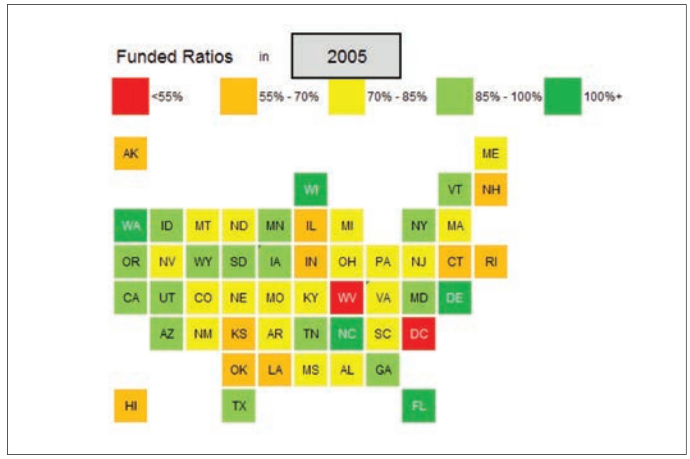


Figure 7
Public Pension Funded Ratios Before …



choropleths, which is a way of coloring geographically-accurate maps based on underlying data—and even these are built into Excel now (I put one in the spreadsheet accompanying this article).

But do we really need to look at all the squiggly-lined items? Especially if we're looking at data that isn't necessarily geographically-determined (such as hurricane incidence) but more generic? Some people have done tables, column charts, and line graphs with the items marked by state abbreviation, but that's difficult to look at.

The tile grid map is a great solution: one uses the familiar general geographic placement of the states (or countries, or whatever geographic units) in a grid pattern (usually square, but can be hexagonal or other). Then one colors the regions based on value, similar to conditional formatting. Indeed, for an in-cell implementation in Excel, one uses conditional formatting![7] Figures 7, 8 and 9 show the tile grid map for the states for different years. (total pension assets over total pension liabilities). I used a drop-down menu to select the fiscal year, conditional formatting colors the squares based on a lookup formula, and the state abbreviations on each square? Well, check out the resources on tile grid maps to see how it was done.[8]

Tile grid maps are being used more in data journalism, but different publications prefer different configurations for the U.S. states. Where exactly should one put Rhode Island? To the east of Connecticut or east of Massachusetts?[9] (Once National Public Radio put Rhode Island to the west of Massachusetts in a post on its website, which was odd.)

There have been tile grid maps created for the countries of Europe, for the provinces of Canada, for the states of Mexico

Figure 8
… and After the 2008 Market Crash

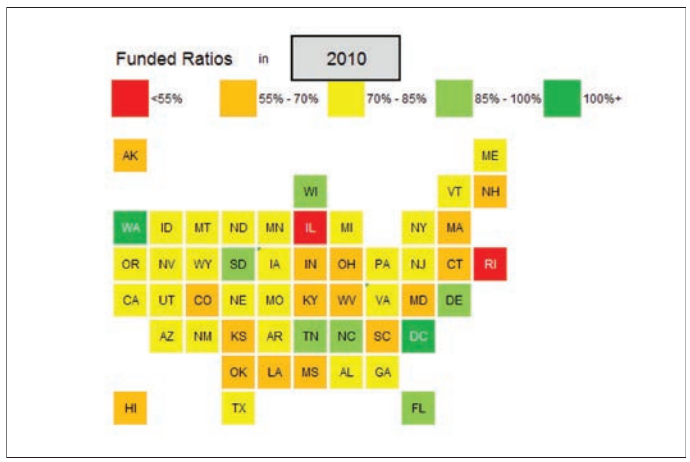
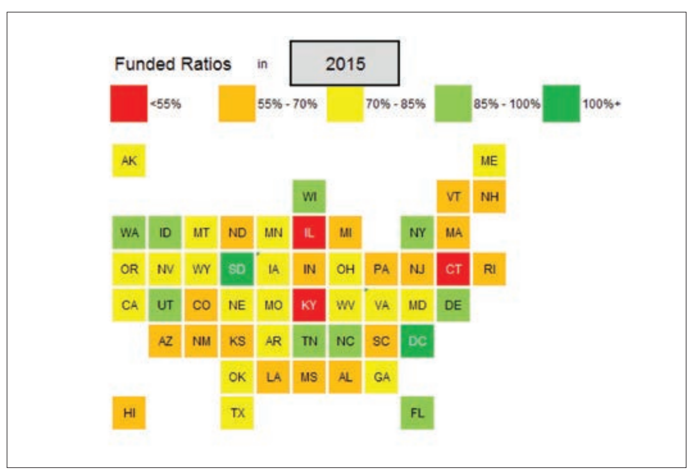


Figure 9
It's Still Getting Worse?



… for pretty much anything one wants, as they are easy to build compared to some other visualizations.

EXPLORE ON YOUR OWN!

In putting together this article, I had thought of techniques I like using that I don't think are necessarily commonly used, but I also used some online dataviz catalogs I found over the past year. I found the tile grid map through one of these galleries and realized it was a great solution for a problem I was having with particular data sets.

Why not check out some of these yourselves and try something new today!

Financial Times' visual vocabulary: *https://github.com/ft-iwnteractive/chart-doctor/blob/master/visual-vocabulary/Visual-vocabulary.pdf*

Chartmaker Directory: *http://chartmaker.visualisingdata.com/*

Python Graph Gallery: *https://python-graph-gallery.com/*

R Graph Gallery: *www.r-graph-gallery.com*

Visualization Universe: *http://visualizationuniverse.com/charts/*

Graphic Continuum: *https://www.informationisbeautifulawards.com/showcase/611-the-graphic-continuum*

Data Viz Project: *http://datavizproject.com/*—2017 Kantar Information is Beautiful award winner

Let me know what your favorite new dataviz technique is!

Mary Pat Campbell, FSA, MAAA, is a vice president, Insurance Research at Conning in Hartford, Conn. She can be reached at *marypat.campbell@gmail.com*.

ENDNOTES

1 Public Plans Database: *http://publicplansdata.org/, downloadable data: http://publicplansdata.org/public-plans-database/download-full-data-set/* Accessed 11 December 2017.

2 Check out the Microsoft support on Conditional Formatting in Excel for some tips:*https://support.office.com/en-us/article/Enter-and-format-data-fef13169-0a84-4b92-a5ab-d856b0d7c1f7#ID0EAABAAA=Conditional_formatting*

3 Edward Tufte Forum: "Sparkline theory and practice". *https://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=0001OR*, accessed 12 Jan 2018.

4 Office Support: "Use sparklines to show data trends". *https://support.office.com/en-us/article/use-sparklines-to-show-data-trends-1474e169-008c-4783-926b-5c60e620f5ca*, accessed 12 Jan 2018.

5 Sparklines package, version 1.7 as of 12 Jan 2018 (version date: 27 Dec 2016)

6 Make a slope graph in Excel, *http://stephanieevergreen.com/slopegraph/*, accessed 30 Jan 2018. How to Make Slope Graphs in Excel, *https://peltiertech.com/slope-graphs-in-excel/*, accessed 30 Jan 2018.

7 Tile grid maps in Excel, *https://policyviz.com/2016/04/13/tile-grid-maps-in-excel/*, accessed 30 Jan 2018

8 Other resources: *https://www.gislounge.com/how-to-make-a-tile-grid-map-using-excel/, https://policyviz.com/2016/05/05/hexagon-tile-map-excel/, https://www.linkedin.com/pulse/excel-map-hack-john-nelson/, https://policyviz.com/2017/05/04/european-tile-grid-map/*

9 *http://blog.yanofsky.info/post/117635988235/there-appears-to-be-some-disagreement-on-the*, accessed 30 Jan 2018.

ACORD: Setting Standards for the Global Insurance Industry

By ACORD

Editors' note: Below is a self-description of the non-profit Association for Cooperative Operations Research and Development (ACORD). While the Technology Section doesn't endorse this association, the editorial team felt it would be quite beneficial to let our members know about insurance industry standard-setting around data exchanges and beyond. Enjoy the read.

ACORD, the global standards-setting body for the insurance industry, facilitates fast, accurate data exchange, and efficient workflows through the development of electronic standards, standardized forms, and tools to support their use.

For nearly 50 years, ACORD has been an industry leader in identifying ways to help its members make improvements across the insurance value chain. Implementing ACORD Standards improves data quality and flow, increases efficiency, and realizes cost savings to the global insurance industry.

Currently, ACORD engages more than 4,000 participating organizations spanning 20 countries, including insurance and reinsurance companies, agents and brokers, software providers, financial services organizations and industry associations. With the tools and resources provided by ACORD, these participants are equipped to deal with the current business environment while influencing and shaping the future of the industry.

Insurance is a unique industry that centers on a promise: that the consumers' risks are understood, insured, and in the event of a claim—paid. This promise is grounded in data, the lifeblood of insurance, which flows from the very first interaction to the final transaction and beyond.

ACORD's objective is to enable efficient and effective flow of insurance data. All insurance transactions rely upon timely and accurate exchange of data across stakeholders. With ACORD's data standards and solutions, these exchanges of information are possible, and the insurance industry is able to operate optimally and provide the best experience for its consumers.

Data standards are particularly critical for the insurance industry because of three major factors that have complicated the business environment:

- Insurance stakeholders have varying business and technical needs, all intended to serve the customer and add value throughout the process. However, these varying, and sometimes competing, needs add complexity to the insurance business model.
- The exponential growth of data across the industry also adds another layer of complication. Organizations have the opportunity and obligation to consume ever-increasing volumes of data, which can help improve business operations but also can make data management more difficult.
- Changing regulations can create issues for organizations across the industry, costing time and money trying to navigate the evolving regulatory landscape.

ACORD views the value of standards through three lenses:

- **Efficiency:** Most insurance stakeholders are asked to do more with less. Implementing process, organization and updated technology can help organizations increase efficiency.
- **Effectiveness:** Effectiveness can only occur if an organization is meeting its objectives. Not only are standards useful in helping organizations reach their existing goals, but they can take the organization to the next level—increasing their value.
- **Flexibility:** The current global business environment is full of uncertainty. The ability for organizations to plan in order to react when the unexpected happens is a crucial survival skill. Standards give organizations the foundation to adapt successfully to an evolving business landscape.

Once organizations adopt and implement data standards to address the challenges they are facing, they realize capability improvements across process, organization and technology dimensions.

To read more about *The Value of Standards* whitepaper developed by ACORD, please visit *www.acord.org*.

ACORD can be contacted at *communications@acord.org.*

SOCIETY OF ACTUARIES®

2018 Predictive Analytics Symposium

September 20–21

Minneapolis, MN

Explore the world of big data and how it impacts the actuarial profession.

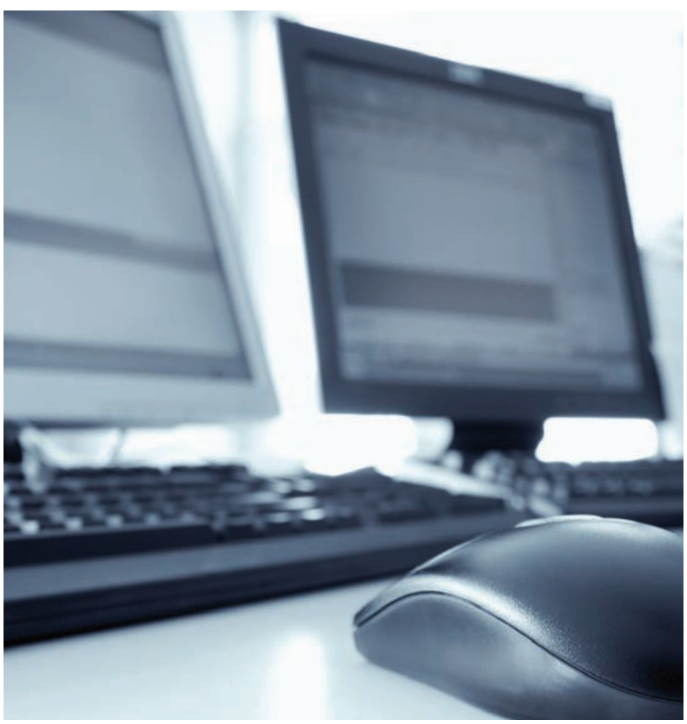
Register at Soa.org/PASymposium

Small Company, Modern Data Strategy

By Ying Zhao and Win Georg

Predictive analytics and big data have become buzzwords of the insurance industry and actuarial profession. Before companies can dip their hands into the ocean of big data, they need a solid foundation for managing and analyzing their existing internal data. This requires an infrastructure that acquires, integrates and manages the entire enterprise data resources. Many companies have been conducting actuarial and financial transformation projects for the past few years to establish this foundation, which has lead to streamlined reporting processes, improved financial analytics and enhanced internal controls.

While large companies are busy with their technological and actuarial innovations, many small and medium-sized life insurance companies are still trying to figure out how to participate in the world of new technology and big data. This article will outline why small insurance companies should act now to



enhance their reporting and analytical capabilities, how to start a seemingly overwhelming project, and what critical factors are needed to ensure the success of such technological and business transformation projects.

Historically, many small companies have been successful by occupying niche marketplaces, providing high-quality customer service and enjoying a capital-rich operating environment. While the industry is gearing up for significant industry changes like principle-based reserves (PBR), some small companies have not felt the same pressure because of PBR's small company exemption. So do small insurance companies need to conduct transformation projects like the big companies have been doing? The answer is YES, and they need to start NOW, for the following reasons.

- The insurance industry has an aging sales force, which is projected to retire in massive numbers in the coming years. InsurTech startups have started their disruption of the traditional distribution model and are attempting to establish new relationships with the end customers. The niche markets that some small companies have been occupying will undoubtedly be affected by this sea change as well. The mom-and-pop approach to customer care will also slowly lose its appeal as the new generation of customers are more technology savvy and demanding of information at their fingertips. Small companies need to upgrade their front and back office technologies in order to stay in the marketplace.

- An increasing demand for better controls and risk management from regulators and auditors has been an industry theme for many years. As more states adopt the model audit rule (MAR) and modern risk management framework (i.e., own risk solvency assessment (ORSA)), companies need to enhance their reporting capabilities and streamline their reporting and control processes to meet the regulatory requirements.

- A changing marketplace brings challenges as well as opportunities. Some companies pursue rapid growth following a merger and acquisition (M&A) strategy. A robust data infrastructure is one of the prequests for successful business integrations and winning the M&A game.

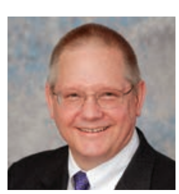
Fortunately, many business leaders already recognize the need to change. However, they are hesitant to act due to the perceived large size of potential investment and the scope of the projects. In addition to the common industry issues, such as legacy administration systems and outdated data infrastructure, small companies may have additional challenges such as limited analytical capabilities and fragmented reporting processes. Some critical business analyses are heavily reliant on capable individuals (mostly actuaries) and performed on desktop applications using personal computing technologies (instead of enterprise technology solutions). So is it possible for small companies to take on transformation projects? The good news is that the technological advancements in the past few years have created many different solutions and now allow companies to take a more flexible approach to such projects.

- Cloud-based database and computing technology have matured to the point that it is a viable alternative to on-premises hardware and software and their associated support costs.
  - The technology, architecture and best practices associated with data warehouse design are mature and well understood by practitioners. Cloud computing offers quick and flexible scalability. If architected correctly, a data management/business intelligence system can be implemented at small scale with modest cost initially and can grow quickly to meet expanding business needs. In fact, systems can be configured to scale dynamically up and down as data volume and computing loads increase and decrease at various times.
  - Significant data management and reporting benefits can be realized with a modest initial investment, allowing a re-engineering effort to be started without a large financial commitment. Since the effort can be started on a small scale, a great deal of the risk associated with a large traditional IT project is automatically eliminated.
- Extensive cloud-based development can be undertaken by a very small team. Relieved of the need for space in the local data center, resources from a backlogged infrastructure support staff, and the delay required to obtain and install new server hardware, a few knowledgeable individuals can create a surprisingly extensive system. A nimble development team can also form a close relationship with the business units and provide quick responses to changing requirements. An agile project manangement approach will increase communication frequency, improve information handoff and shorten release cycle.
- The vendors of cloud-based database platforms have responded to modern regulatory requirements by implementing such features as database auditing, encryption at rest and geo-replication of data. These features allow an organization to meet Sarbanes-Oxley Act (SOX) and MAR requirements in their databases without the need to build and maintain such functionality themselves.
- Some administrative systems and software tools that companies have been using have added analytical capabilities as an extension to their existing systems. These new modules can produce canned or customized visual analysis of the business data residing in the system. This allows companies to bypass data validation, because the analyses come directly out of those systems, and to produce control documents and management reports with a few clicks. While disparities among different systems still exist, this option can provide an immediate solution for data and analytical needs to companies as they explore long-term enterprise data solutions.
- Many InsurTech startup companies have come to the marketplace providing technology solutions to one specific area of the insurance business processes, such as application, sales management, underwriting, claim process, etc. Insurance companies can strategically select the areas that they would like to address and establish business relationship with these companies. Like other technology companies, InsurTech companies may not follow a traditional business model and may allow more flexible forms of relationships other than the traditional buyer-seller relationship, hence a more flexible cost structure for insurance companies.

People in the "transformation business" know that transformations must happen in all aspects of the business, not just technology. Business process, organization structure and personnel need to go through transformations as well. The human factor is as necessary as the machine factor. Without one, the other will be unlikely to succeed. Companies need to train their existing staff and acquire new talents to accommodate the new processes. Actuaries, especially those who are the primary producer of the business intelligence in the small companies right now, can and should become the leading force of these transformations.

Last and most importantly, no transformation can be successful without a strong commitment and support from company's business leaders. A visionary leadership, strategic investment and commitment to success will take small companies to the brave new world of big data and predictive analytics, and find business growth and success in the new era of the insurance industry. ■

Ying Zhao, FSA, MAAA, is AVP–Product Management for Lincoln Benefit Life in Rosemont, Ill. She can be contacted at *ying.zhao@lbl.com*.



Win Georg, is AVP–Information Technology for Lincoln Benefit Life in Rosemont, Ill. He can be contacted at *win.georg@lbl.com*.

Introduction to Distributed Computing

By Jason Altieri

The rise of big data has required changes to the way that data is processed. Distributed computing is one approach used to meet expanding processing capacity needs driven by continually growing datasets. In *Designing Data Intensive Applications* Martin Kleppmann states, "Many applications today are **data-intensive**, as opposed to **compute-intensive**. Raw CPU power is rarely a limiting factor for these applications—bigger problems are usually the amount of data, the complexity of data, and the speed at which it is changing."[1] Where single machines or relational databases used to be sufficient for data processing and analysis, obtaining single machines that can handle today's quantity of data has become cost-prohibitive if not impossible. New software and processes have become necessary to allow efficient and effective use of this data.

In the actuarial world, these technologies have become important to support more advanced modelling on larger and more complicated data. The actuarial profession has been evolving to include more predictive analytics and these methods often require more compute power than traditional actuarial modelling. In addition, even the datasets used to support more traditional actuarial work have grown in recent years. For example, large reference datasets used for benchmarking and government databases, such as Medicare or Medicaid claims can be large. This combination of factors makes an understanding of these new technologies important for actuaries working in a modern data environment.

INTRODUCTION TO PARALLEL DISTRIBUTED COMPUTING

Distributed computing is a framework for handling large quantities of data and complex processes by increasing the amount of hardware applied to a task. Instead of using a single machine, a distributed computing system allows a network of machines to work together to complete a process. There are two ways to leverage the network to complete the process, which is referred to as parallelization. First, different nodes within the network can simultaneously work on different tasks in the process, as long as the tasks are not linearly dependent. Second, by taking a large task and splitting it up into smaller self-contained pieces, multiple nodes on the network can contribute to the completion of the same task. In practice, both of these approaches can be used within the same process provided there is a sufficiently large network.

Several factors have contributed to the rise of distributed computing, including increasingly large datasets, the popularity of statistical learning, and affordable access to hardware (often via cloud providers). Increasingly large datasets have made it more difficult to perform analysis on single computers, both because of memory and time constraints. Even if a single machine is capable of processing a large dataset it may be too slow, especially in industries that require the ability to react quickly to new data. Many common statistical learning algorithms applied to these large datasets also lend themselves to a distributed computing framework. Training these models, particularly on large datasets, can be memory and processing intensive. Additionally, some models benefit from running many iterations of the same model, and almost all statistical learning techniques utilize resampling to tune their accuracy. Both of these factors make statistical learning algorithms a perfect fit for parallelization. Finally, distributed computing has benefited from easier access to hardware. The rise of cloud computing resources has made large amounts of machine time and power broadly available to both companies and individuals. Cloud computing has also enabled users to access a large network of machines for just a limited amount of time and only pay for what they use. These factors, among others, have helped make distributed computing a popular tool for people who work with large quantities of data.

BENEFITS AND DRAWBACKS OF A DISTRIBUTED SYSTEM

Like with most technologies, there are both benefits and drawbacks to the use of distributed computing. Some of the advantages are:

- Distributed computing scales up effectively to very large datasets,
- acquiring large amounts of memory or processing power may be more affordable by networking a series of less expensive machines than buying one sufficiently powerful machine,
- externally maintained infrastructure such as cloud computing platforms can be leveraged and
- can decrease processing time, especially in non-linear pipelines.

The cost savings can be significant, particularly for a large cluster. According to *Designing Data Intensive Applications* "…cost is super-linear: a machine with twice as many CPUs, twice as much RAM and disk typically costs significantly more than twice as much."[2] However, there are disadvantages that can make distributed computing impractical to implement:

- Not all algorithms are good candidates for parallelization,
- lack of efficient scaling down to smaller data,
- overhead in getting programs up and running on additional nodes, and
- overhead in orchestrating the parallelization.

These disadvantages can be a significant barrier to the use of distributed computing in some cases. First, there are some algorithms and use cases that do not fit well in a distributed framework. The distributed computing framework requires the ability to separate data or distinct tasks, and this process does not work well in certain cases. Second, while a distributed approach scales up to very large data effectively, downscaling can be problematic. Each node on the network needs to be notified a task needs to be done, load up the environment to perform the task, and communicate results back to the main process. When the data is small this cycle can take more time than it would take to complete the process on a single machine. Additionally, there is a substantial amount of overhead involved in maintaining the parallelization. Systems need to exist to communicate what work needs to be done, manage the status of the processes on different nodes, and coordinate the compiling of results. Building and implementing a system capable of doing this is a significant investment, which can become prohibitive if a real need does not exist. Fortunately, systems exist to handle this communication and facilitate the use of distributed computing.

DISTRIBUTED COMPUTING TECHNOLOGIES

Several distributed computing technologies exist to help solve the problems related to managing a distributed computing system. There are two categories of solutions: MapReduce implementations and workflow managers. MapReduce implementations use a two-step process to break the data up and distribute it to different nodes, then re-aggregate it to determine a result. Workflow managers use dependencies between tasks to determine if there are tasks that are independent of the results of other tasks. The workflow manager then distributes the independent tasks to different nodes to allow for parallel completion of the tasks. The following is a non-exhaustive list of these solutions:

MAPREDUCE IMPLEMENTATIONS
- Apache Spark,
- Hadoop MapReduce,
- Disco,
- Dask and
- Teradata.

WORKFLOW MANAGERS
- Luigi,
- Airflow,
- Azkaban and
- Oozie.

The remainder of this article will focus on MapReduce, and dive specifically into Apache Spark.

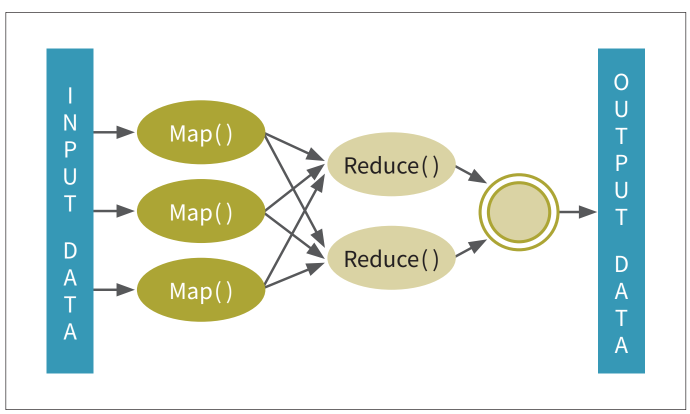
WHAT IS MAPREDUCE?

MapReduce is a process designed to facilitate parallel operations (See Figure 1). The original public implementation was part of the Hadoop ecosystem; however, the same general MapReduce concepts are used in other frameworks. As the name implies, the process is composed of two functions: A map function and a reduce function.

The map function breaks the data up into independent partitions. It then distributes these partitions to various nodes on the network for parallel processing. Each partition will output a group of key-value pairs that completes as much of the process as possible at the independent partition level. As each partition finishes processing, these key-value pairs need to be re-aggregated. This is the purpose of the reduce function. The reduce function pulls the results of each partition back into the main process and further aggregates them to determine the result at the full dataset level.

Manually implementing MapReduce is possible; however, it can be very difficult to do for even moderately complex processes. In order for a MapReduce process to be efficient, it is important to distribute the data to maximize the amount of work performed at the partition level. Additionally, it is important to minimize

Figure 1
The MapReduce Process



the frequency of re-aggregating the results, as that step does not benefit from the parallel framework. Fortunately, MapReduce solutions such as the ones listed above automatically perform this optimization using query planning and analysis tools.

OVERVIEW OF APACHE SPARK

There are many implementations of MapReduce-based distributed computing frameworks with different benefits and drawbacks. Depending on the infrastructure in place and the use case, the best implementation may vary. Apache Spark is one such implementation used as a more detailed example implementation of a MapReduce framework.

According to *Learning Spark*, "Apache Spark is a cluster computing platform designed to be fast and general-purpose."[3] It started as a research project at UC Berkeley back in 2009 by a lab working with Hadoop MapReduce. The researchers identified interactive querying and iterative development as a weakness of the Hadoop implementation and sought to improve it. Early results were positive; showing speed improvements in the 10x–20x range, and the performance has since improved to 100x faster on in-memory jobs. Spark was open-sourced in 2010 and became part of the Apache Software Foundation in 2013. Per the Spark documentation, it scales up to petabytes of data and clusters as large as 8000 nodes in practice.

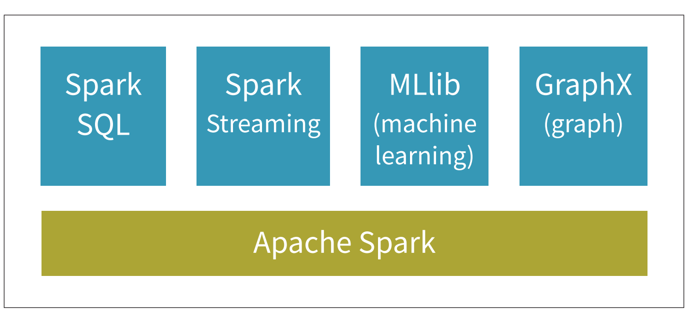
HOW DOES SPARK WORK?

On a technical level, Spark is written in Scala and runs on the Java Virtual Machine. It uses a concept called "Resiliently Distributed Datasets" (RDDs) to support its parallelized operations. According to *Learning Spark* "RDDs represent a collection of items distributed across many compute nodes that can be manipulated in parallel." [4]

Another key element of Spark is tightly integrated components. Spark is broken up into six main components:

- Spark Core: task scheduling, memory management, and other basic functions,

Figure 2
The Spark Ecosystem



- Spark SQL: querying and data manipulation for mostly structured data sources,
- Spark Streaming: API to work with live data updates,
- MLlib: scalable implementations of machine learning algorithms,
- GraphX: library for graph analysis and computation, and
- Standalone Scheduler: built-in cluster manager.

Spark Core is the foundational component that enables the system as a whole to function. From there, the other components act as extensions that allow the user to perform specific tasks such as querying or machine learning. The integration of these different components allows a user to switch between different types of tasks while remaining inside the Spark ecosystem.

Spark can also integrate with other common cluster managers such as Hadoop, YARN and Mesos. This allows Spark to be deployed inside of existing distributed computing infrastructure. Meanwhile the Standalone Scheduler allows deployment of Spark in cases where there is no existing distributed computing infrastructure.

BENEFITS OF SPARK

There are several benefits to using Spark as a distributed computing framework. For instance, knowledge of Scala programming is not necessary to work with data in Spark. There are convenient wrappers available to allow users to interact in a more familiar language such as python (PySpark) or R (SparklyR). Additionally, the APIs support the use of SQL syntax for data interaction. The availability of these common language interfaces helps reduce the learning curve for people looking to get started with Spark.

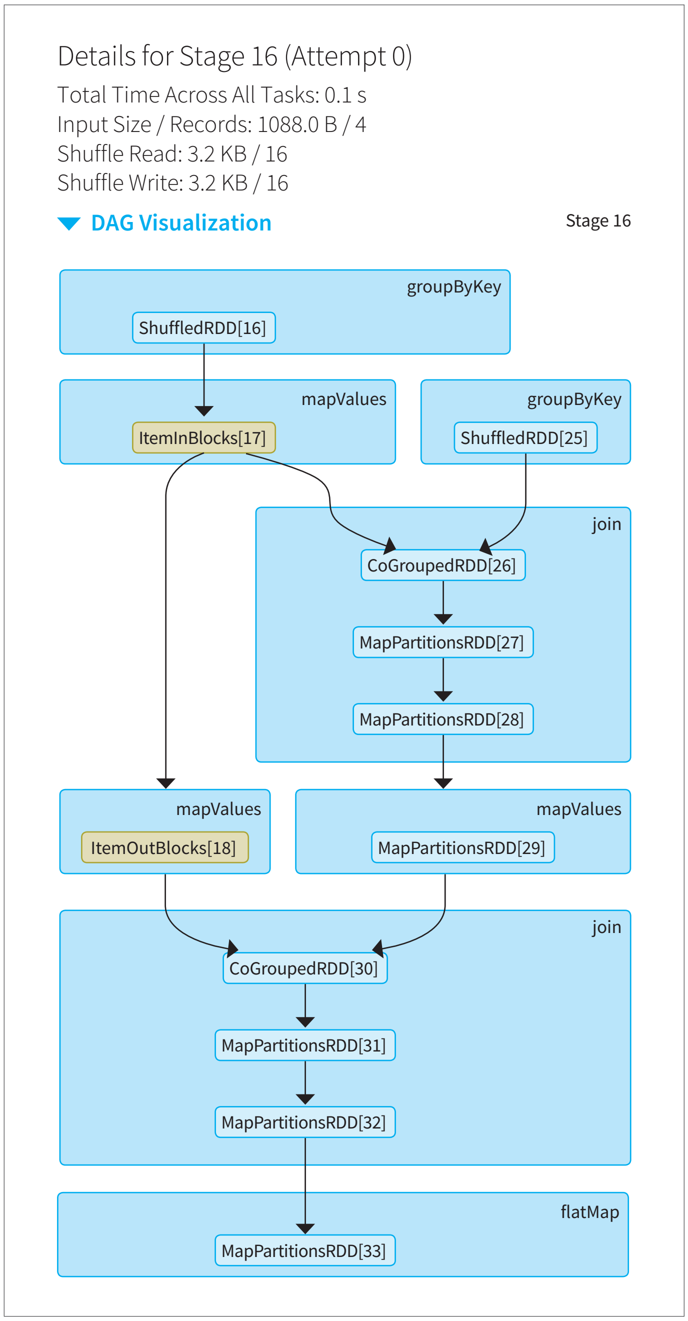
Spark also offers an interface that allows users to track the progress of jobs, data storage and query planning. The interface also offers a directed acyclic graph (DAG) to help visualize the execution of tasks. This allows for relatively straightforward performance monitoring and can assist with optimization of the system.

Additionally, while scaling down to smaller datasets can still be an issue; Spark handles data in the gigabyte range more effectively than some other options do. This makes Spark a viable choice for companies that have data in the gigabyte to terabyte range rather than the 100-terabyte range.

DRAWBACKS OF SPARK

Setting up the infrastructure necessary to run a Spark cluster can be challenging, especially on Windows-based systems. Spark was built to run on Linux, and its design choices reflect that. There is support for Windows; however, it is a clear second-class citizen and requires significantly more effort to implement and maintain.

Figure 3
Sample Directed Acyclic Graph (DAG)



In addition, while Spark scales down relatively effectively as a data manipulation and analysis language, the machine learning components do not. The performance of MLlib on smaller datasets does not compare favorably to common implementations in python and R. In particular, the Spark GBM implementation struggles on smaller datasets. As with any technology platform, it is important to understand the limitations of the specific implementation.

Finally, Spark lacks the flexibility of some lower-level frameworks, such as Dask, to build and control non-standard processes. Spark has some capabilities here, but they are mostly limited to the Scala language APIs.

GET STARTED

Getting started on working with Spark is easy. Databricks community edition offers free web-based notebooks running on top of pre-configured clusters in AWS. This removes the need to deal with setting up infrastructure for people who want to experiment with Spark. If you are interested in giving Spark a try, head over to *https://databricks.com/try-databricks*.

Jason Altieri, ASA, MAAA, is a data scientist with Milliman's PRM Analytics practice. He can be contacted at *Jason.Altieri@milliman.com.*

ENDNOTES

1 Kleppmann, Martin. 2017. *Designing Data-Intensive Applications: The big Ideas Behind Reliable, Scalable, and Maintainable Systems*. Sebastopol, Calif. O'Reilly & Associates, Inc. Pg. 3

2 Ibid. Pg. 146.

3 Karau, Holden, and Matei Zaharia, Andy Konwinski and Patrick Wendell. 2015. *Learning Spark: Lightning-Fast Data Analysis*. Sebastopol, Calif. O'Reilly & Associates, Inc.

4 Ibid. Pg. 3.

REFERENCES

Lockwood, Glenn K. "Map/Reduce Implementations." April 6, 2014. *users.sdsc.edu/~glockwood/comp/mapreduce.php*

Kempf, Rachel. "Members." Bizety. June 5, 2017. *www.bizety.com/2017/06/05/open-source-data-pipeline-luigi-vs-azkaban-vs-oozie-vs-airflow/*

"Apache Spark—Lightning-Fast Cluster Computing." Apache Spark—Lightning-Fast Cluster Computing, Apache, *spark.apache.org/*

"Comparison to Spark." Comparison to Spark—Dask 0.16.1 documentation, *dask.pydata.org/en/latest/spark.html*

"MapReduce." Hortonworks. *hortonworks.com/apache/mapreduce/#section_2*

"Analytics." What is MapReduce? IBM Analytics. *www.ibm.com/analytics/hadoop/mapreduce*

Parallel Cloud Computing: Making Massive Actuarial Risk Analysis Possible

By Joe Long and Dan McCurley

This article will walk through a cloud-use case where we were able to cut a three-month machine learning exploration project down[1] to just under four days using a mixture of open source tools and the Microsoft Azure cloud. This translates to an approximate 25-fold reduction in serial compute time for such a task. We will give a short introduction to the cloud while sharing our experience of managing the pool of data-crunching machines that ran our analysis. In closing, we will discuss lessons learned and ways to improve the plan of attack, as well as touch on the importance of state management to aid in efficiency and the reproducibility of results when using the cloud.

SETTING THE STAGE FOR THE CLOUD

Machine learning is spreading quickly across many industries and is showing promising results for making better predictions and automating manual tasks. However, with increases in data size and the greater power of more complex algorithms, the computing resources it takes to crunch the numbers increase as well. Nowadays, it may take days or months to conduct an analysis on a single machine. There is a solution though: thanks to advances in cloud computing, the phrase "the sky's the limit" has a whole new meaning as we now have the ability to speed up time if the reward outweighs the cost of doing so.

In order to utilize the time-saving efficiencies of the cloud, a large computational process must be able to be broken down into independent tasks that can be run in parallel. Not every process fits this mold. Some processes rely on a series of sequential calculations, where each calculation is dependent on the ones that precede it. An example of such a process would be calculating a single sequence of time-dependent events, which would not be a good use-case for the parallel compute capabilities of the cloud.

Machine learning, however, is full of many processes that can be broken down into independent tasks calculated in parallel, which can then be merged together after all independent calculations have been completed. A good example of this would be an ensemble method such as the random forest algorithm, which is used to develop a predictive model comprised of hundreds to thousands of independent decision trees that are averaged together to produce a single prediction. Another easily parallelizable example is the Monte Carlo simulation. These algorithms are prime candidates for the massively parallel computing abilities of the cloud. Almost all supervised learning algorithms use some kind of resampling technique (e.g., bootstrapping, cross-validation) to optimize the bias-variance trade-off for generalization. Most resampling techniques are embarrassingly parallel and can benefit greatly from cloud computing.

In our case, we used the cloud to help with a large machine learning exploration project, which was comprised of many calculations done in open source R. Our initial exploration started with a single heavy-duty, bare-metal machine that could handle traditional memory and compute intensive tasks. We quickly discovered that in order to run the full exploration analysis we mapped out, we would miss our deadline. Our initial estimate was that the full analysis—when run sequentially on our in-house machine—was expected to take 90 days of continuous computer run time. However, with some manual effort to break the analysis into semi-equal chunks, we estimated we could run it in Microsoft's Azure cloud and complete all of our calculations in less than a week. This approximately 25-fold reduction in serial compute time to run our analysis gave us more time to digest the results, giving us the ability to run further variants of our initial exploration plan. More variants can equal better value to the client.

THE MAGIC BEHIND THE CLOUD

"There is no cloud—it's just someone else's computer" is a common meme used to explain cloud services. While this phrase helps one understand the basic idea of the cloud, it does not fully recognize the great capabilities and flexibilities of the modern cloud infrastructure. The concept of the cloud dates back to the 1960s and is commonly attributed to J.C.R. Licklider and John McCarthy.[2] Joseph Licklider is credited for his core concept of a Galactic Network or "Network of Networks" and John McCarthy for theorizing utility computing. These ideas reached commercial viability in 2002 when Amazon Web Services (AWS) started providing web-based, pay-as-you-go services to companies to store data and run applications. Current major competitors to AWS include Microsoft Azure and Google Cloud.

All of these providers offer similar ways to access their resources. It is helpful to think of these resources in three main categories:

1. Infrastructure as a service (IaaS) creates a virtual data center in the cloud similar to what your company would have in an

information technology (IT) climate-controlled room. It's easy to adopt but expensive to run.

2. The second way to access cloud resources is through platform as a service (PaaS). In this method, the cloud provider takes care of storage and computation and provides a platform to do a focused type of work. If you want a database that is always available, but don't want to deal with any maintenance or tuning, this is an excellent solution.

3. Thirdly, software as a service (SaaS) allows a company to build a custom solution that can only exist in a cloud environment. Salesforce, Office 365 and G Suite are examples of SaaS.

Viewed in this context, our computing project was an example of an IaaS. But by the end of our exploration we had migrated much closer to a PaaS solution. The actual difference can get quite fuzzy.

THE LEARNING CURVE

Once we realized on-premise calculations would take too long, we turned to the task of determining how many (and what capacity) computers would be needed for a cloud solution. After a period of research on best approaches for parallelizing our process in the cloud, we estimated that 63 virtual machines (VMs) should be able to handle the work in a reasonable timeframe. Each machine had eight cores and 56 gigabytes of RAM, giving us a total of over 500 cores and 3,500 gigabytes of RAM at our disposal. For this project, we chose to provision the machines with Windows as the operating system due to familiarity, but we note this costs about 50 percent more in license fees than an equivalent Linux VM. We wrote PowerShell scripts to automate cloning and administration of the machines. Later in this article, we will describe a new tool that makes things much easier (and transitions this solution from pure IaaS to something closer to PaaS). At the time of our project, this setup had a sticker price of less than $2 per hour to run each virtual machine of this size in Azure.

Our first step was creating the initial VM and then installing R and all the R packages we would need to run our analysis. Once we had our initial VMs configured, we created 62 clones of it using the Invoke-Parallel PowerShell script Warren Frame discussed in his "Invoke PowerShell on Azure VMs" article,[3] which had some other helpful pointers we used along the way.

Now we had 63 VMs available to process data but hit a roadblock. How do we launch our R scripts on the VMs in a coordinated way? For this, we ended up using another script by Warren (Invoke-AzureRmVmScript) to invoke commands remotely on the VMs. We wrapped these commands in the Invoke-Parallel script to kick off the R scripts simultaneously across the VMs. An additional script served the purpose of deallocating VMs after the R scripts finished running to measure progress and limit costs. Allocated VMs charge per minute and deallocated VMs carry no compute charges.

Once all the VMs completed their tasks we collected our data and analyzed our results. In the end we ran a total of 90 days' worth of parallel compute time across the VMs, with the longest VM running for a total of three-and-a-half days at a total cost of around $3,000. The equivalent cost of buying and setting up similar machines would have required weeks of setup and tens of thousands of dollars of hardware purchase for the same result. Of course, the cloud approach also required a fair amount of time spent crafting and debugging the PowerShell scripts, which adds significant soft costs in addition to the hard costs. Additionally, when using an IaaS solution over time there would also be the ongoing costs associated with keeping the VM image up-to-date with the latest security updates.

THINGS KEEP ON EVOLVING

After completing our first large run in the cloud, we found that Microsoft was working on an R package simultaneously that automated many of the tasks we had done in PowerShell. This R package is called doAzureParallel, leveraging an Azure service called Batch. The package allows a user to create a pool of VMs in the Azure Batch service with a few lines of R code and then registers it as the parallel back-end for the R foreach package. If you are already familiar with the R foreach package then making the transition to using doAzureParallel is done simply by running some code that creates the pool in Azure Batch. Any existing foreach code using the %dopar% function can then be used as is.

Azure Batch allows you to easily launch a pool of Linux VMs, which as we mentioned earlier is much more cost-effective than using a pool of Windows-based VMs. The auto scaling features of Azure Batch allow dynamically scaling up or down the number of VMs in a pool based on the demand of the tasks you are running. Another option is to use a mix of dedicated or low-priority VMs in a pool. Cloud providers make excess compute capacity available at steeply discounted rates with the caveat that these machines can be interrupted by those willing to pay at the higher rate. If this happens the current task you are running gets canceled and reassigned on another low-priority machine. Therefore, it is recommended to only use the low-priority machines if you have short-running tasks or your calculation can progress despite multiple restart attempts.

> In our case, we used the cloud to help with a large machine learning exploration project.

One recently added feature of doAzureParallel worth noting is its ability to seamlessly run R inside a Docker container on the VMs within your pool. This is similar to how we cloned a custom VM image in our initial IaaS approach. It allows use of a prespecified environment that keeps R versions and packages in sync, which ensures reproducibility of results. The added benefit with the doAzureParallel Docker container approach is that now you can rely on Azure Batch to create up-to-date VMs each time you run an analysis, ensuring that you have the latest security updates. By default, doAzureParallel uses the "rocker/tidyverse:latest" image that is developed and maintained as part of the rocker project.[4] However, you can also specify a custom Docker image, which allows you to lock-in a version of R if you are concerned about duplicating results long-term.

In our case, doAzureParallel has helped us move our initial IaaS approach to more of a PaaS approach. Now we can rely on doAzureParallel to maintain the administration work of creating pools of VMs with up-to-date security updates, which are running our prespecified environments. Using such solutions allows users to focus more on the analysis they are trying to conduct rather than spending the time managing the infrastructure it runs on.

LESSONS LEARNED AND RECOMMENDATIONS

Taking a look back at our journey in the cloud we have some final recommendations for those looking to get the most out of these exciting new tools.

- If you plan on using the cloud for an analysis in R, check out the well documented doAzureParallel package. Even if you don't plan on using R for analysis you might find some workflows that help with other languages as well.
- The tools cloud providers have are constantly evolving and iterating and it is essential to be aware of what new tools are made available. For example, moving from the highly manual cloning of machines to Azure Batch for automated compute pool creation was revolutionary and much easier to use.
- We highly recommend the use of Docker containers or some other state management when conducting work in R or any other language if you need repeatable results over a long span of time.
- Finally, we recommend using Linux-based VMs over Windows if your task allows you to, as it can provide a welcome cost savings. Also investigate the use of low-priority VMs (or spot pricing in the AWS world) if your workflow supports short-running tasks.

Table 1 gives an estimate of potential cost reductions we could have achieved if we were to rerun our analysis applying these recommendations using the doAzureParallel package. For comparison, we have also estimated the cost of using AWS as the cloud provider. Note that these are estimated costs as of Jan. 23, 2018; pricing may vary in your region or the contract you have in place with Microsoft Azure or AWS.

Table 1
Potential Cost Reductions

VM Option	Total Compute Hours	Price per Hour[1]		Total Cost	
		Azure[2]	AWS[3]	Azure	AWS
Windows OS	2,151	$1.17	$1.05	$2,516.67	$2,258.55
Linux OS	2,151	$0.78	$0.67	$1,677.78	$1,441.17
Linux OS with low priority[4]	2,151	$0.14	$0.07	$301.14	$150.57

1. Estimated prices from Microsoft Azure and AWS online pricing for VM compute charges only. Does not Include storage or data transfer prices, which can become meaningful if not managed efficiently.
2. Azure A10 VM with eight cores and 56 gigabytes of RAM in the North Central U.S. region.
3. AWS r.3.2xlarge VM with eight cores and 61 gigabytes of RAM in the U.S. East (Ohio) region.
4. Assumes tasks were run without the VMs being preempted.

As you can see, the cloud is more than just someone else's computer. It's an ecosystem of resources that can be leveraged to explore ideas and complete tasks that were once unfeasible to achieve with the local computing resources of the past. ■

Joe Long is an assistant actuary and data scientist at Milliman. He can be reached at *joe.long@milliman.com*.

Dan McCurley is the cloud solutions architect at Milliman. He can be reached at *dan.mccurley@milliman.com*.

ENDNOTES

1 A research and development project conducted by the Milliman Advanced Risk Adjusters (MARA) product group. See *http://www.millimanriskadjustment.com* for more information about MARA.

2 Mohamed, A. A history of cloud computing. Computer Weekly. Retrieved Feb. 1, 2018, from *http://www.computerweekly.com/feature/A-history-of-cloud-computing*.

3 Warren F. Invoke PowerShell on Azure VMs. Rambling Cookie Monster. Retrieved February 1, 2018, from *http://ramblingcookiemonster.github.io/Invoke-AzureRmVmScript/*.

4 Tan, J.S. (Nov. 21, 2017). Scale up your parallel R workloads with containers and doAzureParallel. Revolutions. Retrieved Feb. 1, 2018, from *http://blog.revolutionanalytics.com/2017/11/doazureparallel-containers.html*.

Large Portfolio Variable Annuity Valuation Powered by GPUs and Deep Learning

By Huina Chen and Henry Bequet

Recent technological advancements in Graphical Processing Units (GPUs) and deep learning are drastically changing the landscapes of many fields, including financial analytics. In this paper, we apply GPUs and deep learning to address computational challenges in the valuation of large portfolios of variable annuities. Our numerical experiments show that using GPUs leads to a 10 times speedup compared with traditional Monte Carlo valuation on multi-threaded CPUs; while using GPU-based deep learning achieves another order of magnitude performance improvement.

INTRODUCTION

Variable annuity is a type of insurance contract that allows for asset accumulation via investing in mutual funds provided by insurers. It often provides guarantees, also called riders, to protect the policyholder from market downturns, such as the 2007–2008 financial crisis. The predominant guarantees are Guaranteed Minimum Death Benefits (GMDB), Guaranteed Minimum Accumulation Benefits (GMAB), Guaranteed Minimum Income Benefits (GMIB), and Guaranteed Minimum Withdrawal Benefits (GMWB).

Insurance companies holding large portfolios of variable annuity policies are exposed to risks from honoring the guarantees should adverse events occur. A popular risk management practice is to dynamically hedge these guarantees. Insurers buy hedging portfolios consisting of financial derivatives, hoping their payoffs offset the payouts of the guarantees to policyholders. The hedging portfolio requires intraday rebalance according to the Greeks, such as dollar delta, of the guarantee liabilities. Traditionally, the guarantees are evaluated using Monte Carlo simulations on every policy in the portfolio. This is because the product structure is complicated and the portfolio is highly heterogeneous (Gan and Lin, 2015). Monte Carlo simulations can be time consuming. For example, in one of our experiments, it took 64 CPU cores 44 minutes to calculate the dollar deltas for a portfolio of one million synthetic variable annuities with either a GMDB rider or both GMDB and GMWB riders for 10,000 scenarios.

Several papers have been published to efficiently evaluate the dollar deltas of large portfolios of variable annuities using a spatial interpolation framework (Gan, 2013; Gan and Lin, 2015; Hejazi et al., 2015). The idea is to select a small sample of variable annuity policies (the representative contracts), calculate their dollar deltas with the expensive Monte Carlo simulations, and then estimate the dollar deltas of other policies in the portfolio as weighted sums of the pre-calculated dollar deltas of the representative policies. The weights are determined according to the distances between the focal policy and the representative ones. Gan and Lin (2015) pointed out that a number of modeling choices significantly impact the accuracy of the spatial interpolation results, including the sampling method and the number of the representative contracts, as well as the distance function used to calculate the weights in the weighted sum. Hejazi and Jackson (2016) proposed a partial neural network to learn the distance function.

In this article, we use GPUs and deep learning to solve large portfolio variable annuity valuation problems with high speed and accuracy. It takes a GPU card with 4,992 cores less than one minute to calculate the dollar deltas for the one million policies using Monte Carlo simulations. The speed is attractive for intraday rebalances of the dynamic hedging program. However, GPUs alone cannot compute fast enough for capital calculation when nested simulations are required. If the inner loop simulations are replaced with the approximation function trained by deep learning, we can perform capital calculation on the same portfolio 10 times faster than the nested Monte Carlo simulations, cutting computation time from days to hours.

GPUS

GPUs were initially designed to perform graphical operations for video games. These operations often involve similar or repeated computations on multiple frames, and need to be completed as fast as possible. The technology was later used in the non-graphical areas, such as solving large systems of equations. These non-graphical tasks gave rise to the General Purpose GPUs or GPGPUs. Good candidates of GPGPUs are applications that require identical processing on many versions of the data, and have a high ratio of computations versus the amount of data that needs to be processed. For the rest of this article, when we refer to GPUs, we really mean GPGPUs.

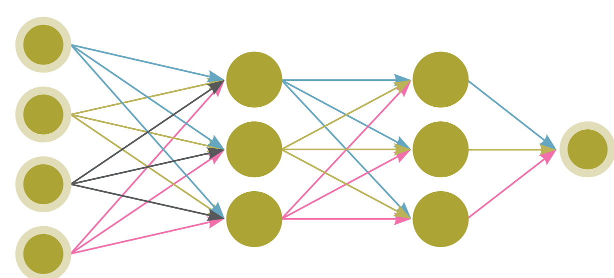
The real power of GPUs lies in its price tag. One NVIDIA Tesla K80 GPU card has 4,992 threads and costs less than $4,000. We can easily insert four K80 cards into a computer and compute 20,000 tasks simultaneously. Computing 20,000 tasks in parallel using CPUs will require a computer grid of millions of dollars. The low cost of GPUs makes daunting computation tasks such as deep learning economically possible. In the meantime, machine learning software, such as SAS (Bequet and Chen, 2016), provide data scientists with an easy methodology to call GPU functions without the knowledge of GPU languages. Inexpensive hardware and easy-to-use software liberate application developers from computational challenges, and enable them to focus on what they are good at: defining problems, collecting and processing data, designing algorithms and analyzing results. Consequently, more and more deep learning applications are springing up in a wide range of fields including health care, transportation, speech recognition, environmental science and more.

DEEP LEARNING

Deep learning is a branch of machine learning. It is inspired by the human brain's biology and its ability to learn via observing and experiencing. Deep learning models are large multilayer artificial neural networks. Artificial neural network started in the 1940s. Its winding journey finally entered into a productive era in recent years, thanks to fast and economical computer accelerators such as GPUs, a flood of digitalized data from the Internet, and virtually infinite storage spaces.

An artificial neural network is a network of computational neurons organized in layers. It has one input layer, at least one hidden layer, and one output layer. We call a neural network a deep net when there are more than one hidden layers. The quintessential deep learning models are the feedforward deep networks. In a feedforward deep net, the input data enter the input layer, go through one hidden layer after another in sequence, and reach the output layer to produce the target value(s). Neurons of one layer take as input the outputs of neurons in the previous layer. There are no feedback connections in which the outputs of a layer are fed back to itself or the previous layers. The feedforward computation can be described as a directed acyclic graph shown in Figure 1.

Figure 1
A Feedforward Deep Learning Network



Consider a feedforward deep network with L layers, and for each layer l, there are I_l nodes, in which the first layer output x^1 equals to the feature vector x from the input data, and the last layer output x^L corresponds to the calculated target variable vector $\hat{y}$. The calculation in each node through the neural network can be recursively represented as

$$x_i^l = g_i^l\left(w_i^{l^T} x^{l-1} + b_i^l\right) \text{ for } l = 2, \ldots L \text{ and } i = 1, \ldots I_l\,,$$

in which function g_i^l is an activation function, such as RELU, logistic sigmoid, or hyperbolic tangent functions; vector w_i^l contains the weights; and scalar b_i^l is a bias term. For notational simplicity, we use vector θ to include weights and biases from all the nodes in the model. The objective is to choose θ to minimize a cost function $J(\theta)$. The cost function defines the error between the target value $\hat{y}$ calculated by the network, and the desired value y passed from the input data. $J(\theta)$ can be mean square error (MSE) for regression problems, or cross-entropy for classification problems. It can be other function depending on the specific application.

Two types of financial applications are good candidates for function approximation using feedforward deep network. 1) Fit functions for hard-to-model assumptions, like policyholder behaviors. 2) Approximate functions to replace computational intensive calculation, such as seriatim valuation or stochastic simulation. An advantage of using feedforward deep network to approximate functions is that there is no need for prior knowledge about the true model. Constructing a neural network is more art than science. Financial analysts, such as actuaries, can construct a feedforward network by trying different combinations of hyperparameters. There are two types of hyperparameters. 1) Model hyperparameters, such as the total number of neurons, the connection between neurons, and the activation functions. 2) Training method hyperparameters, such as cost function, optimization solver, learning rate and initial weights. Often, there is not a single best combination of hyperparameters for a particular problem. Usually the bigger the network and the larger the training data size, the better approximation the trained network achieves. However, it comes with the price of computational efficiency. Our goal is to find a satisfactory set of hyperparameters to achieve target accuracy and efficiency with the constraints of available computation power and training data.

NUMERICAL EXPERIMENTS

In this section, we use an example of large portfolio variable annuity valuation to demonstrate the speed and accuracy that GPUs and Deep Learning can achieve.

We compare the performance of the valuation in the following three settings: Monte Carlo valuation using multi-threaded CPUs, Monte Carlo valuation using GPUs, and GPU-based deep learning valuation. All tests are done in the testing computer with 500GB RAM, 64 hyper-threaded cores at 2.30GHz, and an NVIDIA Tesla K80 GPU card with 4,992 CUDA threads.

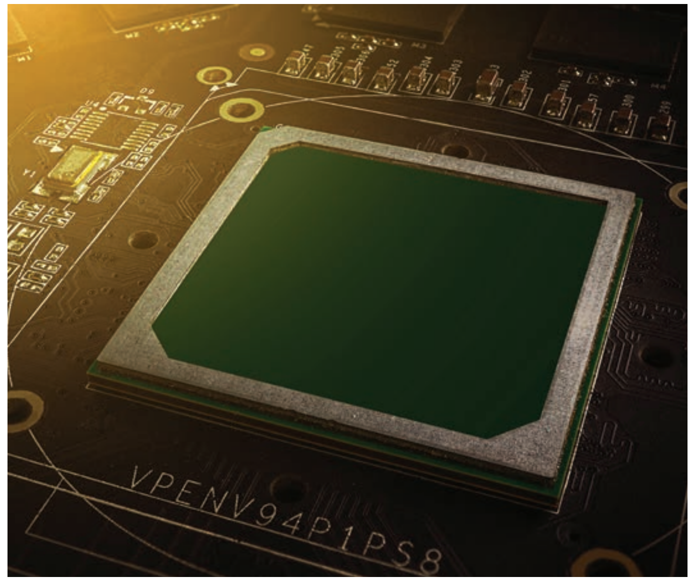
Portfolio Data

The portfolio consists of one million synthetic variable annuity policies with either a GMDB rider or both GMDB and GMWB riders. Each policy has seven attributes which contribute to the policy's valuation. They are guarantee type, gender, age, account value, guarantee value, GMWB withdrawal rate and maturity. Each policy is generated by uniformly drawing values of each attribute from its respective range. For the purpose of comparison, we use the same attributes and value ranges of the input portfolio listed in Table 1 in Hejazi et al. (2015). We also use the same log-normal distribution with a 3 percent risk-free rate and a 20 percent volatility to generate 10,000 risk neutral equity scenarios. The mortality rates follow the same 1996 IAM tables provided by the Society of Actuary. The projection horizon is 25 years.

The model calculates the dollar delta for each of the one million variable annuity policy.

Monte Carlo Valuation

The Monte Carlo valuation algorithm follows Gan (2013). For each policy, we calculate the dollar delta for each of the 10,000 equity scenarios. A policy's dollar delta is the average of the dollar deltas across all scenarios. It takes 44 minutes using the 64 CPU



cores, or 52 seconds using the 4,992 GPU threads, to compute the million policies' dollar deltas on the testing computer.

Deep Learning Valuation

To achieve higher performance, we use deep learning to approximate Monte Carlo valuation on the million variable annuities.

In our experiment, we construct a fully connected feedforward deep neural network with one input layer, eight hidden layers, and one output layer. The input features include two categorical features and six numerical ones. Categorical features are guarantee type (zero for GMDB and one for GMDB+GMWB) and gender (zero for male and one for female). Numerical features are maturity, age, account value, GD/AV (the ratio of guaranteed death benefit over account value), GW/AV (the ratio of guaranteed remaining withdrawal amount over account value) and withdrawal rate. GW/AV and withdrawal rate are zero for policies with only the GMDB rider. For policies with both GMDB and GMWB riders, the time zero values of GD/AV equal to GW/AV equal to the ratio of guarantee value over account value. To ensure fast convergence for network training, we standardize the numerical feature values by taking their z-scores. Each hidden layer has 1,024 neurons with RELU activation function. The output layer calculates the weighted sum of the eighth hidden layer's 1,024 outputs to produce the value of target variable dollar delta.

To train the network, we generate 10,000 variable annuity policies, 8,000 for training and 2,000 for validation. They are 1 percent the size of the input portfolio we need to evaluate. They follow the same distribution as the million-policy portfolio we want to evaluate. We calculate their dollar deltas using Monte Carlo valuation, which takes half a second on the GPUs. Should valuation results from past valuation dates be available, there would be no need to generate new training data with Monte Carlo simulations. Actuaries who are working on production have plenty of historical data to use as inputs for network training.

We train the network using back propagation with the Adam optimizer to find a set of weights and bias to minimize the cost function

$$J(\theta) = \frac{1}{2N}\sum_{i=1}^{N}\left(\frac{\hat{y}_i - y_i}{1+|y_i|}\right)^2.$$

To speed up training, we employ a mini-batch training technique with a batch size of 100. The learning rate is set to 0.001. The initial weight values are generated using truncated normal with mean 0 and standard deviation 0.1. The initial bias values are set to zeros. The network is trained for 88,800 iterations within 14 minutes on the GPUs. The values of $J(\theta)$ are 0.0005 for the training set and 0.0028 for the validation set.

It is worth pointing out the importance of selecting a good cost function that suits the particular problem we are solving. The dollar deltas can vary in a wide range among different variable annuity policies in a portfolio. We do not choose

$$\mathrm{MSE}(\theta) = \frac{1}{2N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2,$$

because it favors those weights and bias that reduce the errors for the y_i's with large absolute values; therefore the accuracy of the model for the y_i's with wsmall absolute values compromised. We also choose not to use

$$\mathrm{MSE}(\theta) = \frac{1}{2N}\sum_{i=1}^{N}\left(\frac{\hat{y}_i - y_i}{y_i}\right)^2,$$

because it is very likely that some policies have dollar deltas at or very close to zero. Using MSRE as the cost function would cause numerical problems. We try a few variations of MSRE. Cost function

$$J(\theta) = \frac{1}{2N}\sum_{i=1}^{N}\left(\frac{\hat{y}_i - y_i}{1+|y_i|}\right)^2$$

gives us the best optimization result.

Once the network is trained, it can be used to approximate the Monte Carlo valuation for variable annuity policies with similar characteristics as the training data, so long as the risk neutral assumptions for equity scenarios stay the same. The trained deep net can replace the entire one-level Monte Carlo valuation. It can also substitute each inner loop Monte Carlo valuation at all time steps along the outer loop scenarios for a nested simulation. In our example, it takes four seconds to compute the dollar deltas for the 1,000,000 policies using the trained deep net. The relative error of the portfolio dollar delta

$$\frac{\sum_{i=1}^{N}\hat{y}_i - \sum_{i=1}^{N}y_i}{\sum_{i=1}^{N}y_i}$$

is 0.0004. It would have taken the same GPU card eight days to complete the nested Monte Carlo valuation for the same portfolio with 1,000 outer loop real world scenarios each having 10,000 inner loop risk neural paths. With the trained deep net to perform the inner valuation, we can complete the nested calculation in 14 hours. We can further reduce the computation time by using more GPU cards simultaneously.

Using the Many Task Computing framework (Bequet and Chen 2017), we are able to integrate CPU and GPU tasks in the same computation job flow without any manual data movement. The end-to-end computation seamlessly conducts data generation and enrichment on CPUs, Monte Carlo simulation and neural network training/inference on GPUs. Figure 2 shows the high level computation job flow.

Performance Results

Table 1 shows the performance results for evaluating one million variable annuity policies using different technologies. We list the hardware information to provide reference for interested readers.

Figure 2
An End-to-End Job Flow for Variable Annuity Valuation With Deep Learning

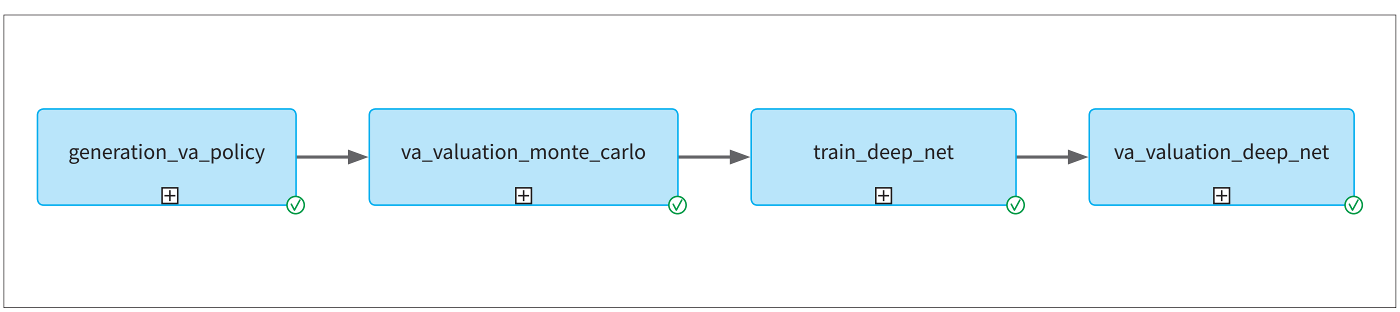


Table 1
Performance under Different Technologies

Technology		Hardware	Monte Carlo Simulation Times (in seconds)
CPU	Monte Carlo Valuation with SAS	64 HT Intel E5-2698 v3 @ 2.30 GHz 500 GB RAM	2,640
GPU	Monte Carlo Valuation with CUDAC	NVIDIA K80 @ 840 MHz 4,992 CUDA Cores	52
	Deep Learning with CUDAC	NVIDIA K80 @ 840 MHz 4,992 CUDA Cores	4

The four-second computation time with deep learning is the time for inference only. We do not include the network training time here because the neural network only needs to be trained once, and can be used for inference many times, as long as the portfolio's characteristics and company's long term view on equity movements do not change.

CONCLUSION AND FUTURE WORK

We have shown that GPUs and GPU-based deep learning can improve computation efficiency by several orders of magnitude. This facilitates timely analysis for better decision making.

As actuaries continue pushing the boundary of product innovation, more complicated modeling is expected, which demands higher computing performance. Fortunately we are living in a world of constant technology breakthroughs. Application Specific Integrated Circuits (ASICs) designed for deep learning training and inference will perform analytics even faster than what we have described in this paper. Preliminary results (Joupi, et al., 2017) indicate that we would at least get another order of magnitude of performance improvements. We will work on financial analytics with ASICs-based deep learning and share the findings with readers in the future. Meanwhile, we see deep learning as a nice tool to help actuaries discover the real patterns of policyholder behaviors. Policyholder behaviors, such as guaranteed living benefits utilization and dynamic lapse, are hard to model. Because deep learning algorithms learn models directly from data, we believe actuaries can train deep neural networks with relevant data and find the credible policyholder behavior assumptions for better valuations. ■

Huina Chen is a principal research statistician developer at SAS Institute. She can be contacted at *huina.chen@sas.com*.

Henry Bequet is a director of development at SAS Institute. He can be contacted at *henry.bequet@sas.com*.

REFERENCES

Bequet, H., Chen, H., 2016. Many Task Computing With GPU Acceleration: An Infrastructure for Easy Principle-Based Modelling. 2016 Society of Actuary Annual Meeting & Exhibit.

Bequet, H., Chen, H., 2017. Accelerate your SAS Programs with GPUs. Paper SASSD706-2017.

Gan, G., 2013. Application of Data Clustering and Machine Learning in Variable Annuity Valuation. Insurance: Mathematics and Economics 53 (3) (2013) 795–801.

Gan, G., Lin, X.S., 2015. Valuation of Large Variable Annuity Portfolios under Nested Simulation: A Functional Data Approach. Insurance: Mathematics and Economics 62, 138-150.

Hejazi, S.A., Jackson, K.R., Gan, G., 2015. A Spatial Interpolation Framework for Efficient Valuation of Large Portfolios of Variable Annuities.

Hejazi, S.A., Jackson, K.R., 2016. Efficient Valuation of SCR via a Neural Network Approach.

Joupi et al., 2017. In-Datacenter Performance Analysis of a Tensor Processing Unit. 44th International Symposium on Computer Architecture (ISCA), 2017.

SOCIETY OF ACTUARIES®

475 N. Martingale Road, Suite 600
Schaumburg, Illinois 60173
p: 847.706.3500 f: 847.706.3599
w: www.soa.org